VANIER COLLEGE – Computer Engineering Technology – Autumn 2021

Network Systems Design (247-509-VA)

Leonardo Fusser (1946995)

LABORATORY EXPERIMENT 2

Managing a Web Server

NOTE:

To be completed in one lab session of 3 hrs.

No formal report is required. Answer all questions, attached necessary screen shot, and include a discussion and conclusion session. To be submitted via Lea, **at the end of the lab session**.

This exercise is to be done individually except where specified in the procedure. **Each** student must submit a lab results with original observations and conclusions.

OBJECTIVES:

After performing this experiment, the student will be able to:

1. Download, install, and verify a webs server application.
2. Verify the default web server configuration file.
3. Use netstat command to display protocol statistic and connection information of a computer.
4. Capture and analyze HTTP traffic with Wireshark.

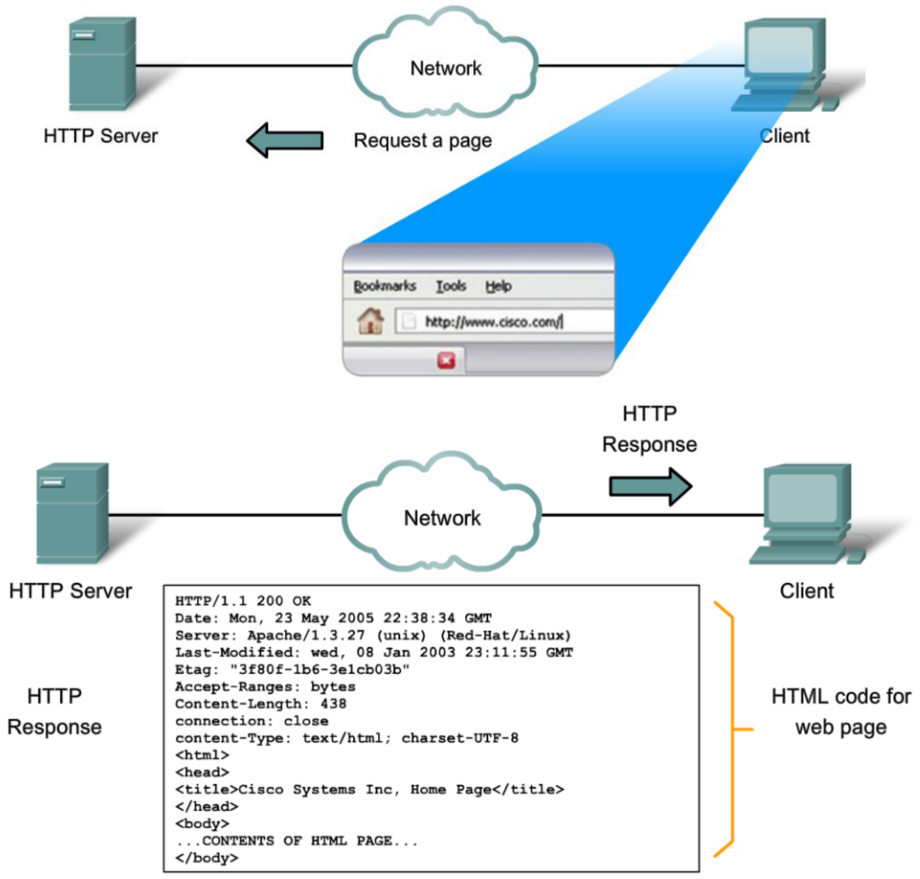
THEORY

WWW Service

When a web address (or URL) is typed into a web browser, the web browser establishes a connection to the web service running on the server using the HTTP protocol. URLs (or Uniform Resource Locator) and URIs (Uniform Resource Identifier) are the names most people associate with web addresses.

Web browsers are the client applications our computers use to connect to the World Wide Web and access resources stored on a web server. As with most server processes, the web server runs as a background service and makes different types of files available.

In order to access the content, web clients make connections to the server and request the desired resources. The server replies with the resources and, upon receipt, the browser interpret the data and present it to the user.



```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (unix) (Red-Hat/Linux)
Last-Modified: wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
connection: close
content-Type: text/html; charset-UTF-8
<html>
<head>
<title>Cisco Systems Inc, Home Page</title>
</head>
<body>
...CONTENTS OF HTML PAGE...
</body>
```

In response to the request, the HTTP server returns code for a web page.

Browsers can interpret and present many data types, such as plain text or Hypertext Markup Language (HTML, the language in which web pages are constructed). Other types of data, however, may require another service or program, typically referred to as plug-ins or add-ons. To help the browser determine what type of file it is receiving, the server specifies what kind of data the file contains.

To better understand how the web browser and web client interact, we can examine how a web page is opened in a browser. For this example, we will use the URL: http://www.cisco.com/web-server.htm.

First, the browser interprets the three parts of the URL:

1. http (the protocol or scheme)
2. www.cisco.com (the server name)
3. web-server.htm (the specific file name requested).

The browser then checks with a name server to convert www.cisco.com into a numeric address, which it uses to connect to the server. Using the HTTP protocol requirements, the browser sends a **GET** request to the server and asks for the file web-server.htm. The server in turn sends the HTML code for this web page to the browser. Finally, the browser deciphers the HTML code and formats the page for the browser window.

HTTP

The Hypertext Transfer Protocol (HTTP), one of the protocols in the TCP/IP suite, was originally developed to publish and retrieve HTML pages and is now used for distributed, collaborative information systems. HTTP is used across the World Wide Web for data transfer and is one of the most used application protocols.

HTTP specifies a request/response protocol. When a client, typically a web browser, sends a request message to a server, the HTTP protocol defines the message types the client uses to request the web page and also the message types the server uses to respond. The three common message types are GET, POST, and PUT.

1. **GET** is a client request for data. A web browser sends the GET message to request pages from a web server
2. **POST** and **PUT** are used to send messages that upload data to the web server
3. **PUT** uploads resources or content to the web server.

Although it is remarkably flexible, HTTP is not a secure protocol. The POST messages upload information to the server in plain text that can be intercepted and read. Similarly, the server responses, typically HTML pages, are also unencrypted.

For secure communication across the Internet, the HTTP Secure (HTTPS) protocol is used for accessing or posting web server information. HTTPS can use authentication and encryption to secure data as it travels between the client and server. HTTPS specifies additional rules for passing data between the Application layer and the Transport Layer.

Web Servers

Web servers are an important part of the business plan for any organization with a presence on the Internet. Web browsers are used by consumers to access business web sites. However, web browsers are only half of the communication channel. The other half of the communication channel is web server support. Web server support is a valuable skill for network administrators. Based on a survey by Netcraft in January, 2007, the following table shows the top three web server applications by percent of use:

Web Server	Percent of use
Apache	60 %
Microsoft	31 %
Sun	1.6 %

PROCEDURE
(Modified based on CCNA's labs)

In this lab you will download, install, and configure the popular Apache web server. A web browser will be used to connect to the server, and Wireshark will be used to capture the communication.

The lab will be using the same Eagle-server Topology. If you have changed your computer, please modify your network IP address accordingly.

Part A: Download, install and verify Apache Web Server

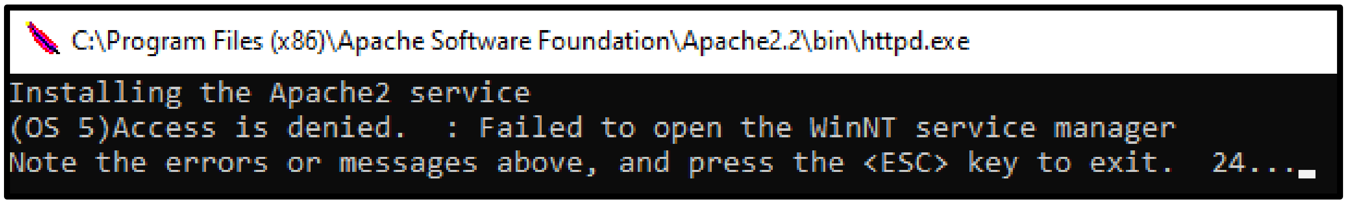
1) The lab will be using the same Eagle-server Topology Diagram as in previous labs (refer to Figure 1 for reference). If you have changed your computer, please modify your network IP address accordingly.
2) Download and install Apache web server application. 2 possible options:
   a) Static version is available at Eagle Server. You may required administration right to properly install Apache Web server. The download URL is ftp://eagle-server.example.com/pub/eagle_labs/eagle1/chapter3/.
   b) Latest version (v2.4) is available for download on internet. The subsequent procedures would appear slightly different due the changes in the version.

   Briefly describe the procedures/commands used to perform the installation.

   See next page.

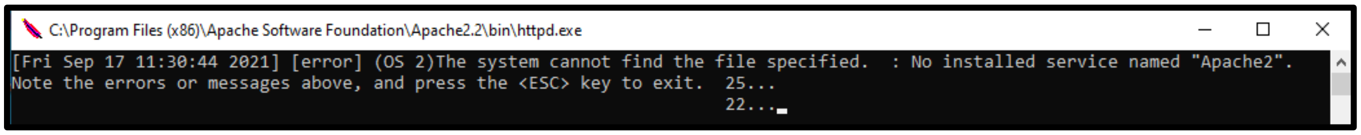
Installing Apache web server on Windows 10 without Administrator rights:

- As mentioned above, in order for Apache web server to be installed correctly, the installer application must be run with administrator rights. If it is not installed with administrator rights, then the following issues will occur:



```
C:\Program Files (x86)\Apache Software Foundation\Apache2.2\bin\httpd.exe
Installing the Apache2 service
(OS 5)Access is denied.  : Failed to open the WinNT service manager
Note the errors or messages above, and press the <ESC> key to exit.  24...
```

Figure 1. First error dialogue window pops up during installation of Apache web server application shown above. Error shows that the Apache service (the one that runs the web server on the host in the background) cannot be installed due to access restrictions.



```
C:\Program Files (x86)\Apache Software Foundation\Apache2.2\bin\httpd.exe
[Fri Sep 17 11:30:44 2021] [error] (OS 2)The system cannot find the file specified.  : No installed service named "Apache2".
Note the errors or messages above, and press the <ESC> key to exit.  25...
                                                                      22...
```

Figure 2. Second error dialogue window pops up after first error dialogue window during installation of Apache web server application shown above. Error shows that the installer cannot find the Apache service and cannot apply the base configuration for the web server.

- **These two errors seen above are critical as they will prevent the Apache web server from operating correctly on the host. This is explained below:**

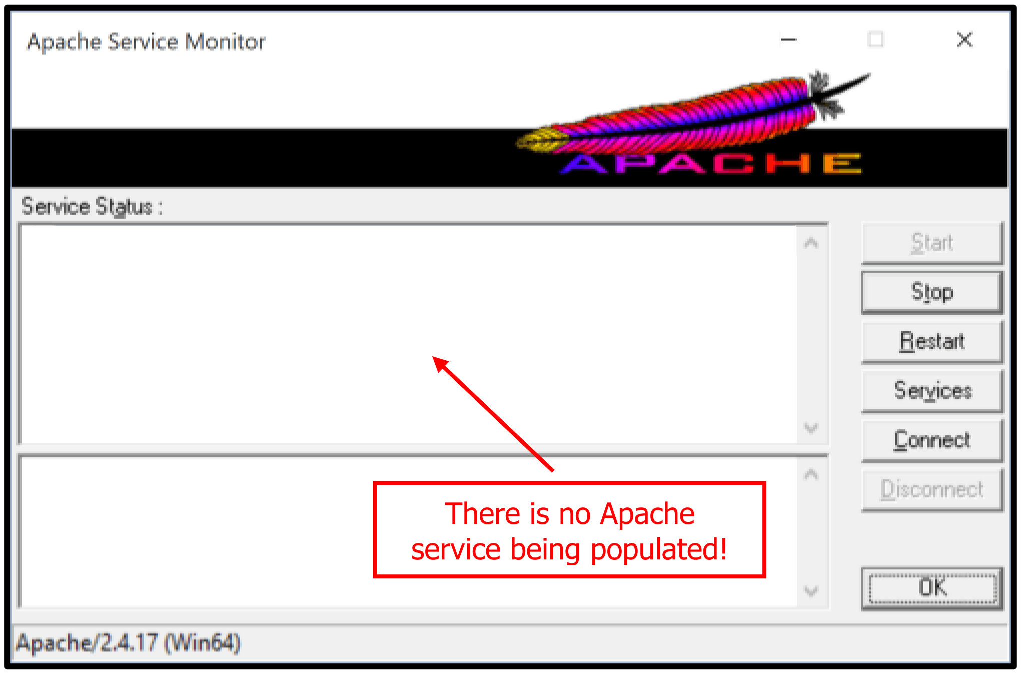
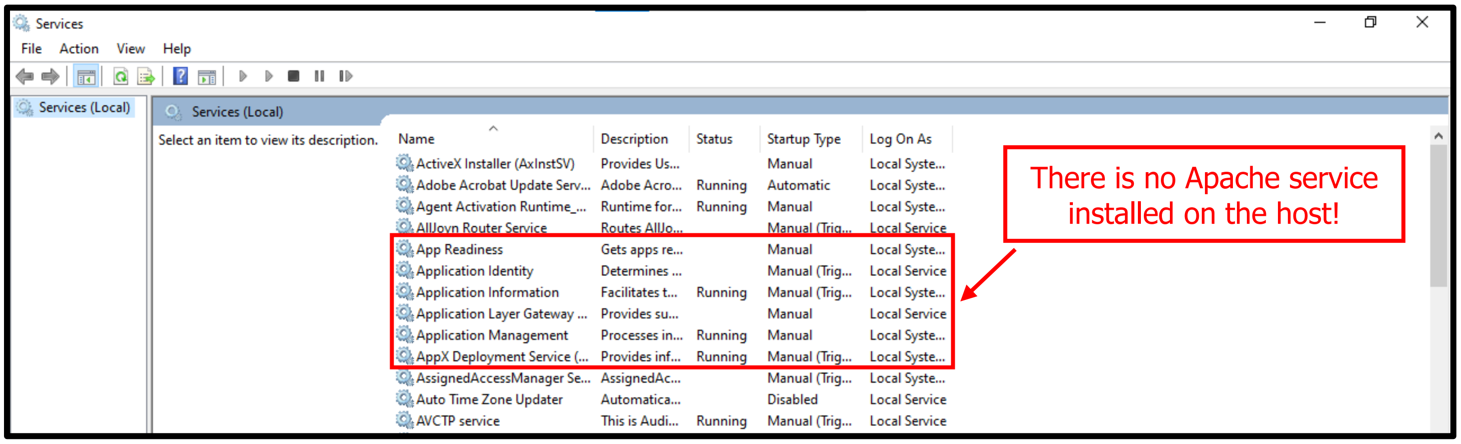


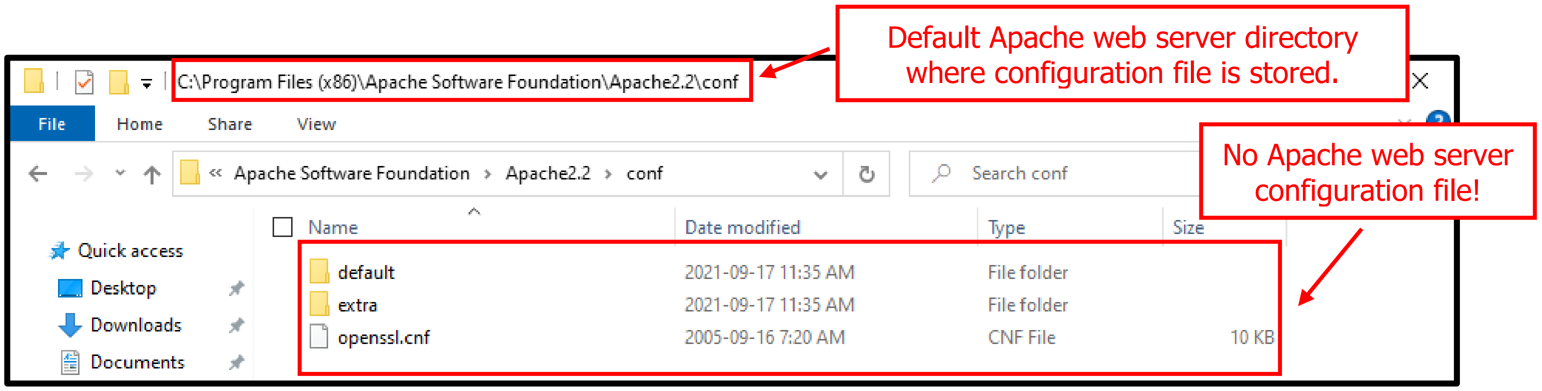
Figure 3. The two critical error dialogues shown above will close automatically within 30 seconds and the installer will close without any further errors. When checking the Apache Service Monitor for the Apache service upon application installation completion, nothing will show up (similar to what is seen in the window above). Without this service, there is essentially no web server installed/running on the host and the rest of the lab cannot be completed.



Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Manual	Local Syste...
Adobe Acrobat Update Serv...	Adobe Acro...	Running	Automatic	Local Syste...
Agent Activation Runtime_...	Runtime for...	Running	Manual	Local Syste...
AllJoyn Router Service	Routes AllJo...		Manual (Trig...	Local Service
App Readiness	Gets apps re...		Manual	Local Syste...
Application Identity	Determines ...		Manual (Trig...	Local Service
Application Information	Facilitates t...	Running	Manual (Trig...	Local Syste...
Application Layer Gateway ...	Provides su...		Manual	Local Service
Application Management	Processes in...	Running	Manual	Local Syste...
AppX Deployment Service (...	Provides inf...	Running	Manual (Trig...	Local Syste...
AssignedAccessManager Se...	AssignedAc...		Manual (Trig...	Local Syste...
Auto Time Zone Updater	Automatica...		Disabled	Local Service
AVCTP service	This is Audi...	Running	Manual (Trig...	Local Service

There is no Apache service installed on the host!

Figure 3. We know for certain that the Apache service is not installed on the host because it does not appear under the complete list of services currently installed on the host as seen in the window above (you can access the complete list of services installed on the host by typing "run.exe" in the Windows search menu and then typing "services.msc" in the run dialogue window. Click "Ok" and the services window will appear within a few seconds). As mentioned previously, without this service installed, there is essentially no web server installed/running on the host and the rest of the lab cannot be completed.


C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf
File
Home
Share
View
« Apache Software Foundation › Apache2.2 › conf
Search conf
Quick access
Desktop
Downloads
Documents

Default Apache web server directory where configuration file is stored.

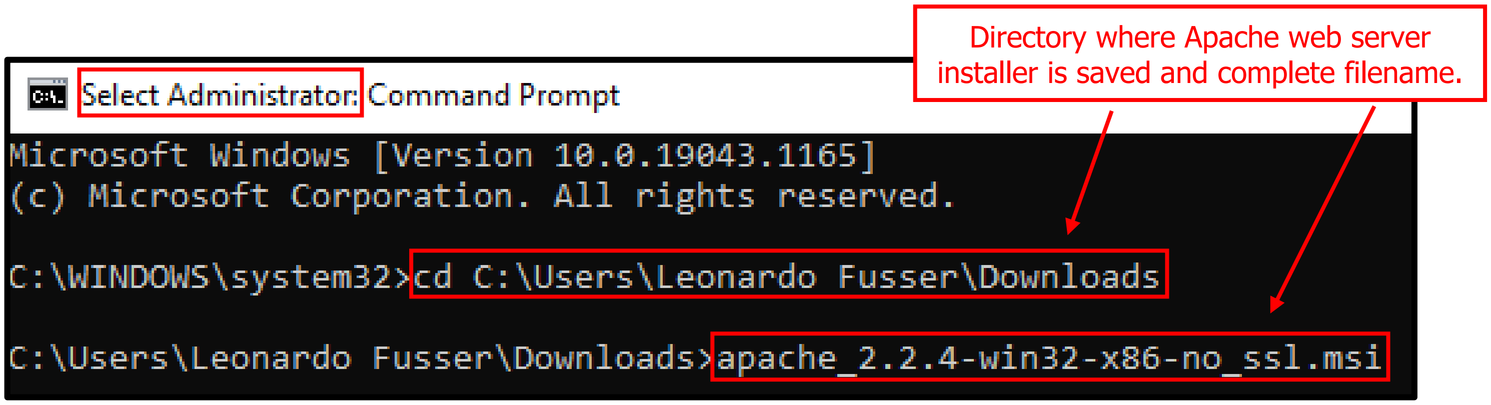
No Apache web server configuration file!

Name	Date modified	Type	Size
default	2021-09-17 11:35 AM	File folder	
extra	2021-09-17 11:35 AM	File folder	
openssl.cnf	2005-09-16 7:20 AM	CNF File	10 KB

Figure 4. Since the installer couldn't install the Apache web server application properly, it couldn't have applied the base configuration files needed for the Apache web server to operate correctly. Critical files such as the "httpd.conf" (configuration file) are missing and thus the web server cannot run as shown in the window above. The directory that stores the Apache web server configuration file is the following (assuming default directory was used during installation): C:\Program Files (x86)\Apache Software Foundation\Apachex.x\conf *(x.x is the Apache version installed).*

Installing Apache web server on Windows 10 <u>with</u> Administrator rights:

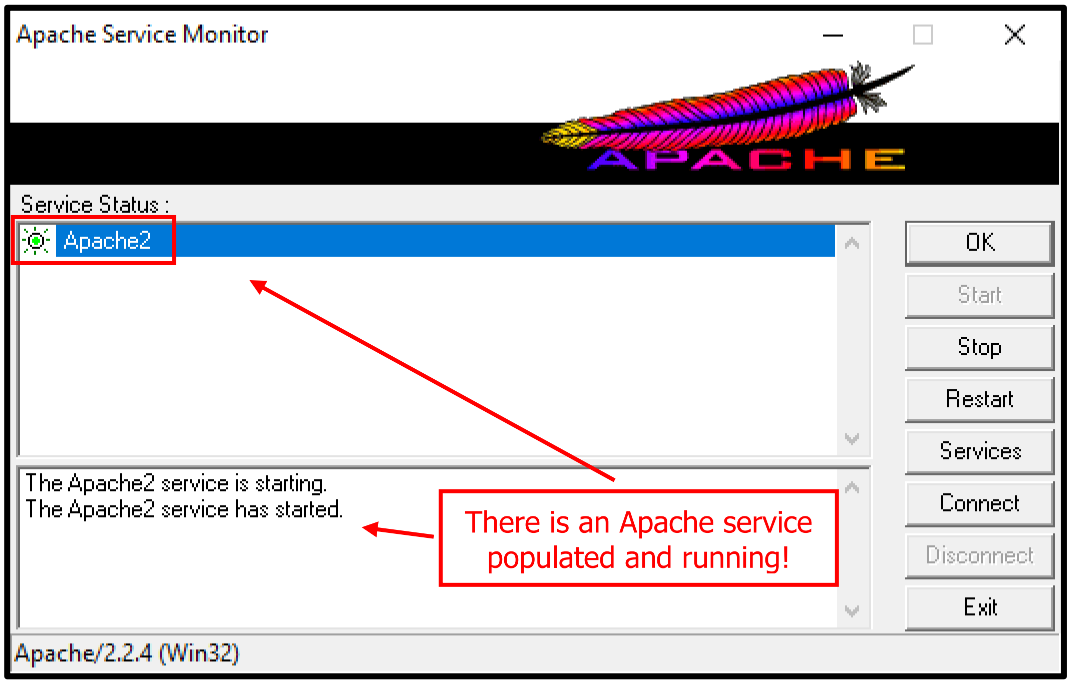
- As mentioned above, in order for Apache web server to be installed correctly, the installer application must be run with administrator rights. In order to do so, the following steps must be followed. The results are shown as well:


Select Administrator: Command Prompt
Directory where Apache web server installer is saved and complete filename.

```
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\Leonardo Fusser\Downloads

C:\Users\Leonardo Fusser\Downloads>apache_2.2.4-win32-x86-no_ssl.msi
```

Figure 5. One way of running the Apache web server installer with administrator rights can be done through the command prompt window as show above. Open command prompt as administrator (shown in red box on top left of window above), then change the directory to the one where the Apache web server installer is saved. Specify the complete name of the file with extension of the Apache web server installer and hit enter once completely typed. The Apache web server installer window will pop up shortly after, this time running with Administrator rights.

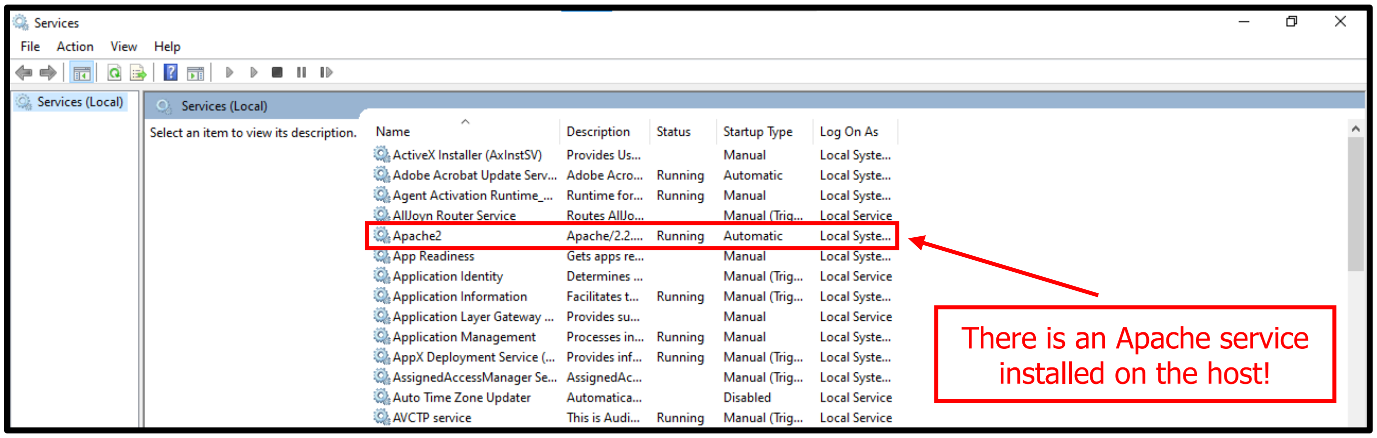
- **Assuming the installer was executed correctly, there should be no errors appearing during installation of the Apache web server. Once complete, the Apache web server will have everything that is needed to run a simple web server on the host. This is explained below:**



```
The Apache2 service is starting.
The Apache2 service has started.
```

Apache/2.2.4 (Win32)

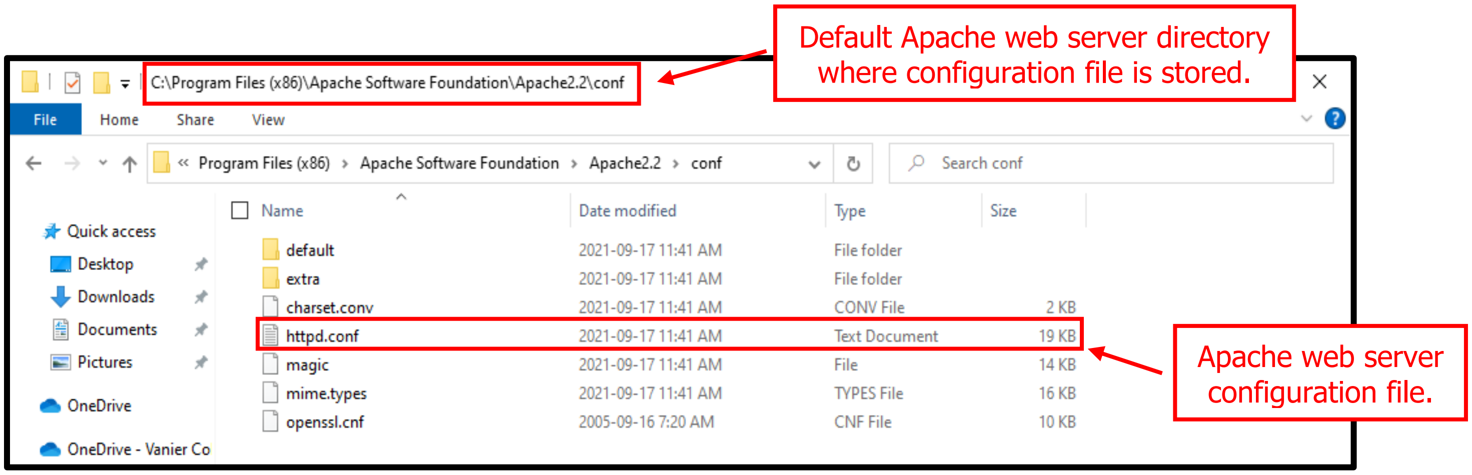
Figure 6. Assuming installer complete successfully, an Apache service should be populated in the Apache Service Monitor like what is shown in the window above. The Apache service could be then started and a simple web server will be running on the host once fully started.



Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides Us...		Manual	Local Syste...
Adobe Acrobat Update Serv...	Adobe Acro...	Running	Automatic	Local Syste...
Agent Activation Runtime_...	Runtime for...	Running	Manual	Local Syste...
AllJoyn Router Service	Routes AllJo...		Manual (Trig...	Local Service
Apache2	Apache/2.2....	Running	Automatic	Local Syste...
App Readiness	Gets apps re...		Manual	Local Syste...
Application Identity	Determines ...		Manual (Trig...	Local Service
Application Information	Facilitates t...	Running	Manual (Trig...	Local Syste...
Application Layer Gateway ...	Provides su...		Manual	Local Service
Application Management	Processes in...	Running	Manual	Local Syste...
AppX Deployment Service (...	Provides inf...	Running	Manual (Trig...	Local Syste...
AssignedAccessManager Se...	AssignedAc...		Manual (Trig...	Local Syste...
Auto Time Zone Updater	Automatica...		Disabled	Local Service
AVCTP service	This is Audi...	Running	Manual (Trig...	Local Service

There is an Apache service installed on the host!

Figure 7. We know for certain that the Apache service is installed on the host because it appears under the complete list of services currently installed on the host as seen in the window above (you can access the complete list of services installed on the host by typing "run.exe" in the Windows search menu and then typing "services.msc" in the run dialogue window. Click "Ok" and the services window will appear within a few seconds). The Apache service can be also configured here to run, stop and other features and options.


Default Apache web server directory where configuration file is stored.

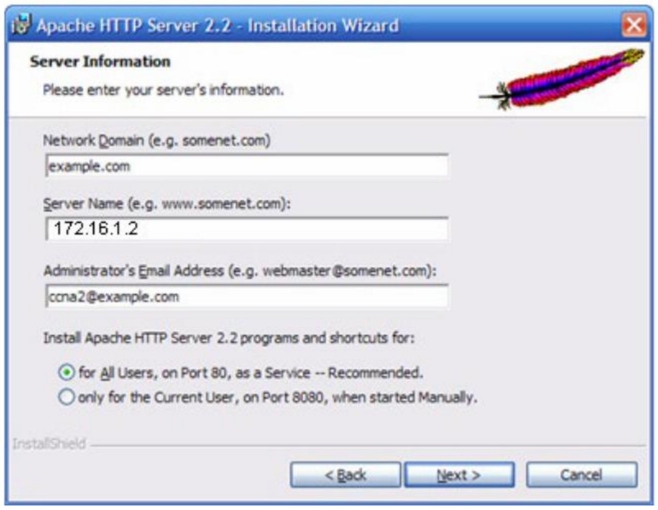
C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf

Name	Date modified	Type	Size
default	2021-09-17 11:41 AM	File folder	
extra	2021-09-17 11:41 AM	File folder	
charset.conv	2021-09-17 11:41 AM	CONV File	2 KB
httpd.conf	2021-09-17 11:41 AM	Text Document	19 KB
magic	2021-09-17 11:41 AM	File	14 KB
mime.types	2021-09-17 11:41 AM	TYPES File	16 KB
openssl.cnf	2005-09-16 7:20 AM	CNF File	10 KB

Apache web server configuration file.

Figure 8. Since the installer installed the Apache web server application properly, it also applied the base configuration files needed for the Apache web server to operate correctly. Critical files such as the "httpd.conf" (configuration file) are populated and thus the web server can run properly as shown in the window above. The directory that stores the Apache web server configuration file is the following (assuming default directory was used during installation): C:\Program Files (x86)\Apache Software Foundation\Apachex.x\conf *(x.x is the Apache version installed).*

3) Start the installation with default settings and consent to the licensing agreement. Customized configuration of the web server as shown below, where
   a) server name is the IP address of your computer, and
   b) Administrator's email is your email address


Apache HTTP Server 2.2 - Installation Wizard

Server Information

Please enter your server's information.

Network Domain (e.g. somenet.com): example.com

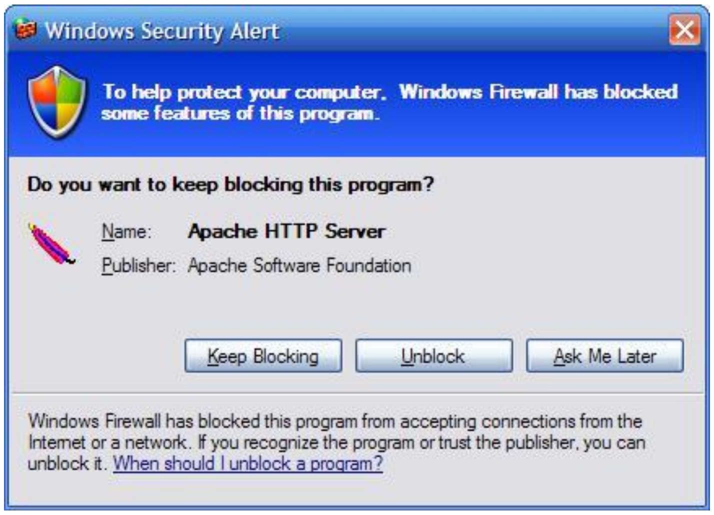
Server Name (e.g. www.somenet.com): 172.16.1.2

Administrator's Email Address (e.g. webmaster@somenet.com): ccna2@example.com

Install Apache HTTP Server 2.2 programs and shortcuts for:

[✓ for All Users, on Port 80, as a Service -- Recommended.]
only for the Current User, on Port 8080, when started Manually.

4) Accept the rest of the recommendation and default settings, and finish the installation.
5) If a Windows Security Alert is displayed, select unblock. This will permit connections to the web server.


Windows Security Alert

To help protect your computer, Windows Firewall has blocked some features of this program.

Do you want to keep blocking this program?

Name: **Apache HTTP Server**

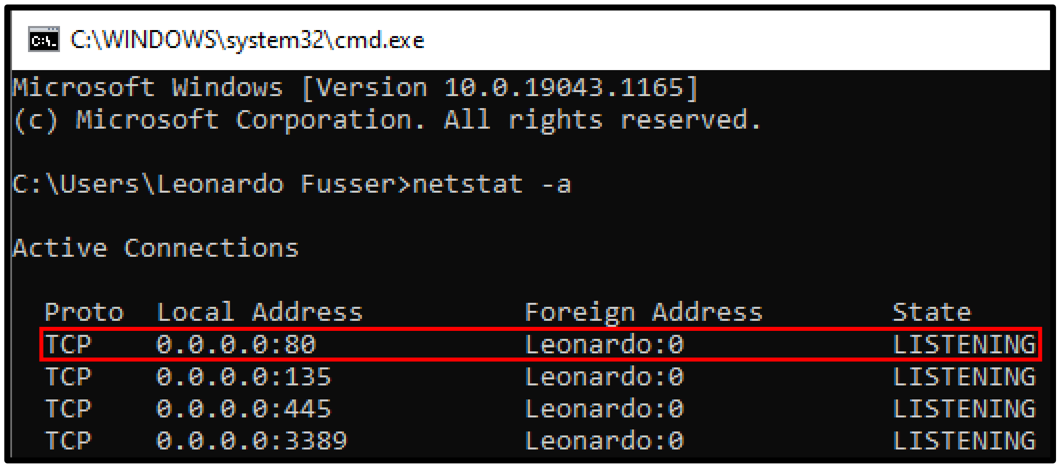
Publisher: Apache Software Foundation

Windows Firewall has blocked this program from accepting connections from the Internet or a network. If you recognize the program or trust the publisher, you can unblock it. When should I unblock a program?

6) Verify the web server.
   a) Run netstat –a from command line to discover open and connected ports on your computer. Example:

```
C:\>netstat -a
Active Connections

Proto    Local Address                  Foreign Address      State
TCP      GW-desktop-hom:http            GW-desktop-hom:0     LISTENING
TCP      GW-desktop-hom:epmap           GW-desktop-hom:0     LISTENING
TCP      GW-desktop-hom:microsoft-ds    GW-desktop-hom:0     LISTENING
TCP      GW-desktop-hom:3389            GW-desktop-hom:0     LISTENING
<output omitted>
C:\>
```

b) Using the above command, verify that the web server is operating properly on your host computer. **Perform a screen shot on your findings.**

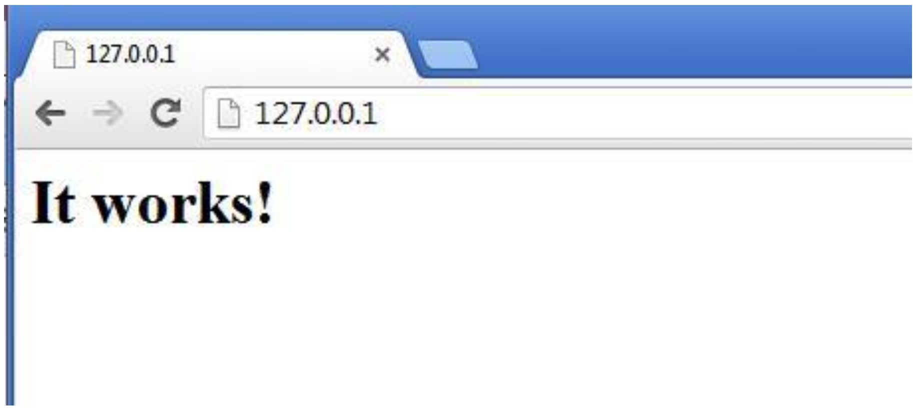


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Leonardo Fusser>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:80             Leonardo:0             LISTENING
  TCP    0.0.0.0:135            Leonardo:0             LISTENING
  TCP    0.0.0.0:445            Leonardo:0             LISTENING
  TCP    0.0.0.0:3389           Leonardo:0             LISTENING
```

Figure 9. When the Apache web server is started and running, it will appear in command prompt after netstat command is run. Since the Apache web server is configured to work on port 80 (default), it will listen to HTTP requests on that port on the host. As shown above in red box, it appears as the first entry under netstat command.

The Apache web server monitor icon should be visible on the lower right side of the screen, close to the time. If not, you may manually run it.

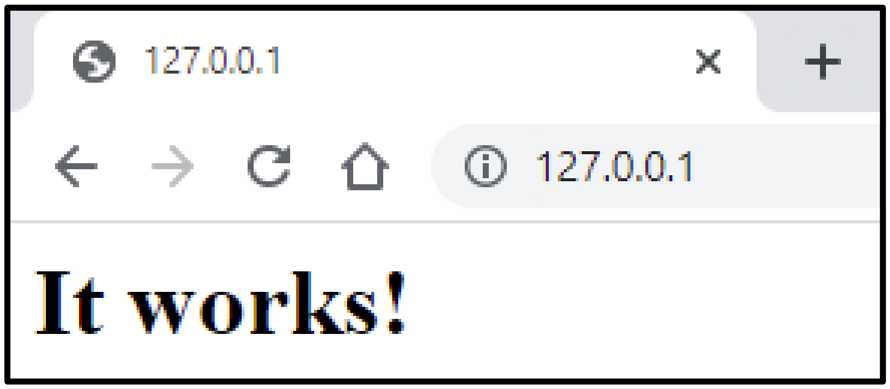
c) Open a web browser, and connect to the URL of your computer. A web page similar to Figure 2 will be displayed if the web server is working properly.
The 127.0.0.0 / 8 network address is reserved and is used for local IP addresses. The same page should be displayed if the URL is changed to the IP address on the Ethernet interface or to any host IP address in the 127.0.0.0 / 8 network range.



It works!

Figure 2 : Default web page

See next page.



It works!

Figure 10. Result shows default webpage being displayed successfully from Apache web server running on host on port 80.

d) Test the web server on several different IP addresses from 127.0.0.0/8 network range. Fill in the following table with the results:

IP Address	Status	Explanation
127.0.0.1	Works.	This is a usable host address in the 127.0.0.0/8 network range.
127.255.255.254	Works.	This is a usable host address in the 127.0.0.0/8 network range.
127.255.255.255	No.	This is a broadcast address. Not a usable host address!
127.0.0.0	No.	This is a network address. Not a usable host address!

7) Verify and modify the default Web Server Configuration File.

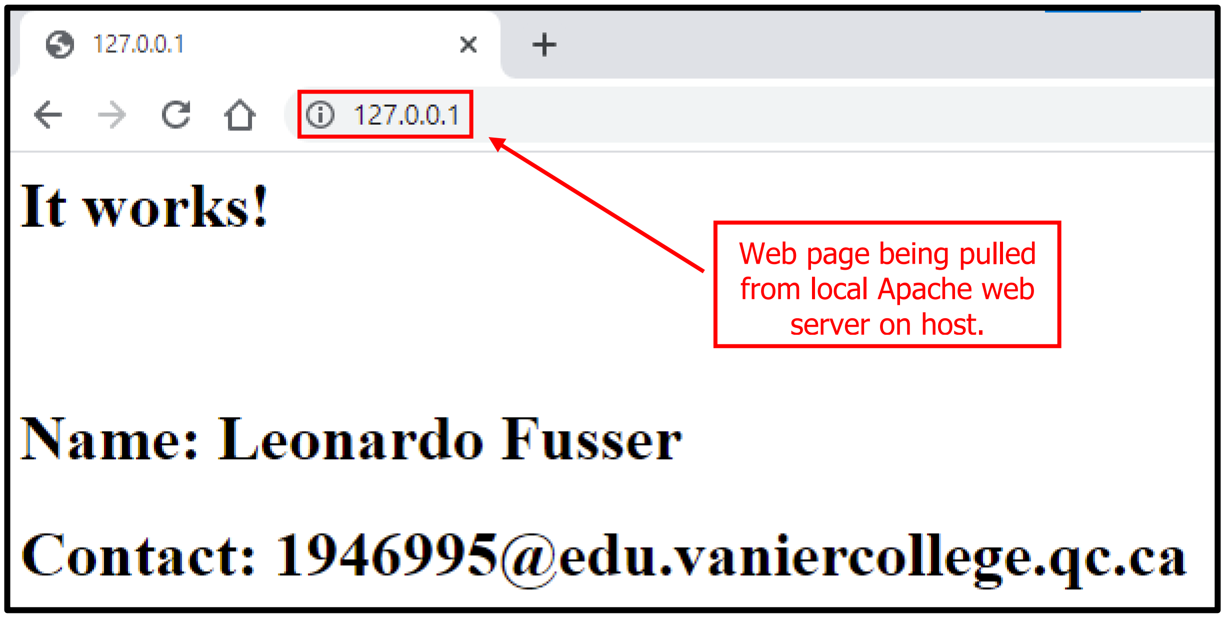
a) Open Apache web server configuration file, `httpd.conf`.

b) Numerous configuration parameters allow the Apache web server to be fully customizable. The "#" character indicates a comment for system administrators, exempt from access by the web server. Scroll down the configuration file, and verify the following settings:

Value	Meaning
#Listen 12.34.56.78:80 Listen 80	Listen on TCP port 80 for all incoming connections. To accept connections from only this host, change the line to Listen 127.0.0.1:80.
ServerAdmin ccna2@example.com	If there are problems, e-mail the web server at this e-mail address.
ServerName 172.16.1.2:80	For servers without DNS names, use the IP address:port number.
DocumentRoot "${SRVROOT}/htdocs"	This is the root directory for the web server.
<IfModule dir_module> DirectoryIndex index.html </IfModule>	DirectoryIndex sets the file that Apache will serve if a directory is requested. If no page is requested from that directory, display index.html if it is present.

c) Modify the default webpage to display something more personal such as email contact of administrator etc.

d) Save your changes and display the new default page. **Show your code and webpage in your report.**

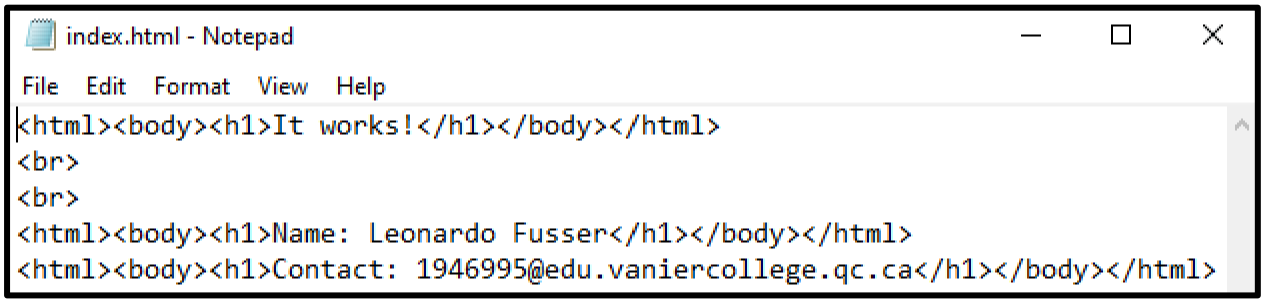


It works!

Name: Leonardo Fusser

Contact: 1946995@edu.vaniercollege.qc.ca

Figure 11. Result shows customized default webpage being displayed successfully from Apache web server running on host on port 80.



```
<html><body><h1>It works!</h1></body></html>
<br>
<br>
<html><body><h1>Name: Leonardo Fusser</h1></body></html>
<html><body><h1>Contact: 1946995@edu.vaniercollege.qc.ca</h1></body></html>
```

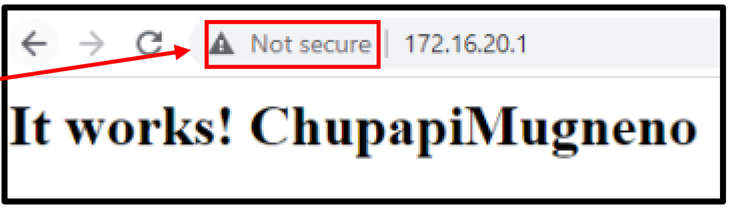
Figure 12. Complete customized HTML code for default web page on Apache web server shown above.

Part B: Capture and Analyze HTTP Traffic with Wireshark

8) Analyze HTTP traffic.

   a) Open a web browser and connect to another computer with an active webserver. **Obtain a screen shot on the webpage obtained. Why does index.html not have to be entered in the URL for the file contents to be displayed?**

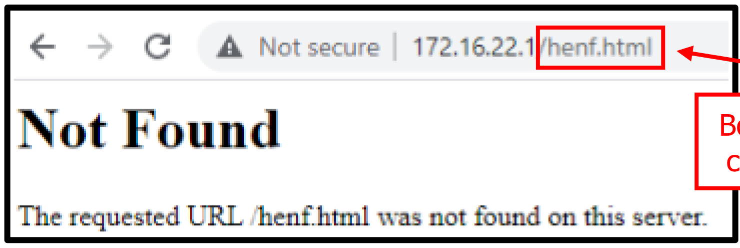
Warning from browser stating that the connection is not secure (because of HTTP).



It works! ChupapiMugneno

Figure 13. Result of web page obtained from someone else's Apache web server shown above. Since all other Apache web servers have the same base configuration, if a specific web page (HTML file) is not specified in the web browser (when request to the web server does not contain a demanded HTML file), the Apache web server will respond back with a default web page for the browser to display (in this case, it is the "index.html" file, located somewhere in the default directory where Apache web server is stored on the host). Looking back to the previous step (Figure 11), this is the same file where a customized message was added.

   b) Start Wireshark, and set the capture interface to the interface bound to the 172.16 network. Open a web browser. Deliberately enter a web page that is not on the web server. Note that an error message is displayed in the web browser. **Obtain a screen shot on the webpage obtained. What do you observe from the HTTP capture that is related to this error, especially the line-based text data content of html?**

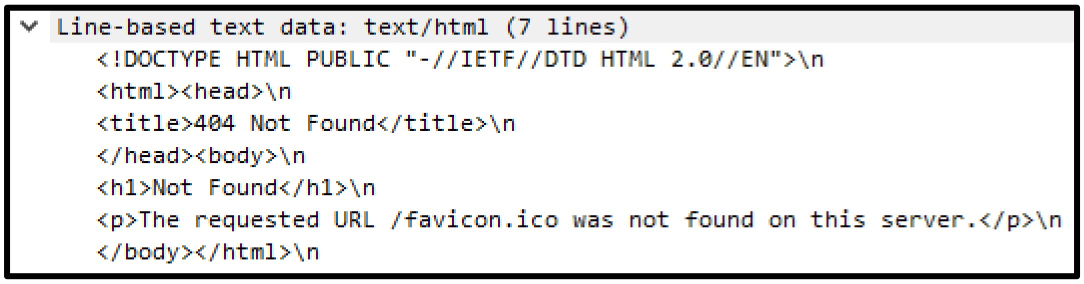


Not Found

The requested URL /henf.html was not found on this server.

Bogus file being requested from client (browser) to web server.

Figure 14. Result of requesting a non-existent web page (HTML file) on another Apache web server shown above. Results from Wireshark capture are shown below on next page.

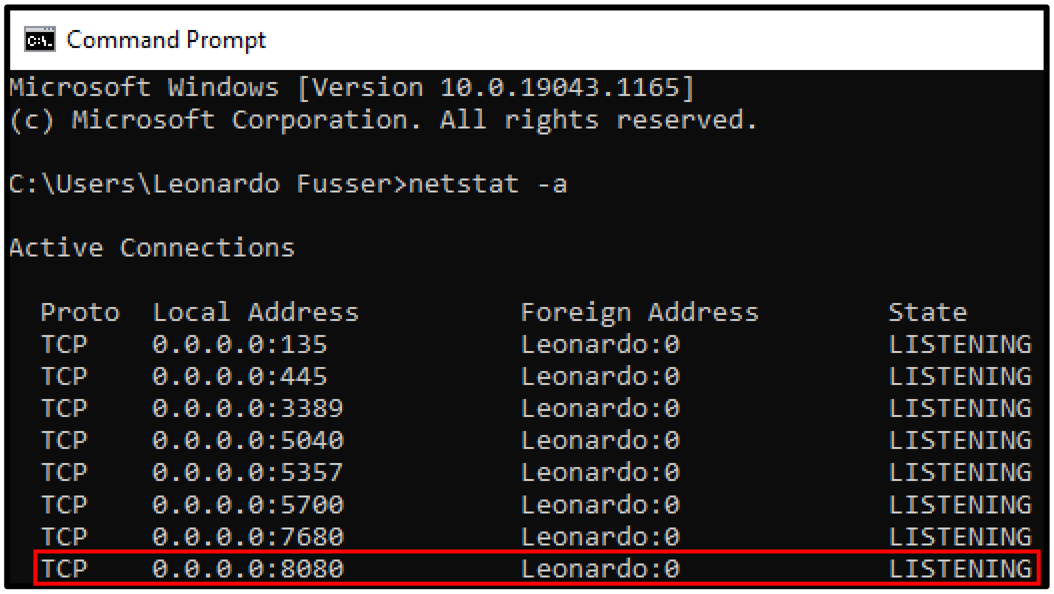


```
Line-based text data: text/html (7 lines)
    <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n
    <html><head>\n
    <title>404 Not Found</title>\n
    </head><body>\n
    <h1>Not Found</h1>\n
    <p>The requested URL /favicon.ico was not found on this server.</p>\n
    </body></html>\n
```

Figure 15. Corresponding Wireshark packet capture result for HTTP 404 error from "Figure 14" shown above. The major flaw here is that the Wireshark packet capture is able to show the complete web page content in HTML format. This confirms what was mentioned previously in this document that HTTP is not secure since flaws like this are able to occur (anyone with a network analyzer would be able to see this -> affected user(s) could be compromised because of this flaw!). HTTPS would be the more secure way of communicating with a web server since it is harder to tell what is being transmitted (data is encrypted).

9) Modify the default web server configuration file `httpd.conf` and change the `Listen 80` line to `Listen 8080`. Restart Apache service. Open a web browser and access URL `http://127.0.0.1:8080`. **How could you verify that new web server TCP port is 8080? Show a screen shot of your verification.**

*** You may need to verify that the port 8080 is unused prior to setting it in the conf file. If it is used, then choose another unused port for this step.*



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Leonardo Fusser>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            Leonardo:0             LISTENING
  TCP    0.0.0.0:445            Leonardo:0             LISTENING
  TCP    0.0.0.0:3389           Leonardo:0             LISTENING
  TCP    0.0.0.0:5040           Leonardo:0             LISTENING
  TCP    0.0.0.0:5357           Leonardo:0             LISTENING
  TCP    0.0.0.0:5700           Leonardo:0             LISTENING
  TCP    0.0.0.0:7680           Leonardo:0             LISTENING
  TCP    0.0.0.0:8080           Leonardo:0             LISTENING
```

Figure 16. Similar to what is shown in "Figure 9", to verify that the Apache web server is listening to HTTP requests on the specified port (8080 in this case) configured in the "httpd.conf" file for the Apache web server, netstat command is used in command prompt. As shown above in red box, it appears at the bottom entry under netstat command. This result shows that the Apache web server is indeed listening to HTTP requests on port 8080 (specified in "httpd.conf" file).

Discussion:

- The outcome of the lab was a success but a few hiccups were encountered in the beginning of the lab. The first mistake occurred during the Apache web server installation where the installer was not run with administrative privileges. This caused the installer to not be able to install some components and not configure the web server correctly for default use. This is explained in more detail under "Part A: Q2b)" of this document. Eventually, the solution was to force the installer to run with administrative privileges through the command prompt (explained in same section) and this time around the installer was able to install all components and configure default files for default use of the web server. A quick check was done using the "netstat" command in command prompt in Windows 10 to verify that the Apache service was running and listening to HTTP requests (by default, upon completion of the Apache web server installer, the Apache service is automatically started). Once that was done, a second test to ensure that the web server was indeed up and running was to pull up a web page from the web server. This yielded successful (message appeared saying "It works!") which meant that the Apache web server was installed correctly and was running correctly. The next step was to modify the default web page (HTML file) to add some custom text to the web page being displayed on the browser. When it was complete, the web page file ("index.html") was saved and the browser reloaded the same previous webpage, this time showing the additions that were made to the file. Moving forward, a simple test was done to see if the browser was able to display the same customized web page from other hosts who had the same web server running (on port 80 as well), and the result yielded successful when the browser showed the other host's customized web pages. Another test was done to see what would occur if a bogus web page was requested from the browser (client) to one of the other host's web server. The result showed that despite being able to connect to the specified web server, the browser was not able to retrieve the requested web page since it didn't exist on the host's web server. At the same time these tests were occurring, a Wireshark capture was in session analyzing all this behaviour (between client and web server). Similar to the previous lab, the results show that all behaviour is very clear, since private information, such as the contents of the web pages being requested from client to web server, were shown in the Wireshark capture. This was because HTTP was being used, and none of the data is encrypted, leaving many flaws behind.

Conclusion:

- Successfully configured a web server application on host (in this case, Apache web server).
- Successfully viewed and modified web server configuration file on host.
- Successfully viewed other web server web pages on other hosts.
- Successfully interpreted and verified web server functionality using netstat command.
- Successfully performed thorough analysis of Wireshark capture pertaining to web server activities.