19. Web Server (Apache)

1. What is Web server and explain it?

A Web server is a system that delivers content or services to end users over the Internet. A Web server consists of a physical server, server operating system (OS) and software used to facilitate HTTP communication.

A computer that runs a Web site. Using the HTTP protocol, the Web server delivers Web pages to browsers as well as other data files to Web-based applications. The Web server includes the hardware, operating system, Web server software, TCP/IP protocols and site content (Web pages, images and other files). If the Web server is used internally and is not exposed to the public, it is an "intranet server" and if the Web server is used in the internet and is exposed to the public, it is an Internet server.

2. What is Protocol?

A uniform set of rules that enable two devices to connect and transmit the data to one another. Protocols determine how data are transmitted between computing devices and over networks. They define issues such as error control and data compression methods. The protocol determines the following type of error checking to be used, data compression method (if any), how the sending device will indicate that it has finished a message and how the receiving device will indicate that it has received the message. Internet protocols include TCP/IP (Transmission Control Protocol / Internet Protocol), HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol) and SMTP (Simple Mail Transfer Protocol).

3. How a Web server works?

(i) If the user types an **URL** in his browsers address bar, the browser will splits that URL into a number of separate parts including address, path name and protocol.

(ii) A DNS (Domain Naming Server) translates the domain name the user has entered into its IP address, a numeric combination that represents the site's true address on the internet.

(iii) The browser now determines which protocol (rules and regulation which the client machine used to communicate with servers) should be used. For example FTP (File Transfer Protocol) and HTTP (Hyper Text Transfer Protocol).

(iv) The server sends a **GET** request to the Web Server to retrieve the address it has been given. For example when a user types http://www.example.com/Myphoto.jpg , the browser sends a **GET** Myphoto.jpg command to example.com server and waits for a response. The server now responds to the browser's requests. It verifies that the given address exist, finds the necessary files, runs the appropriate scripts, exchanges cookies if necessary and returns the results back to the browser. If it cannot locate the file, the server sends an error message to the client.

(v) Then the browser translates the data it has been given into HTML and displays the results to the user.

4. In how many ways can we host the websites?

IP based Web Hosting :

IP based web hosting is usedIP address or hostname web hosting.

Name based Web Hosting :

Hosting the multiple websites using single IP address.

Port based Web Hosting :

Web hosting using another port number ie., other than the default port number.

User based Web Hosting :

We can host the Web sites using the user name and password.

5. What is Apache Web Server?

Apache is a open source web server. It is mostly used web server in the internet. httpd is the deamon that speaks the http or https protocols. It is a text based protocol for sending and receiving the objects over a network connection. The http protocol is sent over the wired network in clear text using default port number 80/tcp. To protect the website we can use https web server for data encryption.

6. What is the profile for Web server?

Package : **httpd**
script : **/etc/init.d/httpd**
Deamon : **httpd**

Configuration file : **/etc/httpd/conf/httpd.conf** (for http)
/etc/httpd/conf.d/ssl.conf (for https)
Document Root : **/var/www/html**
Log files : **/var/log/httpd/access_log**
/var/log/httpd/error_log
Port Number : **80/http and 443/https**

* If we want to configure the httpd server, we have to follow the **ISET** rules. where **I -** Install, **S -** Start,
E - Enable and **T -** Test.
* To access the websites using the CLI mode **e-links, curl tools** are used and to access the websites using
the browser in Linux Firefox is used.

7. How to make the http web server available to the cleint?

(a) First assign the static IP address and hostname to the server.
(b) Check whether the server package by **# rpm -qa httpd*** command.
(c) If not installed, install the web server package by **# yum install httpd* -y** command.
(d) Start the web server and enable web server service at next boot.
 # service httpd start (to start the webserver deamon in RHEL - 6)
 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)
 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)
 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)
(e) Open the browser and access the web server document.
 # firefox (to open the firefox browser)
* Then in address bar type as **http://localhost/manual** and press Enter key.

8. How to configure the IP based virtual host Web server?

(a) First assign the static IP address and hostname to the server.
(b) Check whether the server package by **# rpm -qa httpd*** command.
(c) If not installed, install the web server package by **# yum install httpd* -y** command.
(d) Check the configuration file to configure the http web server by **# rpm -qac httpd** command.
(e) If required open the web server document by **# rpm -qad httpd** command.
(f) Go to the configuration file directory by **# cd /etc/httpd/conf.d**
(g) Create the configuration for IP based hosting.
 # vim /etc/httpd/conf.d/ip.conf

```
<VirtualHost  <IP address of the web server> : 80>
ServerAdmin  root@<hostname of the web server>
ServerName  <hostname of the web server>
DocumentRoot  /var/www/html
        </VirtualHost>

        <Directory  "/var/www/html">
AllowOverride  none
Require  All  Granted
</Directory>
```

(save and exit this file)

Example :

 # vim /etc/httpd/conf.d/ip.conf (create the configuration file)

```
<VirtualHost  172.25.9.11:80>
```

ServerAdmin root@server9.example.com
ServerName server9.example.com
DocumentRoot /var/www/html
</VirtualHost>

<Directory "/var/www/html">
AllowOverride none
Require All Granted
</Directory>

(h) Go to document root directory and create the index.html file.

 # cd /var/www/html

 # vim index.html

```
<html>
    <H1>
        This is IP based Web Hosting
    </H1>
</html>
```

(save and exit this file)

(i) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(j) Add the service to the IP tables and firewall.

In RHEL - 6 :

iptables -A INPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
service iptables save
service iptables restart

In RHEL - 7 :

firewall-cmd --permanent --add-service=http
firewall-cmd --complete-reload

(k) Go to client system, open the firefox browser and type as **http://server9.example.com** in address bar and check the index page is displayed or not.

(l) We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump server9.example.com (access the index page)

9. How to configure the name based web hosting?

(a) Make a directory for virtual or named based hosting.

 # mkdir /var/www/virtual

(b) Go to the configuration file directory by **# cd /etc/httpd/conf.d**

(c) Create the configuration for name based hosting.

 # vim /etc/httpd/conf.d/virtual.conf

<VirtualHost <IP address of the web server> : 80>
ServerAdmin root@<hostname of the web server>
ServerName <virtual hostname of the web server>
DocumentRoot /var/www/virtual

</VirtualHost>

<Directory "/var/www/virtual">
AllowOverride none
Require All Granted
</Directory> (save and exit this file)

Example :

 # vim /etc/httpd/conf.d/virtual.conf (create the configuration file)

<VirtualHost 172.25.9.11:80>
ServerAdmin root@server9.example.com
ServerName www9.example.com
DocumentRoot /var/www/virtual
</VirtualHost>

<Directory "/var/www/virtual">
AllowOverride none
Require All Granted
</Directory>

(d) Go to named based virtual directory and create the index.html file.

 # cd /var/www/virtual

 # vim index.html

<html>
<H1>
This is Name based Web Hosting
</H1>
</html>

(save and exit this file)

(e) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(f) Add the service to the IP tables and firewall.

In RHEL - 6 :

iptables -A INPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
service iptables save
service iptables restart

In RHEL - 7 :

firewall-cmd --permanent --add-service=http
firewall-cmd --complete-reload

(g) Go to client system, open the firefox browser and type as **http://www9.example.com** in address bar and check the index page is displayed or not.

(h) We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump www9.example.com (access the index page)

10. How to configure the port based web hosting?

(a) Make a directory for port based hosting.

 # mkdir /var/www/port

(b) Go to the configuration file directory by **# cd /etc/httpd/conf.d**

(c) Create the configuration for port based hosting.

 # vim /etc/httpd/conf.d/port.conf

```
<VirtualHost  <IP address of the web server> : 8999>
ServerAdmin  root@<hostname of the web server>
ServerName  <port based hostname of the web server>
DocumentRoot  /var/www/port
        </VirtualHost>

        <Directory  "/var/www/port">
AllowOverride  none
Require  All  Granted
</Directory>
```

(save and exit this file)

Example :

 # vim /etc/httpd/conf.d/virtual.conf (create the configuration file)

```
<VirtualHost  172.25.9.11:8999>
ServerAdmin      root@server9.example.com
ServerName       port9.example.com
DocumentRoot     /var/www/port
        </VirtualHost>

        <Directory  "/var/www/port">
AllowOverride  none
Require  All  Granted
</Directory>
```

(d) Go to port based virtual directory and create the index.html file.

 # cd /var/www/port

 # vim index.html

```
        <html>
                <H1>
                        This is Port  based Web Hosting
                </H1>
        </html>
```

(save and exit this file)

(e) Generally port based web hosting requires DNS server. So, we can solve this problem by the following way.

For that open the **/etc/hosts** file enter the server name and IP addresses on both server and client.

 # vim /etc/hosts

172.25.9.11 port5.example.com

(save and exit this file)

(f) By default the web server runs on port number 80. If we want to configure on deferent port number, we have to add the port number in the main configuration file.

 # vim /etc/httpd/conf/httpd.conf

* Go to **Listen : 80** line and open new line below this line and type as,

Listen : 8999

(save and exit this file)

(g) By default SELinux will allow 80 and 8080 port numbers only for webserver. If we use different port numbers other than 80 or 8080 then execute the following command.

 # semanage port -a -t http_port_t -p tcp 8999

(h) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(i) Add the service to the IP tables and firewall.

In RHEL - 6 :

iptables -A INPUT -i eth0 -p tcp -m tcp --deport 8999 -j ACCEPT
iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 8999 -j ACCEPT
service iptables save
service iptables restart

In RHEL - 7 :

firewall-cmd --permanent --add-service=http
firewall-cmd --permanent -add-port=8999/tcp
firewall-cmd --complete-reload

(j) Go to client system, open the firefox browser and type as **http://port9.example.com** in address bar and check the index page is displayed or not.

(k) We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump port9.example.com (access the index page)

11. How to configure user authentication based web hosting?

It will ask user name and password to access this website. So, we have to provide http password.

(f) Go to the configuration file directory by **# cd /etc/httpd/conf.d**

(g) Create the configuration for user authentication based hosting.

 # vim /etc/httpd/conf.d/userbase.conf

<VirtualHost <IP address of the web server> : 80>
ServerAdmin root@<hostname of the web server>
ServerName <hostname of the web server>
DocumentRoot /var/www/html
</VirtualHost>

<Directory "/var/www/html">
AllowOverride none
Require All Granted
AuthType Basic
AuthName "This site is protected"
AuthUserFile /etc/httpd/pass
Require User <user name>
</Directory> (save and exit this file)

Example :

 # vim /etc/httpd/conf.d/userbase.conf (create the configuration file)

```
<VirtualHost 172.25.9.11:80>
ServerAdmin root@server9.example.com
ServerName server9.example.com
DocumentRoot /var/www/html
        </VirtualHost>

        <Directory "/var/www/html">
AllowOverride none
Require All Granted
AuthType Basic
AuthName "This site is protected"
AuthUserFile /etc/httpd/pass
Require User raju
</Directory>
```

(h) Go to document root directory and create the index.html file.

 # cd /var/www/html

 # vim index.html

```
<html>
        <H1>
                This is User Authentication based Web Hosting
        </H1>
</html>
```

(save and exit this file)

(i) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(j) Add the service to the IP tables and firewall.

In RHEL - 6 :

```
# iptables -A INPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
# iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
# service iptables save
# service iptables restart
```

In RHEL - 7 :

```
# firewall-cmd --permanent --add-service=http
# firewall-cmd --complete-reload
```

(k) Create the user and assign the http password.

 # useradd raju

* Don't give the normal password because this user requires the http password.

 # htpasswd -c m /etc/httpd/pass <user name>

Example : # htpasswd -c m /etc/httpd/pass raju

(l) Go to client system, open the firefox browser and type as **http://server9.example.com** in address bar and check the index page is displayed or not. Then it asks password, so we have to provide http password.

(m)We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump server9.example.com (access the index page)

* Then it asks password, so we have to provide http password.

12. How to restrict the web sites access from hosts or domains or networks?

(a) Go to the configuration file directory by **# cd /etc/httpd/conf.d**

(b) Create the configuration for IP based hosting.

 # vim /etc/httpd/conf.d/restrict.conf

```
<VirtualHost 172.25.9.11:80>
ServerAdmin root@server9.example.com
ServerName server9.example.com
DocumentRoot /var/www/html
        </VirtualHost>

        <Directory "/var/www/html">
AllowOverride none
Require All Granted
Order Allow, Deny
Allow from 172.25.9.0 or 172.25.0 (allows 172.25.9 network or 172.25 network to access the websites)
Deny from .my133t.org (deny all the systems of *.my133t.org domain to access the websites)
</Directory>
```

13. How to Redirect the website?

* Redirecting means whenever we access the website, it redirects to another website.

(a) Go to the configuration file directory by **# cd /etc/httpd/conf.d**

(b) Create the configuration for redirect based hosting.

 # vim /etc/httpd/conf.d/rediect.conf

```
<VirtualHost 172.25.9.11:80>
ServerAdmin root@server9.example.com
ServerName server9.example.com
DocumentRoot /var/www/html
Redirect / "http://www.google.com"
        </VirtualHost>

        <Directory "/var/www/html">
AllowOverride none
Require All Granted
</Directory>
```

(save and exit this file)

(c) Go to document root directory and create the index.html file.

 # cd /var/www/html

 # vim index.html

```
        <html>
                <H1>
                        This is Redirect based Web Hosting
                </H1>
        </html>
```

(save and exit this file)

(d) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(e) Add the service to the IP tables and firewall.

In RHEL - 6 :

```
# iptables -A INPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
# iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
# service iptables save
# service iptables restart
```

In RHEL - 7 :

```
# firewall-cmd --permanent --add-service=http
# firewall-cmd --complete-reload
```

(f) Go to client system, open the firefox browser and type as **http://server9.example.com** in address bar and check the redirection google web page is displayed or not.

(g) We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump server9.example.com (access the index page)

* This website redirects to the google website.

14. How to configure the website with alias name?

(a) Go to the configuration file directory by **# cd /etc/httpd/conf.d**

(b) Create the configuration for alias based hosting.

 # vim /etc/httpd/conf.d/alias.conf

```
<VirtualHost 172.25.9.11:80>
ServerAdmin root@server9.example.com
ServerName server9.example.com
DocumentRoot /var/www/html
Alias /private /var/www/html/private
        </VirtualHost>

        <Directory "/var/www/html/private">
AllowOverride none
Require All Granted
</Directory>
```

(save and exit this file)

(c) Create private directory in /var/www/html.

 # mkdir /var/www/html/private

(c) Go to document root private directory and create the index.html file.

 # cd /var/www/html/private

 # vim index.html

```
        <html>
                <H1>
                        This is Alias based Web Hosting
                </H1>
        </html>
```

(save and exit this file)

(d) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(e) Add the service to the IP tables and firewall.

In RHEL - 6 :

iptables -A INPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
service iptables save
service iptables restart

In RHEL - 7 :

firewall-cmd --permanent --add-service=http
firewall-cmd --complete-reload

(f) Go to client system, open the firefox browser and type as **http://server9.example.com/privae** in address bar and check the private or alias based web page is displayed or not.

(g) We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump server9.example.com/private (access the index page)

15. How to configure the directory based web hosting?

(a) Go to the configuration file directory by **# cd /etc/httpd/conf.d**

(b) Create the configuration for direct based hosting.

 # vim /etc/httpd/conf.d/confidential.conf

```
<VirtualHost 172.25.9.11:80>
ServerAdmin root@server9.example.com
ServerName server9.example.com
DocumentRoot /var/www/html
        </VirtualHost>

        <Directory "/var/www/html/confidential">
AllowOverride none
Require All Granted
</Directory>
```

(save and exit this file)

(c) Create confidentialdirectory in /var/www/html.

 # mkdir /var/www/html/confidential

(c) Go to confidential directory and create the index.html file.

 # cd /var/www/html/confidential

 # vim index.html

```
<html>
        <H1>
                This is Alias based Web Hosting
        </H1>
</html>
```

(save and exit this file)

(d) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(e) Add the service to the IP tables and firewall.

In RHEL - 6 :

```
# iptables -A INPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
# iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
# service iptables save
# service iptables restart
```

In RHEL - 7 :

```
# firewall-cmd --permanent --add-service=http
# firewall-cmd --complete-reload
```

(f) Go to client system, open the firefox browser and type as **http://server9.example.com/confidential** in address bar and check the directory based web page is displayed or not.

(g) We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump server9.example.com/confidential (access the index page)

16. How to configure the web server to display the user defined home page not the index.html page?

Normally Apache will look the index.html as the home page by default. If the name changed it will display the home page without configure that one. For that we can do the above as follows.

(i) Go to configuration file directory by **# cd /etc/httpd/conf.d** command.

(ii) Create a userpage configuration file.

 # vim userpage.conf

```
<VirtualHost 172.25.9.11:80>
ServerAdmin root@server9.example.com
ServerName server9.example.com
DocumentRoot /var/www/html
DirectoryIndex userpage.html
        </VirtualHost>

        <Directory "/var/www/html">
        AllowOverride none
        Require All Granted
        </Directory>
```

(save and exit this file)

(iii) Go to document root directory by **# cd /var/www/html** command.

(iv) **# vim userpage.html**

```
        <html>
                <H1>
                        This is userpage as home page web hosting
                </H1>
        </html>
```

(save and exit this file)

(d) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(e) Add the service to the IP tables and firewall.

In RHEL - 6 :

iptables -A INPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
service iptables save
service iptables restart

In RHEL - 7 :

firewall-cmd --permanent --add-service=http
firewall-cmd --complete-reload

(f) Go to client system, open the firefox browser and type as **http://server9.example.com** in address bar and check the user defined web page is displayed or not.

(g) We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump server9.example.com (access the index page)

17. How to configure CGI based web hosting?

CGI content will change dynamically every time the client accessed it. Normal web server will not be used to support this type of web hosting. To access these dynamic pages, we have to configure the web server as ".wsgi" server. The following steps will configure the CGI web server.

(a) Install the CGI package by **# yum install mod_wsgi* -y** command.

(b) Download or create the CGI script file in web server's document root directory.

Example : # cp webapp.wsgi /var/www/html

(c) Create the configuration file for CGI based web hosting.

```
<VirtualHost 172.25.9.11:80>
ServerAdmin root@server9.example.com
ServerName webapp9.example.com
DocumentRoot /var/www/html
WSGIScriptAlias / /var/www/html/webapp.wsgi
</VirtualHost>
```

(d) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(e) Add the service to the IP tables and firewall.

In RHEL - 6 :

iptables -A INPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 80 -j ACCEPT
service iptables save
service iptables restart

In RHEL - 7 :

firewall-cmd --permanent --add-service=http

firewall-cmd --complete-reload

(f) Go to client system, open the firefox browser and type as **http://webapp9.example.com** in address bar and check the CGI based web page is displayed or not.

(g) We can also access the website using **elinks** CLI tool.

 # yum install elinks* -y (install the elinks package)

 # elinks --dump webapp9.example.com (access the index page)

18. What is secured web server?

Secured web server means normal Apache web server with SSL support. In normal web server the data communication is done in plain text format. So, there is no security for data because everyone can access the data. If we want to provide security to the data, then we have to configure the web server with SSL support.

19. What is the profile of secured web server?

Package	:	**mod_ssl**
Configuration file	:	**/etc/httpd/conf.d/ssl.conf**
Private key location	:	**/etc/pki/tls/private**
Public key location	:	**/etc/pki/tls/certs**
Authentication certificate	:	**/etc/pki/tls/certs**
Port number	:	**443**

* Private key extention is **" . key "** and public key extention is **" . crt "**

20. How to configure the secured web server?

(a) Install the web server and secure shell packages.

 # yum install httpd* mod_ssl* -y command.

(b) Download the private key and public certificates.

 # cd /etc/pki/tls/private

 # wget http://classroom.example.com/pub/tls/private/server<no.> . key

 # cd /etc/pki/tls/certs

 # wget http://classroom.example.com/pub/tls/certs/server<no.> . crt

 # wget http://classroom.example.com/pub/example-ca.crt

(c) Create the configuration file for secured web server.

 # vim /etc/httpd/conf.d/https.conf

```
<VirtualHost 172.25.9.11:443>
ServerAdmin root@server9.example.com
ServerName server9.example.com
DocumentRoot /var/www/html
</VirtualHost>
```

(d) We have to copy 7 lines from ssl.conf file to https.conf file.

 # vim -O ssl.conf https.conf

Copy the line numbers 70, 75, 80, 93, 100, 107, 116 copy and paste them in https.conf file.

So, after copied those line the https.conf file should be as below.

```
<VirtualHost 172.25.9.11:443>
ServerAdmin root@server9.example.com
ServerName server9.example.com
SSLEngine on
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
SSLCertificateFile /etc/pki/tls/certs/server9.crt
SSLCertificateKeyFile /etc/pki/tls/private/server9.key
#SSLCertificateChainFile /etc/pki/tls/certs/example-ca.crt
DocumentRoot /var/www/html
```

</VirtualHost>
<Directory "/var/www/html">
AllowOverride
Require All Granted
</Directory>

(save and exit this file)

(e) Go to document root directory by **# cd /var/www/html** command.

(f) **# vim index.html**

<html>
<H1>
This is a secured web hosting
</H1>
</html>

(save and exit this file)

(g) Restart the web server deamon.

 # service httpd start (to start the webserver deamon in RHEL - 6)

 # chkconfig httpd on (to enable the service at next boot in RHEL - 6)

 # systemctl restart httpd (to start the webserver deamon in RHEL - 7)

 # systemctl enable httpd (to enable the service at next boot in RHEL - 7)

(h) Add the service to the IP tables and firewall.

In RHEL - 6 :

iptables -A INPUT -i eth0 -p tcp -m tcp --deport 443 -j ACCEPT
iptables -A OUTPUT -i eth0 -p tcp -m tcp --deport 443 -j ACCEPT
service iptables save
service iptables restart

In RHEL - 7 :

firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
firewall-cmd --complete-reload

(i) Go to client system, open the firefox browser and type as **https://server9.example.com/** in address bar and check the secured web page is displayed or not.

21. How to generate our own private and public keys using crypto-utils package?

(i) Install the package by **# yum install crypto-utils* -y** command.

(ii) Create our own public and private keys by **# genkey <hostname of the server>** command.

Example : #genkey server9.example.com (one window will be opened and we have to enter the details)

Click on Next ---> Don't change the default size ---> Next ---> No --->The keys are generated in their directories.

Other useful commands :

httpd -t (to check the web server configuration file for syntax errors)