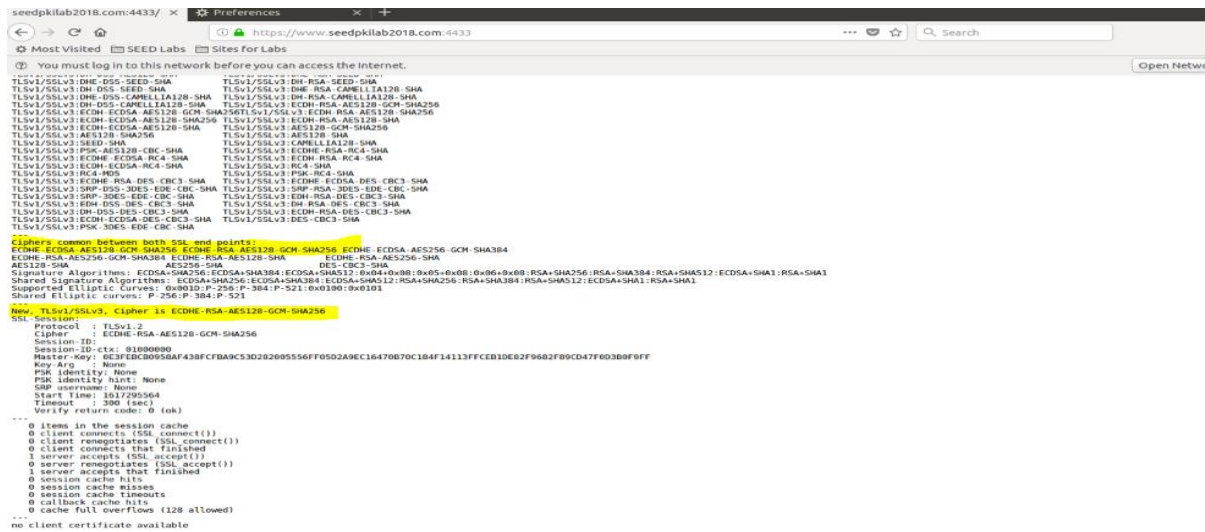
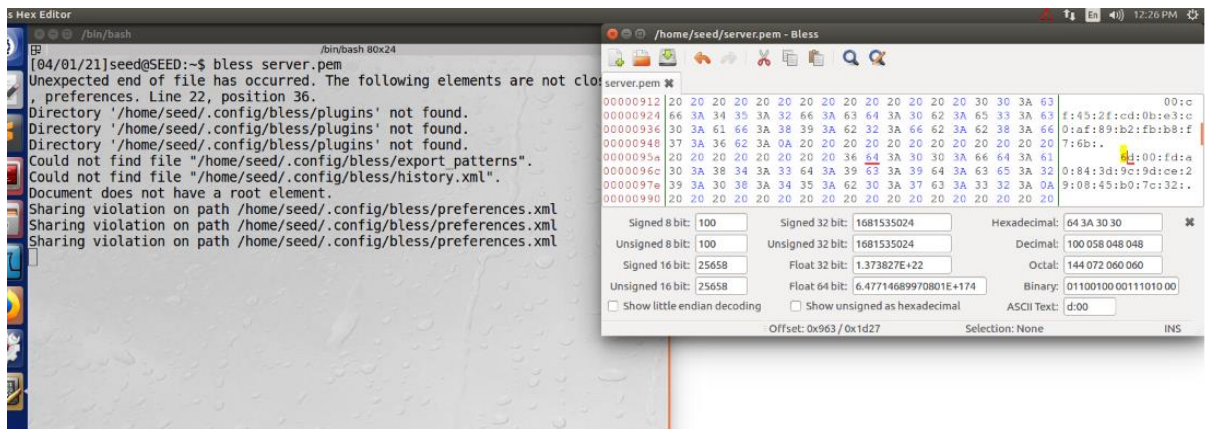
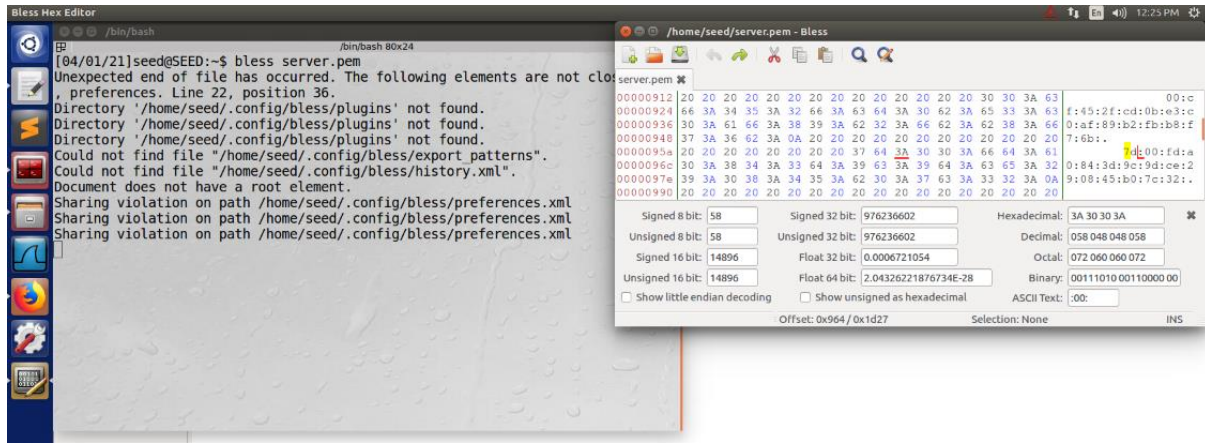


## CS 767 - Public Key Infrastructure Lab

### Task 3: Deploying Certificate in an HTTPS Web Server :-



```
← → ↻ 🏠 https://www.seedpkilab2018.com:4433
Most Visited SEED Labs Sites for Labs
You must log in to this network before you can access the internet. Open

TLsv1/SSLv3: ECDHE-RSA-AES128-SHA TLsv1/SSLv3: ECDHE-ECDSA-AES128-SHA
TLsv1/SSLv3: SRP-DSS-AES-128-CBC-SHA TLsv1/SSLv3: SRP-RSA-AES-128-CBC-SHA
TLsv1/SSLv3: SRP-AES-128-CBC-SHA TLsv1/SSLv3: DH-DSS-AES128-GCM-SHA256
TLsv1/SSLv3: DH-DSS-AES128-GCM-SHA256 TLsv1/SSLv3: DH-RSA-AES128-GCM-SHA256
TLsv1/SSLv3: DH-RSA-AES128-GCM-SHA256 TLsv1/SSLv3: DH-RSA-AES128-SHA256
TLsv1/SSLv3: DH-DSS-AES128-SHA256 TLsv1/SSLv3: DH-RSA-AES128-SHA256
TLsv1/SSLv3: DH-DSS-AES128-SHA TLsv1/SSLv3: DH-RSA-AES128-SHA
TLsv1/SSLv3: DH-DSS-SEED-SHA TLsv1/SSLv3: DH-RSA-SEED-SHA
TLsv1/SSLv3: DH-DSS-SEED-SHA TLsv1/SSLv3: DH-RSA-CAMELLIA128-SHA
TLsv1/SSLv3: DH-DSS-CAMELLIA128-SHA TLsv1/SSLv3: DH-RSA-CAMELLIA128-SHA
TLsv1/SSLv3: ECDH-ECDSA-AES128-GCM-SHA256 TLsv1/SSLv3: ECDH-RSA-AES128-GCM-SHA256
TLsv1/SSLv3: ECDH-ECDSA-AES128-SHA256 TLsv1/SSLv3: ECDH-RSA-AES128-SHA256
TLsv1/SSLv3: ECDH-ECDSA-AES128-SHA TLsv1/SSLv3: ECDH-RSA-AES128-SHA
TLsv1/SSLv3: AES128-SHA256 TLsv1/SSLv3: AES128-SHA
TLsv1/SSLv3: SEED-SHA TLsv1/SSLv3: CAMELLIA128-SHA
TLsv1/SSLv3: PSK-AES128-CBC-SHA TLsv1/SSLv3: ECDHE-RSA-RC4-SHA
TLsv1/SSLv3: ECDHE-ECDSA-RC4-SHA TLsv1/SSLv3: ECDH-RSA-RC4-SHA
TLsv1/SSLv3: ECDH-ECDSA-RC4-SHA TLsv1/SSLv3: RC4-SHA
TLsv1/SSLv3: RC4-MD5 TLsv1/SSLv3: PSK-RC4-SHA
TLsv1/SSLv3: ECDHE-RSA-DES-CBC3-SHA TLsv1/SSLv3: ECDHE-ECDSA-DES-CBC3-SHA
TLsv1/SSLv3: SRP-DSS-3DES-EDE-CBC-SHA TLsv1/SSLv3: SRP-RSA-3DES-EDE-CBC-SHA
TLsv1/SSLv3: SRP-3DES-EDE-CBC-SHA TLsv1/SSLv3: EDH-RSA-DES-CBC3-SHA
TLsv1/SSLv3: EDH-DSS-DES-CBC3-SHA TLsv1/SSLv3: DH-RSA-DES-CBC3-SHA
TLsv1/SSLv3: DH-DSS-DES-CBC3-SHA TLsv1/SSLv3: ECDH-RSA-DES-CBC3-SHA
TLsv1/SSLv3: ECDH-ECDSA-DES-CBC3-SHA TLsv1/SSLv3: DES-CBC3-SHA
TLsv1/SSLv3: PSK-3DES-EDE-CBC-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:0x04+0x08:0x05+0x08:0x06+0x08:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA1:RSA+SHA1
...
Reused, TLsv1/SSLv3, Cipher is ECDHE-RSA-AES128-GCM-SHA256
SSL-Session:
  Protocol : TLSv1.2
  Cipher : ECDHE-RSA-AES128-GCM-SHA256
  Session-ID: C0B015E290F063CD0F899E0A0F31E9547FE1D285D420017CCE807031E2B0013
  Session-ID-ctx: 01000000
  Master-Key: 1A5E455B144109759E57E564993BA67F136A023913308FB8208F2A075F432160033E29FC6074710762EBA623CA80908
  Key-Arg : None
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  Start Time: 1617295267
  Timeout : 300 (sec)
  Verify return code: 0 (ok)
...
  0 items in the session cache
  0 client connects (SSL_connect())
  0 client renegotiates (SSL_connect())
  0 client connects that finished
  3 server accepts (SSL_accept())
  0 server renegotiates (SSL_accept())
  3 server accepts that finished
  2 session cache hits
  0 session cache misses
  0 session cache timeouts
  0 callback cache hits
  0 cache full overflows (128 allowed)
...
no client certificate available
```

Step 4. Testing our HTTPS website. Now, point the browser to <https://SEEDPKILab2018.com:4433>. Deliverable.:

1. Modify a single byte of server.pem, and restart the server, and reload the URL. What do you observe? Make sure you restore the original server.pem afterward. Note: the server may not be able to restart if certain places of server.pem is corrupted; in that case, choose another place to modify.

**Solution:** I observed content of the page were changed when I modify a single bit from 7d to 6d and also, I noted that from “New TLSv1/SSLv3” to “Reused TLSv1/SSLv3”.

2. Since SEEDPKILab2018.com points to the localhost, if we use <https://localhost:4433> instead, we will be connecting to the same web server. Please do so, describe and explain your observation.

**Solution:** When I used the <https://localhost:4433> instead of <https://www.seedpkilab2018.com:4433>, I observed that https is not allowing because of certificate issue (“localhost:4433 uses an invalid security certificate”) and browsers will not consider just any certificate valid, which indicates that the common name in the certificate does not match the URL used to access the web site.

#### Task 4: Deploying Certificate in an Apache-Based HTTPS Website:-

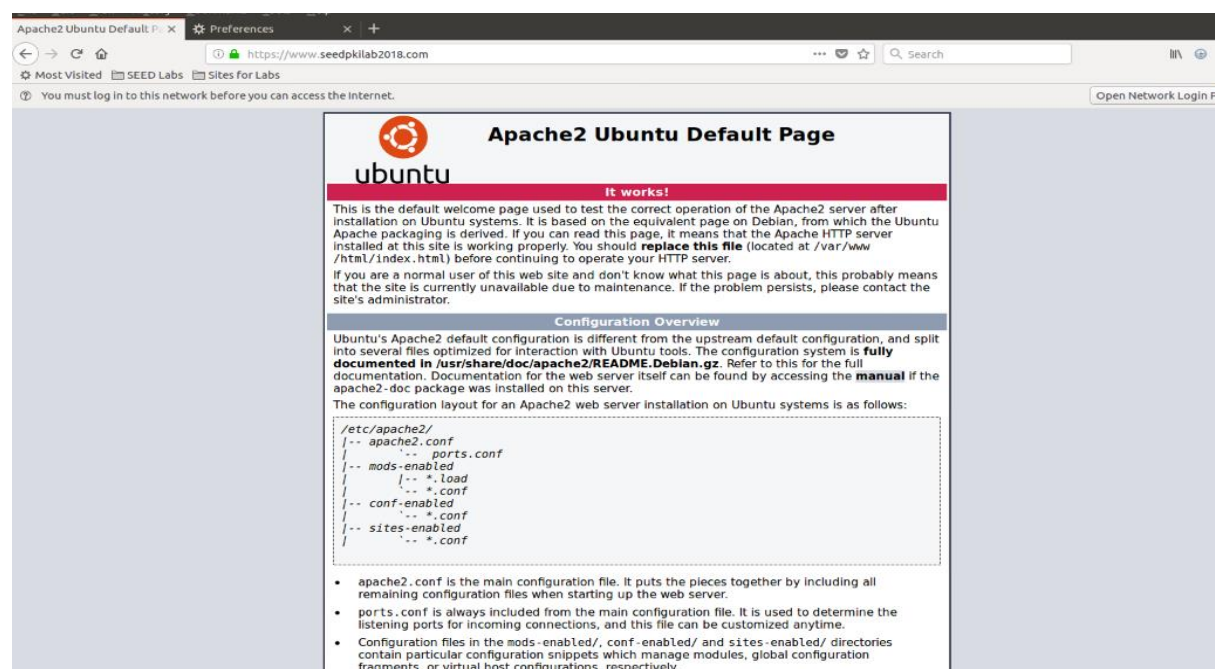
Please describe the steps that you have taken, the contents that you add to Apache's configuration file, and screenshots of the final outcome showing that you can successfully browse the HTTPS site.

I created the KGBANK folder, inside that I have copied crt and key file to pem extension. I opened default-ssl.conf and added the piece of code and run the apache 2 services.

PFB Screen shots of the steps that I have followed.

```
<VirtualHost *:443>
    ServerName www.seedpkilab2018.com
    DocumentRoot /var/www/html
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /home/seed/KGBANK/KG_cert.pem
    SSLCertificateKeyFile /home/seed/KGBANK/KG_key.pem
</VirtualHost>
```

```
[04/01/21]seed@SEED:~$ cd KGBANK
[04/01/21]seed@SEED:~/KGBANK$ cat server.crt > KG_cert.pem
[04/01/21]seed@SEED:~/KGBANK$ cat server.key > KG_key.pem
[04/01/21]seed@SEED:~/KGBANK$ sudo vim /etc/apache2/sites-available/default-ssl.conf
[04/01/21]seed@SEED:~/KGBANK$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using ::1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
[04/01/21]seed@SEED:~/KGBANK$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
[04/01/21]seed@SEED:~/KGBANK$ sudo service apache2 restart
Enter passphrase for SSL/TLS keys for www.seedpkilab2018.com:443 (RSA): ****
[04/01/21]seed@SEED:~/KGBANK$ sudo service apache2 reload
[04/01/21]seed@SEED:~/KGBANK$
```





## Task 5: Launching a Man-In-The-Middle Attack

Step 1:

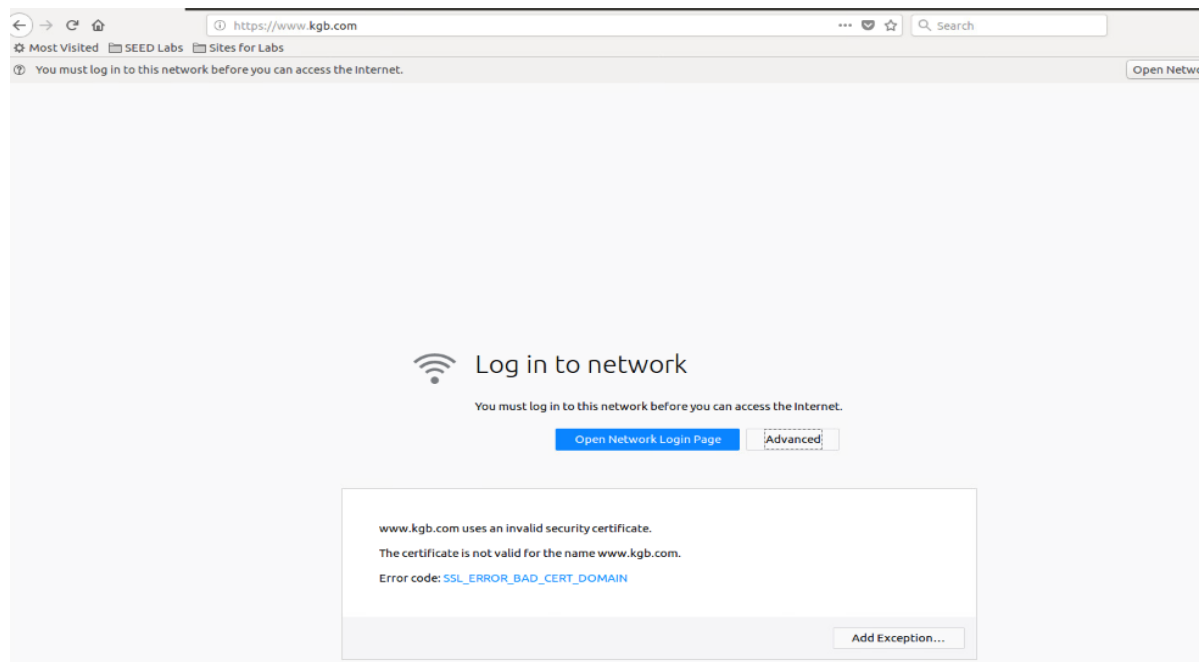
```
<VirtualHost *:443>
    ServerName www.seedpkilab2018.com
    DocumentRoot /var/www/html
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /home/seed/KGBANK/KG_cert.pem
    SSLCertificateKeyFile /home/seed/KGBANK/KG_key.pem
</VirtualHost>

<VirtualHost *:443>
    ServerName www.kgb.com
    DocumentRoot /var/www/html
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /home/seed/KGBANK/KG_cert.pem
    SSLCertificateKeyFile /home/seed/KGBANK/KG_key.pem
</VirtualHost>
```

Step 2:

```
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
127.0.0.1 User
127.0.0.1 Attacker
127.0.0.1 Server
127.0.0.1 www.SeedLabSQLInjection.com
127.0.0.1 www.xsslabelgg.com
127.0.0.1 www.csrflabelgg.com
127.0.0.1 www.csrflabelattacker.com
127.0.0.1 www.repackagingattacklab.com
127.0.0.1 www.seedlabclickjacking.com
127.0.0.1 www.seedpkilab2018.com
127.0.0.1 www.kgb.com
```

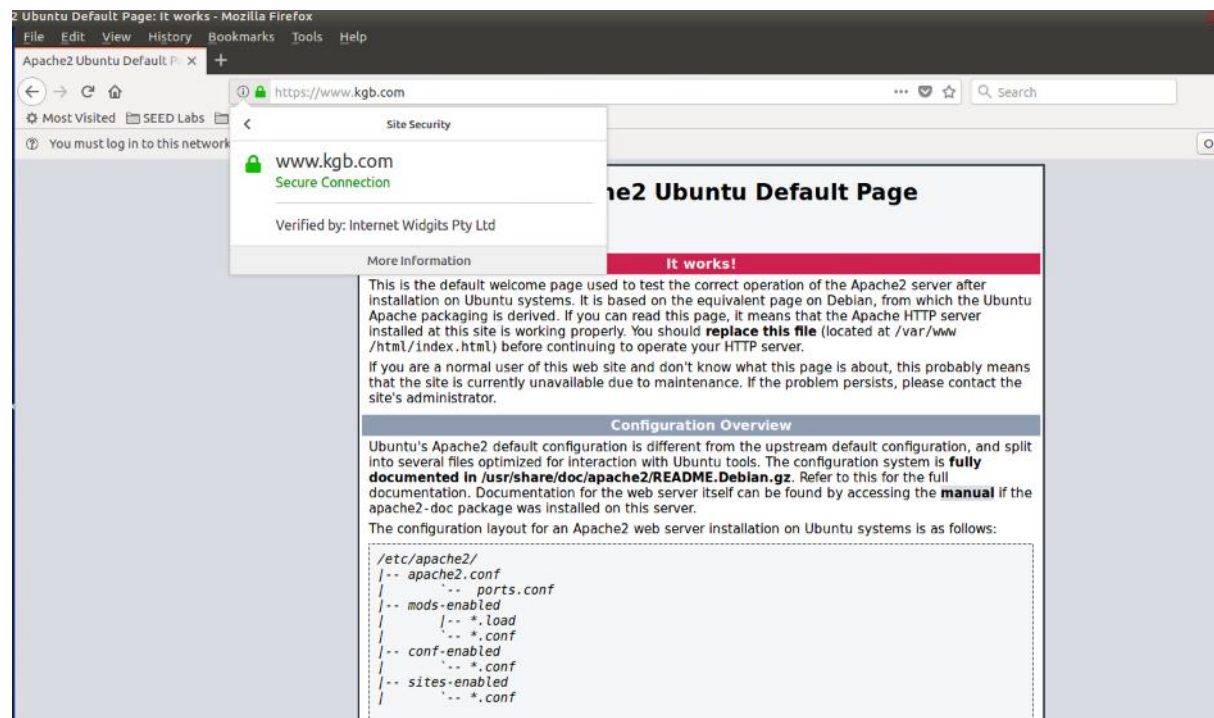
Step 3:



## Task 6: Launching a Man-In-The-Middle Attack with a Compromised CA :

```
<VirtualHost *:443>
    ServerName www.seedpkilab2018.com
    DocumentRoot /var/www/html
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /home/seed/KGBANK/KG_cert.pem
    SSLCertificateKeyFile /home/seed/KGBANK/KG_key.pem
</VirtualHost>

<VirtualHost *:443>
    ServerName www.kgb.com
    DocumentRoot /var/www/html
    DirectoryIndex index.html
    SSLEngine On
    SSLCertificateFile /home/seed/KGBANK/kgbcrt.pem
    SSLCertificateKeyFile /home/seed/KGBANK/KG_key.pem
</VirtualHost>
```



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `https://www.kgb.com`. A security warning overlay is present, indicating a "Secure Connection" for `www.kgb.com`, verified by Internet Widgits Pty Ltd. The main content area displays the "Apache2 Ubuntu Default Page" with a red banner stating "It works!". Below this, there is a "Configuration Overview" section with the following text:

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

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