

Lab Manual 5

Checkpoint-1:

```
GNU nano 6.2 hosts
127.0.0.1 localhost
127.0.0.1 example.com
127.0.0.1 webserverlab.com
127.0.1.1 me0r

# 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
```

```
GNU nano 6.2 html/index.html
<html>

<head>
  <title> Lab task 5 by taohid </title>
</head>

<body>
  <h1> hello I am taohid, This is Lab task 5 </h1>
</body>

</html>
```

```
$ cd
$ sudo mkdir -p /var/www/example.com/html
$ sudo chown -R $USER:$USER /var/www/example.com/html
$ sudo chmod -R 755 /var/www/example.com
$ cd /var/www/example.com
$ pwd
/var/www/example.com
$ sudo nano html/index.html
$ sudo nano html/index.html
$ cd
```

Checkpoint-2:

```
Terminal
GNU nano 6.2 /etc/apache2/sites-available/example.com.conf *
<VirtualHost *:80>
    ServerAdmin webmaster@example.com
    ServerName example.com
    ServerAlias www.example.com

    # Redirect all traffic to HTTPS
    # Redirect permanent / https://example.com/

    ErrorLog ${APACHE_LOG_DIR}/example.com_error.log
    CustomLog ${APACHE_LOG_DIR}/example.com_access.log combined
</VirtualHost>
```

```
Terminal
$ sudo nano /etc/apache2/sites-available/example.com.conf
$ sudo a2ensite example.com.conf
[sudo] password for codermehraj:
Site example.com already enabled
$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
systemctl reload apache2
$ sudo systemctl reload apache2
$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
$ sudo systemctl restart apache2
```



hello I am taohid, This is Lab task 5

```
Terminal
$ sudo nano /etc/apache2/sites-available/000-default.conf
$ sudo nano /etc/apache2/sites-available/example.com.conf
$ sudo systemctl reload apache2
$
```

```
Terminal
GNU nano 6.2 /etc/apache2/sites-available/example.com.conf
<VirtualHost *:80>
    ServerAdmin webmaster@example.com
    ServerName example.com
    ServerAlias www.example.com

    DocumentRoot /var/www/example.com/html

    # Redirect all traffic to HTTPS
    # Redirect permanent / https://example.com/

    ErrorLog ${APACHE_LOG_DIR}/example.com_error.log
    CustomLog ${APACHE_LOG_DIR}/example.com_access.log combined
</VirtualHost>
```

```
#####
[ ca ]
default_ca      = CA_default          # The default ca section

#####
[ CA_default ]

dir              = ./demoCA           # Where everything is kept
certs            = $dir/certs         # Where the issued certs are kept
crl_dir          = $dir/crl           # Where the issued crl are kept
database         = $dir/index.txt     # database index file.
#unique_subject  = no                 # Set to 'no' to allow creation of
                                      # several certs with same subject.
new_certs_dir    = $dir/newcerts      # default place for new certs.

certificate      = $dir/cacert.pem    # The CA certificate
serial          = $dir/serial         # The current serial number
crlnumber        = $dir/crlnumber     # the current crl number
                                      # must be commented out to leave a V1 CRL
crl              = $dir/crl.pem       # The current CRL
private_key      = $dir/private/cakey.pem # The private key

x509_extensions = usr_cert           # The extensions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt        = ca_default          # Subject Name options
cert_opt        = ca_default          # Certificate field options
```

```
Terminal
$ cd /usr/lib/ssl/
$ ls
certs misc openssl.cnf private
$ cd
$ mkdir taohid_lab5/
$ ls
Desktop Documents Downloads Music Pictures Public snap taohid_lab5 Templates Videos
$ cd taohid_lab5
$ ls
$ cp /usr/lib/ssl/openssl.cnf .
$ ls
openssl.cnf
$ nano openssl.cnf
$ nano openssl.cnf
$ nano openssl.cnf
$ mkdir demoCA
$ cd demoCA
$ mkdir certs
$ mkdir crl
$ mkdir newcerts
$ touch index.txt
$ touch serial
$
```

```
$ ls
Desktop Documents Downloads Music Pictures Public snap taohid_lab5 Templates Videos
$ cd taohid_lab5
$ ls
demoCA openssl.cnf
$ openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl.cnf
.....+.....*.....+.
.....+.
.....+.
.....+.
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BD
State or Province Name (full name) [Some-State]:Barisal
Locality Name (eg, city) []:Barisal
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Taohid
Organizational Unit Name (eg, section) []:SUST
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:
$
```


[illegible]

```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BD
State or Province Name (full name) [Some-State]:Barisal
Locality Name (eg, city) []:Barisal
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Taohid
Organizational Unit Name (eg, section) []:SUST
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:hello
An optional company name []:
$ openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 4096 (0x1000)
    Validity
        Not Before: Jul  7 15:26:34 2024 GMT
        Not After : Jul  7 15:26:34 2025 GMT
    Subject:
        countryName             = BD
        stateOrProvinceName     = Barisal
        organizationName        = Taohid
        organizationalUnitName  = SUST
        commonName              = example.com
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Subject Key Identifier:
            C5:5B:0B:D7:EB:01:E7:20:AF:F9:41:E5:CE:B0:35:F3:EB:9C:1C:80
        X509v3 Authority Key Identifier:
            51:15:6D:FE:C0:0C:25:F7:85:A8:6B:EF:53:F8:75:F4:EB:41:C8:A4
Certificate is to be certified until Jul  7 15:26:34 2025 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
$ cp server.key server.pem
$ cat server.crt >> server.pem
$
```

```
example.com:4433/ x Apache Ubuntu Default P... x +
https://example.com:4433

s server -cert server.pem -www
Secure Renegotiation IS NOT supported
Ciphers supported in s server binary
...
TLSv1.3 : TLS_AES_256_GCM_SHA384 : TLSv1.3 : TLS_CHACHA20_POLY1305_SHA256
TLSv1.3 : TLS_AES_128_GCM_SHA256 : TLSv1.2 : ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1.2 : ECDHE-RSA-AES256-GCM-SHA384 : TLSv1.2 : DHE-RSA-AES256-GCM-SHA384
TLSv1.2 : ECDHE-ECDSA-CHACHA20-POLY1305 : TLSv1.2 : ECDHE-RSA-CHACHA20-POLY1305
TLSv1.2 : DHE-RSA-CHACHA20-POLY1305 : TLSv1.2 : ECDHE-ECDSA-AES128-GCM-SHA256
TLSv1.2 : ECDHE-RSA-AES128-GCM-SHA256 : TLSv1.2 : DHE-RSA-AES128-GCM-SHA256
TLSv1.2 : ECDHE-ECDSA-AES256-SHA384 : TLSv1.2 : ECDHE-RSA-AES256-SHA384
TLSv1.2 : DHE-RSA-AES256-SHA256 : TLSv1.2 : ECDHE-ECDSA-AES128-SHA256
TLSv1.2 : ECDHE-RSA-AES128-SHA256 : TLSv1.2 : DHE-RSA-AES128-SHA256
TLSv1.0 : ECDHE-ECDSA-AES256-SHA : TLSv1.0 : ECDHE-RSA-AES256-SHA
SSLv3 : DHE-RSA-AES256-SHA : TLSv1.0 : ECDHE-ECDSA-AES128-SHA
TLSv1.0 : ECDHE-RSA-AES128-SHA : SSLv3 : DHE-RSA-AES128-SHA
TLSv1.2 : RSA-PSK-AES256-GCM-SHA384 : TLSv1.2 : DHE-PSK-AES256-GCM-SHA384
TLSv1.2 : RSA-PSK-CHACHA20-POLY1305 : TLSv1.2 : DHE-PSK-CHACHA20-POLY1305
TLSv1.2 : ECDHE-PSK-CHACHA20-POLY1305 : TLSv1.2 : AES256-GCM-SHA384
TLSv1.2 : PSK-AES256-GCM-SHA384 : TLSv1.2 : PSK-CHACHA20-POLY1305
TLSv1.2 : RSA-PSK-AES128-GCM-SHA256 : TLSv1.2 : DHE-PSK-AES128-GCM-SHA256
TLSv1.2 : AES128-GCM-SHA256 : TLSv1.2 : PSK-AES128-GCM-SHA256
TLSv1.2 : AES256-SHA256 : TLSv1.2 : AES128-SHA256
TLSv1.0 : ECDHE-PSK-AES256-CBC-SHA384 : TLSv1.0 : ECDHE-PSK-AES256-CBC-SHA
SSLv3 : SRP-RSA-AES-256-CBC-SHA : SSLv3 : SRP-AES-256-CBC-SHA
TLSv1.0 : RSA-PSK-AES256-CBC-SHA384 : TLSv1.0 : DHE-PSK-AES256-CBC-SHA384
SSLv3 : RSA-PSK-AES256-CBC-SHA : SSLv3 : DHE-PSK-AES256-CBC-SHA
SSLv3 : AES256-SHA : TLSv1.0 : PSK-AES256-CBC-SHA384
TLSv1.0 : PSK-AES256-CBC-SHA : TLSv1.0 : ECDHE-PSK-AES128-CBC-SHA256
SSLv3 : ECDHE-PSK-AES128-CBC-SHA : SSLv3 : SRP-RSA-AES-128-CBC-SHA
SSLv3 : SRP-AES-128-CBC-SHA : TLSv1.0 : RSA-PSK-AES128-CBC-SHA256
TLSv1.0 : DHE-PSK-AES128-CBC-SHA256 : SSLv3 : RSA-PSK-AES128-CBC-SHA
SSLv3 : DHE-PSK-AES128-CBC-SHA : SSLv3 : AES128-SHA
TLSv1.0 : PSK-AES128-CBC-SHA256 : SSLv3 : PSK-AES128-CBC-SHA
...
Ciphers common between both SSL end points:
TLS_AES_128_GCM_SHA256 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_256_GCM_SHA384
ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-RSA-CHACHA20-POLY1305 ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES256-SHA ECDHE-ECDSA-AES128-SHA ECDHE-RSA-AES128-SHA
ECDHE-RSA-AES256-SHA AES128-GCM-SHA256 AES256-GCM-SHA384
AES128-SHA AES256-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA1:RSA+SHA1
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512
Supported groups: x25519:secp256r1:secp384r1:secp521r1:fFdh2048:fFdh3072
Shared groups: x25519:secp256r1:secp384r1:secp521r1:fFdh2048:fFdh3072
...
New, TLSv1.3, Cipher is TLS_AES_128_GCM_SHA256
SSL-Session:
  Protocol : TLSv1.3
  Cipher : TLS_AES_128_GCM_SHA256
  Session-ID: 0E73131845371C8CD1074F4CC56E5E216500B3F7C9092E023681912CAAD0E
  Session-ID-ctx: 01000000
  Resumption PSK: 040E93C9313FD0BF2C2AF05308C1EF7BA5ADFEB18B23E20FC978B6395A2747
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  Start Time: 172936648
  Timeout : 7200 (sec)
  Verify return code: 0 (ok)
  Extended master secret: no
  Max Early Data: 0
...
0 items in the session cache
0 client connects (SSL_connect())
0 client renegotiates (SSL_connect())
```

```
Terminal

$ pwd
/home/codermehraj/certs
$ ls
$ openssl genrsa -des3 -out myCA.key 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
$ openssl genrsa -des3 -out myCA.key 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
$ openssl req -x509 -new -nodes -key myCA.key -sha256 -days 1825 -out myCA.pem
Enter pass phrase for myCA.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BD
State or Province Name (full name) [Some-State]:Barisal
Locality Name (eg, city) []:Barisal
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Taohid
Organizational Unit Name (eg, section) []:SUST
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:
```



```
$ sudo apt-get install -y ca-certificates
[sudo] password for codermehraj:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
$ sudo cp ~/certs/myCA.pem /usr/local/share/ca-certificates/myCA.crt
$ sudo update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

Adding debian:myCA.pem
done.
done.
$
```

```
$ sudo apt-get install -y ca-certificates
[sudo] password for codermehraj:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
$ sudo cp ~/certs/myCA.pem /usr/local/share/ca-certificates/myCA.crt
$ sudo update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

Adding debian:myCA.pem
done.
done.
$
```

```
Terminal
For some fields there will be a default value,
if you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BD
State or Province Name (full name) [Some-State]:Dhaka
Locality Name (eg, city) []:Dhaka
Organization Name (eg, company) [Internet Widgits Pty Ltd]:TaoHid
Organizational Unit Name (eg, section) []:SUST
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:hello
An optional company name []:
$ openssl x509 -req -in hellfish.test.csr -CA myCA.pem -CAkey myCA.key \
-C "createserial -out hellfish.test.crt -days 825 -sha256 -extfile hellfish.test.ext"
> Can't open "hellfish.test.ext" for reading, No such file or directory
40B73C50FA720000:error:00000002:system library:BIO_new_file:No such file or directory:../crypto/bio/bss_file.c:67:calling fopen(hellfish.test.ext, r)
40B73C50FA720000:error:10000000:BIO routines:BIO_new_file:no such file:../crypto/bio/bss_file.c:75:
$ ls
hellfish.test.csr  hellfish.test.key  myCA.key  myCA.pem
$ cd
$ ls
certs  Desktop  Documents  Downloads  Music  Pictures  Public  snap  taoHid_lab5  Templates  Videos
$ cd ..
$ cd ..
$ pwd
/
$ ls
bin  boot  cdrom  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run /sbin  snap  srv  sys  tmp  usr  var
$ cd var
$ ls
backups  cache  crash  lib  local  lock  log  mail  metrics  opt  run  snap  spool  tmp  www
$ cd www
$ ls
example.com  html
$ cd example.com
$ ls
html
$ cd ..
$ cd ..
$ pwd
/var
$ cd ..
$ ls
bin  boot  cdrom  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run /sbin  snap  srv  sys  tmp  usr  var
$ cd etc
$ cd apache2
$ cd sites-available
$ ls
000-default.conf  default-ssl.conf  default-ssl.conf.bak  example.com.conf
$ pwd
/etc/apache2/sites-available
$ ls
000-default.conf  default-ssl.conf  default-ssl.conf.bak  example.com.conf
$
```


Checkpoint-4:


Lab task 5 by taohid

Apache2 Ubuntu Default Page

+

← → ↻

https://webserverlab.com



Apache2 Default Page

It works!

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.

Configuration Overview

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 [just/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
|
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dissmod`, `a2ensite`, `a2disssite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2` and is managed using `systemd`, so to start/stop the service use `systemctl start apache2` and `systemctl stop apache2`, and use `systemctl status apache2` and `journalctl -u apache2` to check status. `system` and `apache2ctl` can also be used for service management if desired. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots