WAPH - Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Student

Name : Vijaykumar Gandi

Email : gandivr@mail.uc.edu

Profile Pic :

Repository information for Lab-1

Repository URL : https://github.com/gandivr/waph-gandivr/tree/main/labs/lab1

Lab-1 : FOundations of the Web

This Lab Covers more on the Foundations of the web , triggering more on HTTP Request , Response using WIreshark and telnet and basic web programming. It also covers the aspects of security in different ways.

Part I: Web and hTTP Protocol

In part-1 I came across the web and the HTTP Protocol, where I went through two sub tasks mentioned below.

Task 1: Installing and Configuring Wireshark

Installation:

- I have Installed Wireshark using the command: sudo apt install wireshark-qt.And Launched Wireshark from the terminal with elevated privileges: sudo wireshark &.

Configuration:

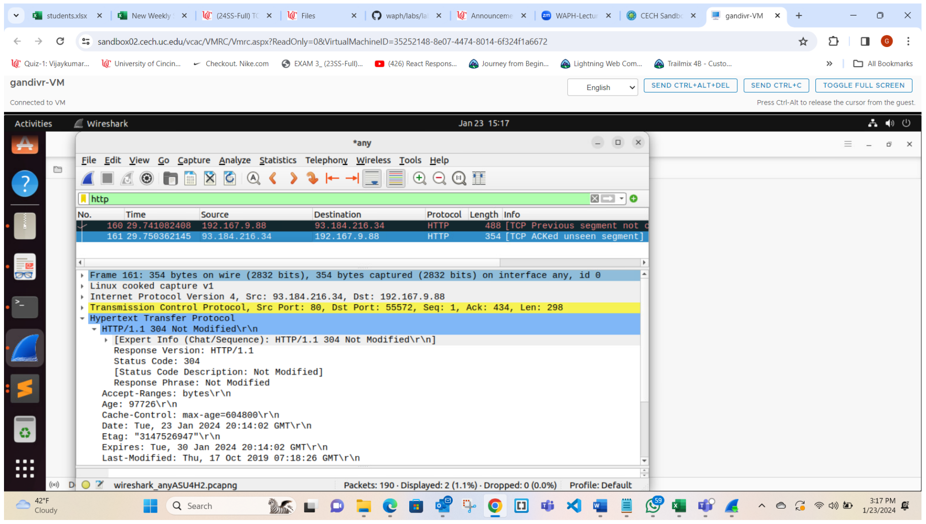
- I have Configured Wireshark to capture data from all network interfaces.Initiated packet capture by clicking on the Start icon in Wireshark.

Browsing example.com:

Firstly, I have Accessed example.com in the browser and then Halted the packet capture by clicking on the Stop icon in Wireshark. Then we have to apply the filter and I have Applied a filter to focus on HTTP results, where I Explored detailed request and response messages by clicking on them. Utilized Follow > HTTP Stream to visualize the stream of request and response messages.

Screenshots:

Request message:



```
http
No.  Time         Source         Destination    Protocol Length Info
 160 29.741082408 192.167.9.88   93.184.216.34  HTTP     488 [TCP Previous segment not c
 161 29.750362145 93.184.216.34  192.167.9.88   HTTP     354 [TCP ACKed unseen segment]

Frame 161: 354 bytes on wire (2832 bits), 354 bytes captured (2832 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 93.184.216.34, Dst: 192.167.9.88
Transmission Control Protocol, Src Port: 80, Dst Port: 55572, Seq: 1, Ack: 434, Len: 298
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]
    Response Version: HTTP/1.1
    Status Code: 304
    [Status Code Description: Not Modified]
    Response Phrase: Not Modified
  Accept-Ranges: bytes\r\n
  Age: 97726\r\n
  Cache-Control: max-age=604800\r\n
  Date: Tue, 23 Jan 2024 20:14:02 GMT\r\n
  Etag: "3147526947"\r\n
  Expires: Tue, 30 Jan 2024 20:14:02 GMT\r\n
  Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT\r\n

wireshark_anyASU4H2.pcapng    Packets: 190 · Displayed: 2 (1.1%) · Dropped: 0 (0.0%)    Profile: Default
```

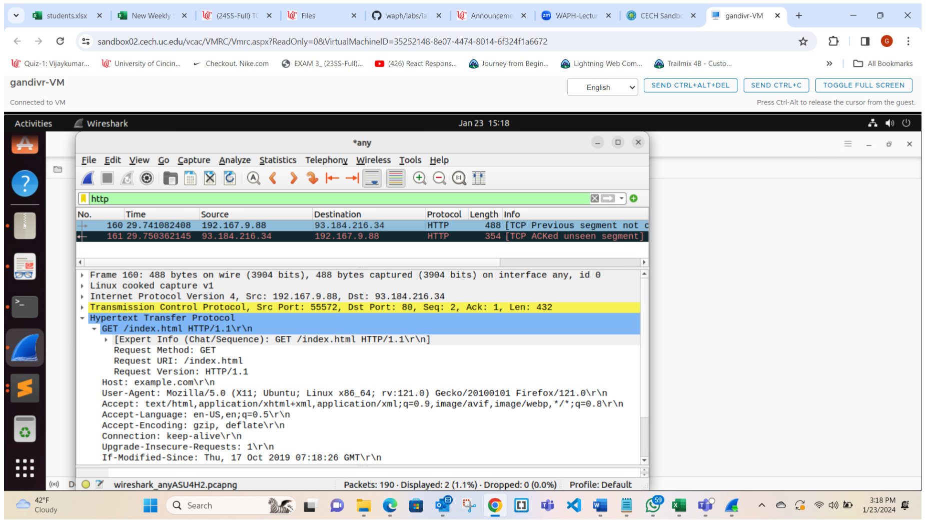
Figure 1: Screen-1

Response message:

Stream view:

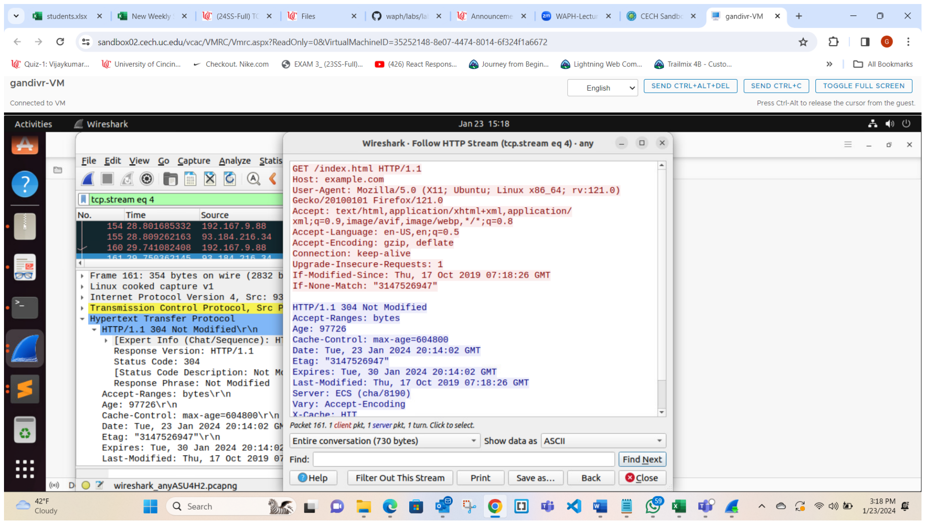
Task 2: Telnet Connection and Analysis

- Coming to task-2 I got a good understanding of HTTP using telnet and Wireshark.We must start with the Telnet Connection. Firstly, I have Started packet capture in Wireshark. Then I have Established a connection to the web server at port 80 using the command: telnet example.com 80.Then after connecting I sent an HTTP request in the terminal by using the snippet below.



```
Filter: http

No.  Time          Source         Destination    Protocol Length Info
160 29.741082408  192.167.9.88   93.184.216.34  HTTP     488 [TCP Previous segment not c
161 29.750362145  93.184.216.34  192.167.9.88   HTTP     354 [TCP ACKed unseen segment]

▸ Frame 160: 488 bytes on wire (3904 bits), 488 bytes captured (3904 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 192.167.9.88, Dst: 93.184.216.34
▸ Transmission Control Protocol, Src Port: 55572, Dst Port: 80, Seq: 2, Ack: 1, Len: 432
▾ Hypertext Transfer Protocol
  ▾ GET /index.html HTTP/1.1\r\n
    ▸ [Expert Info (Chat/Sequence): GET /index.html HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /index.html
      Request Version: HTTP/1.1
    Host: example.com\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:121.0) Gecko/20100101 Firefox/121.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    If-Modified-Since: Thu, 17 Oct 2019 07:18:26 GMT\r\n

wireshark_anyASU4H2.pcapng    Packets: 190 · Displayed: 2 (1.1%) · Dropped: 0 (0.0%)    Profile: Default
```

Figure 2: Screen-2



```
Filter: tcp.stream eq 4

No.  Time          Source
154 28.801685332  192.167.9.88
155 28.809262163  93.184.216.34
160 29.741082408  192.167.9.88
161 29.750362145  93.184.216.34

▸ Frame 161: 354 bytes on wire (2832 b
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 93
▸ Transmission Control Protocol, Src P
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 304 Not Modified\r\n
    ▸ [Expert Info (Chat/Sequence): HT
      Response Version: HTTP/1.1
      Status Code: 304
      [Status Code Description: Not Mo
      Response Phrase: Not Modified
    Accept-Ranges: bytes\r\n
    Age: 97726\r\n
    Cache-Control: max-age=604800\r\n
    Date: Tue, 23 Jan 2024 20:14:02 GM
    Etag: "3147526947"\r\n
    Expires: Tue, 30 Jan 2024 20:14:02
    Last-Modified: Thu, 17 Oct 2019 07
```

```
Wireshark · Follow HTTP Stream (tcp.stream eq 4) · any

GET /index.html HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:121.0)
Gecko/20100101 Firefox/121.0
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Thu, 17 Oct 2019 07:18:26 GMT
If-None-Match: "3147526947"

HTTP/1.1 304 Not Modified
Accept-Ranges: bytes
Age: 97726
Cache-Control: max-age=604800
Date: Tue, 23 Jan 2024 20:14:02 GMT
Etag: "3147526947"
Expires: Tue, 30 Jan 2024 20:14:02 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (cha/8190)
Vary: Accept-Encoding
X-Cache: HIT

Packet 161. 1 client pkt, 1 server pkt, 1 turn. Click to select.
Entire conversation (730 bytes)    Show data as ASCII
```

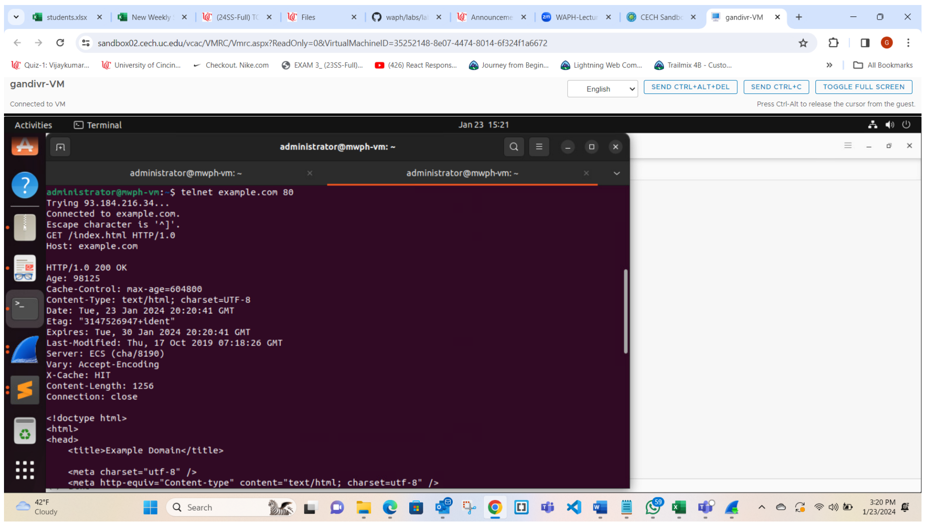
Figure 3: Screen-3

```
GET /index.html HTTP/1.0
Host: example.com
```

- We have to stop the Wireshark and observe the request and response slots in both Wireshark and terminal, then I followed the same and Terminated Wireshark capture and examined HTTP request and response messages.

Screenshots:

Terminal Screenshot (HTTP request and response):



```
administrator@mwph-vm:~$ telnet example.com 80
Trying 93.184.216.34...
Connected to example.com.
Escape character is '^]'.
GET /index.html HTTP/1.0
Host: example.com

HTTP/1.0 200 OK
Age: 98125
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Tue, 23 Jan 2024 20:20:41 GMT
Etag: "3147526947+ident"
Expires: Tue, 30 Jan 2024 20:20:41 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (cha/8190)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256
Connection: close

<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
```

Figure 4: Screen-4

HTTP request in Wireshark:

HTTP response in Wireshark:

Part II - Basic Web Application Programming:

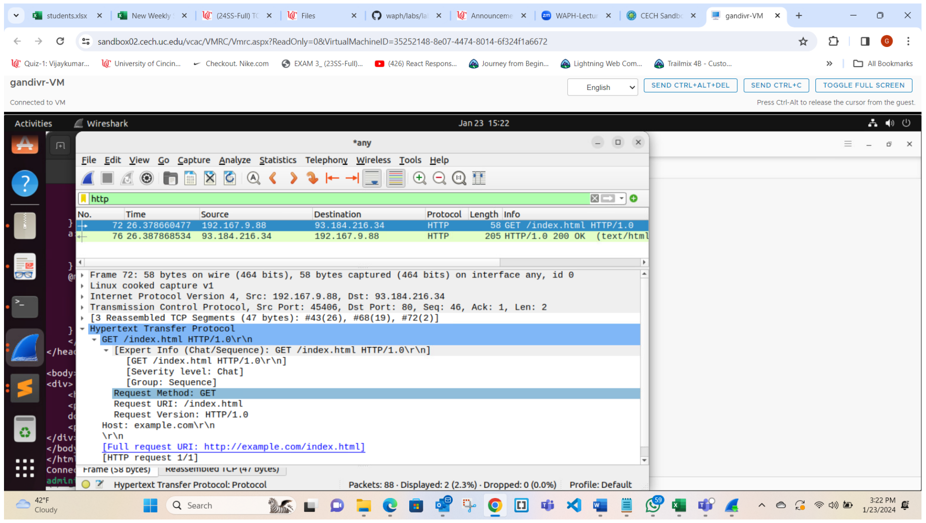
Task 1: CGI web applications in C.

HelloWorld CGI Program:

I have Created and compiled a helloworld.c program. Then Enabled CGI daemon, restarted Apache server where I Copied helloworld.cgi to /usr/lib/cgi-bin. I have Accessed http://localhost/cgi-bin/helloworld.cgi in the browser.

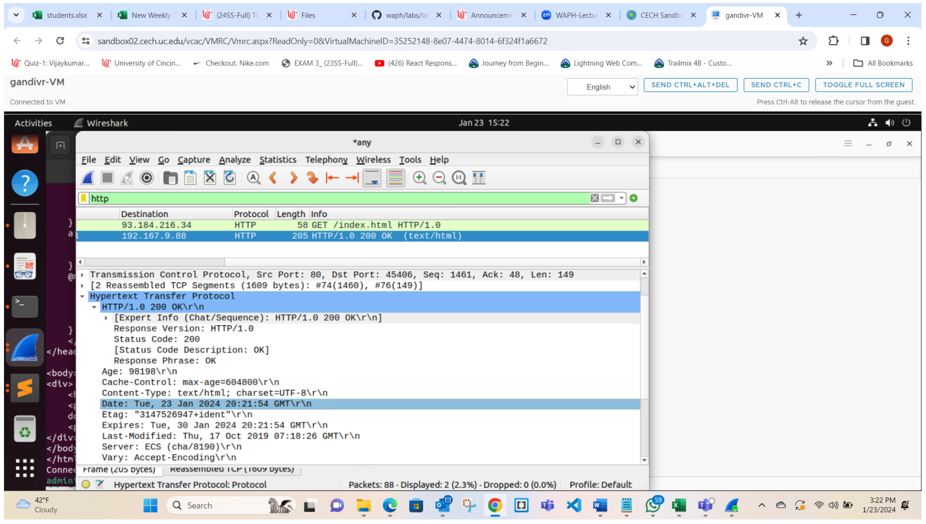
Observed output:

Program Output:



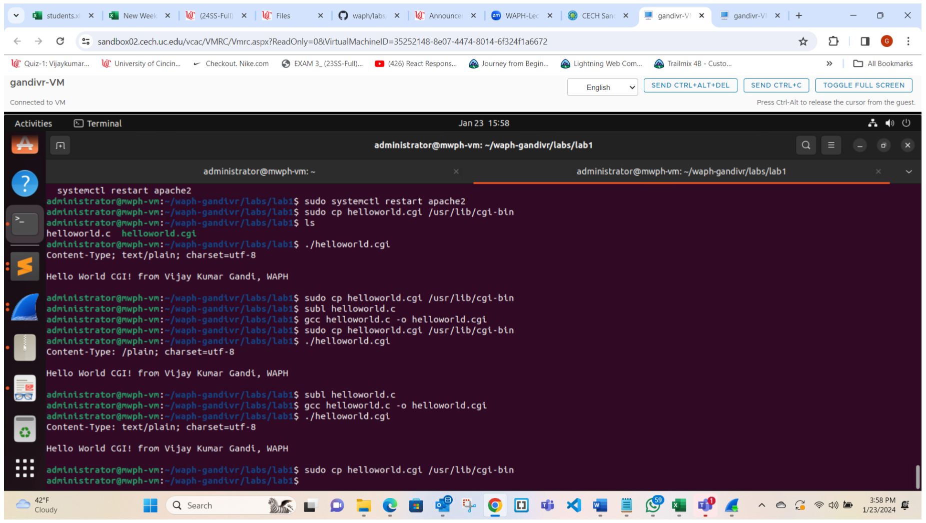
```
http
No.   Time           Source          Destination     Protocol Length Info
   72 26.378660477   192.167.9.88    93.184.216.34   HTTP       58 GET /index.html HTTP/1.0
   76 26.387868534   93.184.216.34   192.167.9.88    HTTP      205 HTTP/1.0 200 OK  (text/html

Frame 72: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.167.9.88, Dst: 93.184.216.34
Transmission Control Protocol, Src Port: 45406, Dst Port: 80, Seq: 46, Ack: 1, Len: 2
[3 Reassembled TCP Segments (47 bytes): #43(26), #68(19), #72(2)]
Hypertext Transfer Protocol
  GET /index.html HTTP/1.0\r\n
    [Expert Info (Chat/Sequence): GET /index.html HTTP/1.0\r\n]
      [GET /index.html HTTP/1.0\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /index.html
    Request Version: HTTP/1.0
  Host: example.com\r\n
  \r\n
  [Full request URI: http://example.com/index.html]
  [HTTP request 1/1]

Hypertext Transfer Protocol: Protocol    Packets: 88 · Displayed: 2 (2.3%) · Dropped: 0 (0.0%)    Profile: Default
```

Figure 5: Screen-5



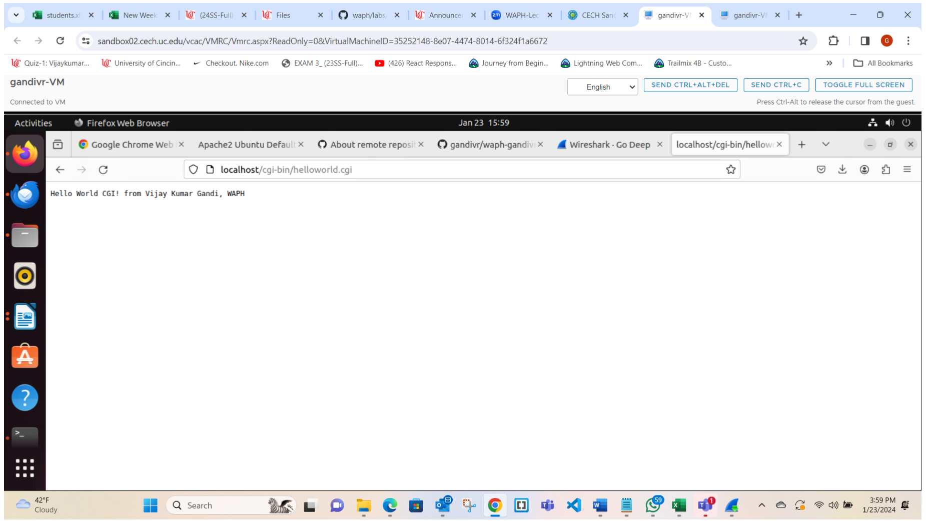
```
http
Destination      Protocol Length Info
93.184.216.34    HTTP       58 GET /index.html HTTP/1.0
192.167.9.88     HTTP      205 HTTP/1.0 200 OK  (text/html)

Transmission Control Protocol, Src Port: 80, Dst Port: 45406, Seq: 1461, Ack: 48, Len: 149
[2 Reassembled TCP Segments (1609 bytes): #74(1460), #76(149)]
Hypertext Transfer Protocol
  HTTP/1.0 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.0 200 OK\r\n]
    Response Version: HTTP/1.0
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Age: 98198\r\n
  Cache-Control: max-age=604800\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  Date: Tue, 23 Jan 2024 20:21:54 GMT\r\n
  Etag: "3147526947+ident"\r\n
  Expires: Tue, 30 Jan 2024 20:21:54 GMT\r\n
  Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT\r\n
  Server: ECS (cha/8190)\r\n
  Vary: Accept-Encoding\r\n

Hypertext Transfer Protocol: Protocol    Packets: 88 · Displayed: 2 (2.3%) · Dropped: 0 (0.0%)    Profile: Default
```

Figure 6: Screen-6



```
 systemctl restart apache2
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo systemctl restart apache2
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo cp helloworld.cgi /usr/lib/cgi-bin
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ ls
helloworld.c  helloworld.cgi
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ ./helloworld.cgi
Content-Type; text/plain; charset=utf-8

Hello World CGI! from Vijay Kumar Gandi, WAPH

administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo cp helloworld.cgi /usr/lib/cgi-bin
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ subl helloworld.c
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ gcc helloworld.c -o helloworld.cgi
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo cp helloworld.cgi /usr/lib/cgi-bin
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ ./helloworld.cgi
Content-Type: /plain; charset=utf-8

Hello World CGI! from Vijay Kumar Gandi, WAPH

administrator@mwph-vm:~/waph-gandivr/labs/lab1$ subl helloworld.c
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ gcc helloworld.c -o helloworld.cgi
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ ./helloworld.cgi
Content-Type: text/plain; charset=utf-8

Hello World CGI! from Vijay Kumar Gandi, WAPH

administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo cp helloworld.cgi /usr/lib/cgi-bin
administrator@mwph-vm:~/waph-gandivr/labs/lab1$
```

Figure 7: Screen-7



localhost/cgi-bin/helloworld.cgi

```
Hello World CGI! from Vijay Kumar Gandi, WAPH
```

Figure 8: Screen-8

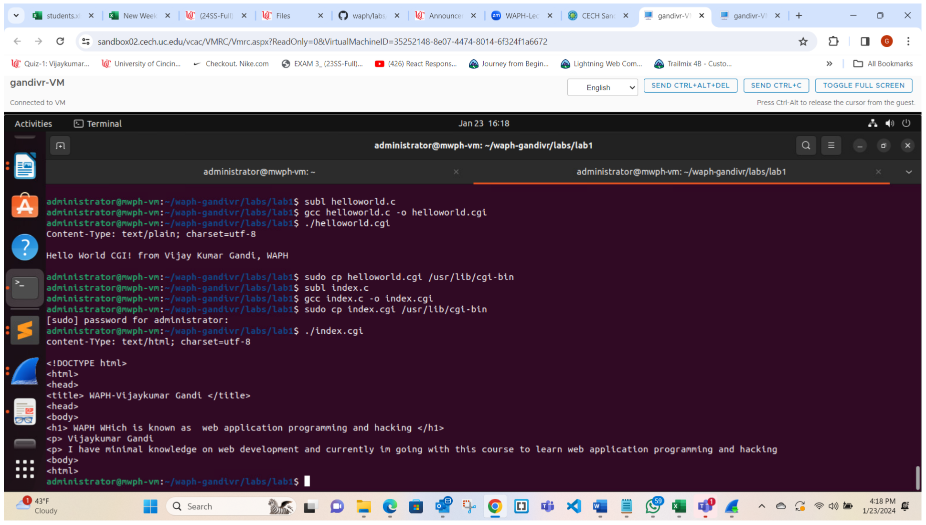
Index CGI Program:

And then I have Created and compiled another c program (index.c). Copied index.cgi to /usr/lib/cgi-bin. Then I have Accessed http://localhost/cgi-bin/index.cgi in the browser.

Below is the Program Source Code.

```
#include <stdio.h>
int main(void) {
printf("content-TYpe: text/html; charset=utf-8\n\n");
printf("<!DOCTYPE html>\n");
printf("<html>\n");
printf("<head>\n");
printf("<title> WAPH-Vijaykumar Gandi </title>\n");
printf("<head>\n");
printf("<body>\n");
printf("<h1> WAPH WHich is known as  web application programming and hacking </h1>\n");
printf("<p> Vijaykumar Gandi\n");
printf("<p> I have minimal knowledge on web development and currently im going with this cou
printf("<body>\n");
printf("<html>\n");
return 0;

}
```

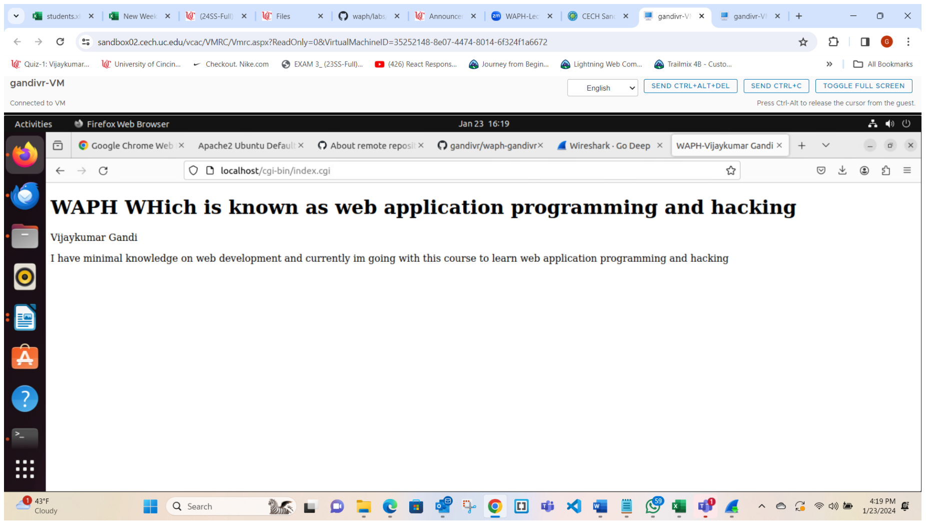
Observed output:



```
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ subl helloworld.c
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ gcc helloworld.c -o helloworld.cgi
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ ./helloworld.cgi
Content-Type: text/plain; charset=utf-8

Hello World CGI! from Vijay Kumar Gandi, WAPH

administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo cp helloworld.cgi /usr/lib/cgi-bin
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ subl index.c
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ gcc index.c -o index.cgi
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo cp index.cgi /usr/lib/cgi-bin
[sudo] password for administrator:
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ ./index.cgi
content-TYpe: text/html; charset=utf-8

<!DOCTYPE html>
<html>
<head>
<title> WAPH-Vijaykumar Gandi </title>
<head>
<body>
<h1> WAPH WHich is known as  web application programming and hacking </h1>
<p> Vijaykumar Gandi
<p> I have minimal knowledge on web development and currently im going with this course to learn web application programming and hacking
<body>
<html>
administrator@mwph-vm:~/waph-gandivr/labs/lab1$
```

Figure 9: Screen-10

Output:



WAPH WHich is known as web application programming and hacking

Vijaykumar Gandi

I have minimal knowledge on web development and currently im going with this course to learn web application programming and hacking

Figure 10: Screen-11

Task 2: PHP Programs

HelloWorld PHP Program:

I have Installed PHP and the PHP Apache module. And then Created helloworld.php and copied it to /var/www/html. And then I Accessed http://localhost/helloworld.php in the browser.

Observed output:

OUtput:

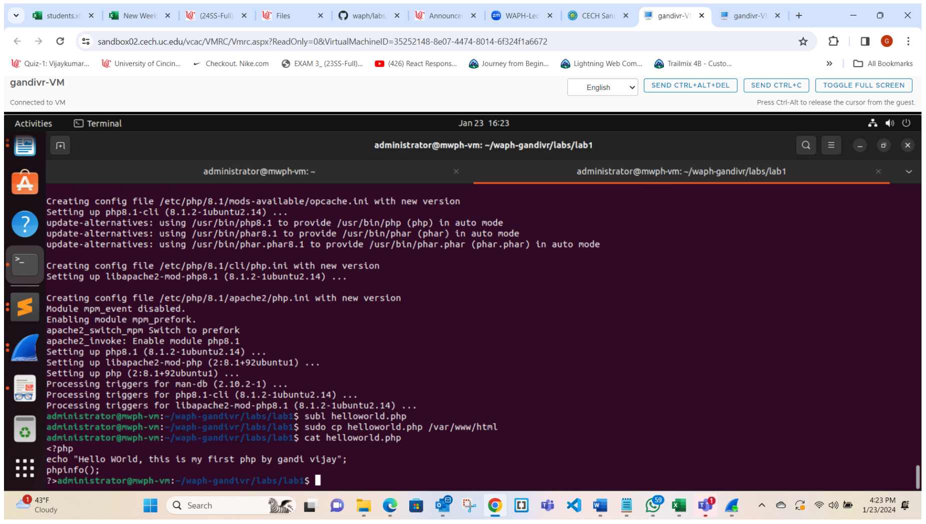
PHP with URL Parameters:

I have Created echo.php with PHP code. And then I have Accessed http://localhost/echo.php?data=HelloWorld in the browser.

Observed output:

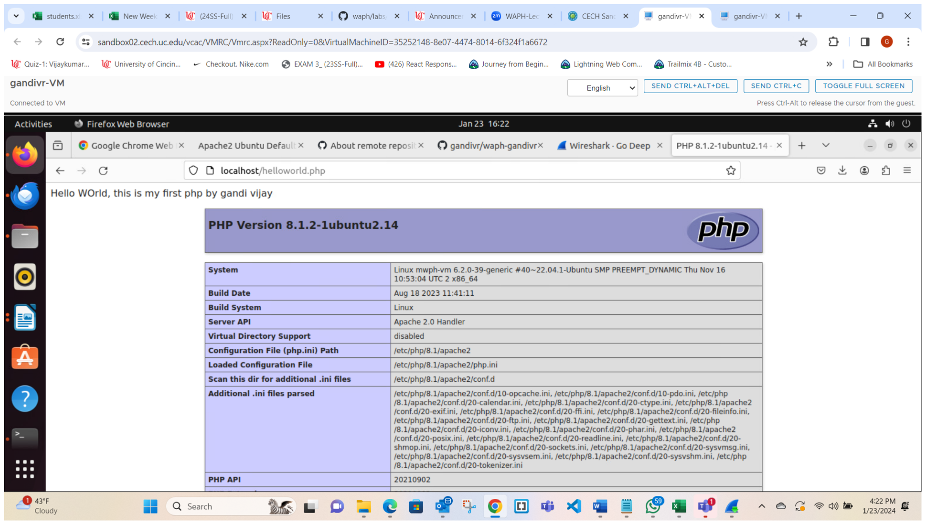
Security Risks:

- Any user-supplied data is immediately returned by the code, which, if improperly handled, can reveal sensitive information. Passwords, internal system information, and other private information may fall under this category. Unexpected input may result from the 'data' parameter not being validated. Processing unexpected or improper data could come from this.



```
Creating config file /etc/php/8.1/mods-available/opcache.ini with new version
Setting up php8.1-cli (8.1.2-1ubuntu2.14) ...
update-alternatives: using /usr/bin/php8.1 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar8.1 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar8.1 to provide /usr/bin/phar.phar (phar.phar) in auto mode

Creating config file /etc/php/8.1/cli/php.ini with new version
Setting up libapache2-mod-php8.1 (8.1.2-1ubuntu2.14) ...

Creating config file /etc/php/8.1/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php8.1
Setting up php8.1 (8.1.2-1ubuntu2.14) ...
Setting up libapache2-mod-php (2:8.1+92ubuntu1) ...
Setting up php (2:8.1+92ubuntu1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for php8.1-cli (8.1.2-1ubuntu2.14) ...
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2.14) ...
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ subl helloworld.php
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo cp helloworld.php /var/www/html
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ cat helloworld.php
<?php
echo "Hello WOrld, this is my first php by gandi vijay";
phpinfo();
?>administrator@mwph-vm:~/waph-gandivr/labs/lab1$
```

Figure 11: Screen-13



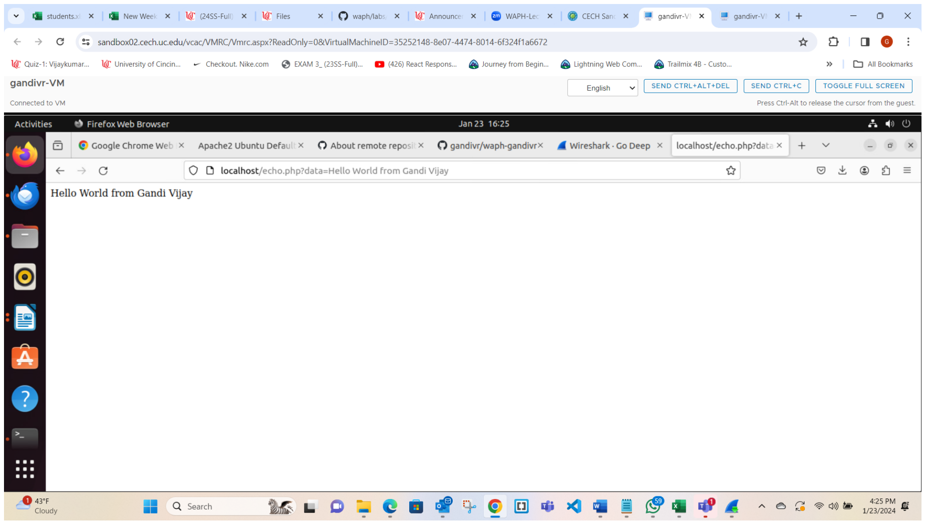
localhost/helloworld.php

Hello WOrld, this is my first php by gandi vijay

PHP Version 8.1.2-1ubuntu2.14

System	Linux mwph-vm 6.2.0-39-generic #40~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Nov 16 10:53:04 UTC 2 x86_64
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902

Figure 12: Screen-12



Hello World from Gandi Vijay

Figure 13: Screen-14

Task 3: Wireshark Analysis for GET and POST Requests

GET Request:

I have Started Wireshark capture. And then I have Accessed http://localhost/echo.php?data=HelloWorld in the browser. We have to stop Wireshark and observe the request and response in both terminal and Wireshark then I have followed the same and Stopped Wireshark and examined request and response messages.

GET Request OUtputs:

Output:

Output:

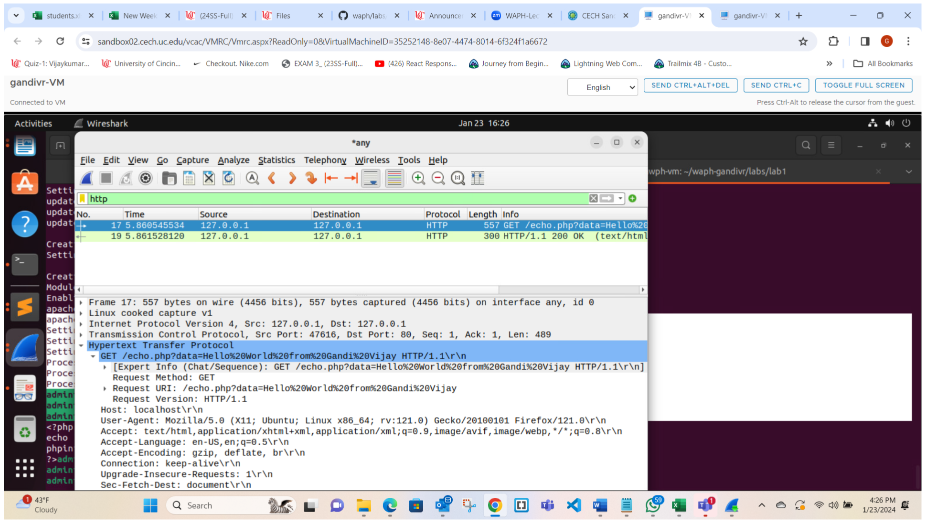
POST Request:

I have Initiated Wireshark capture. Then I Executed curl -X POST http://localhost/echo.php -d “data=Hello World!”. I have Checked output in the terminal. As the process I have Closed Wireshark and reviewed HTTP stream.

POST Request:

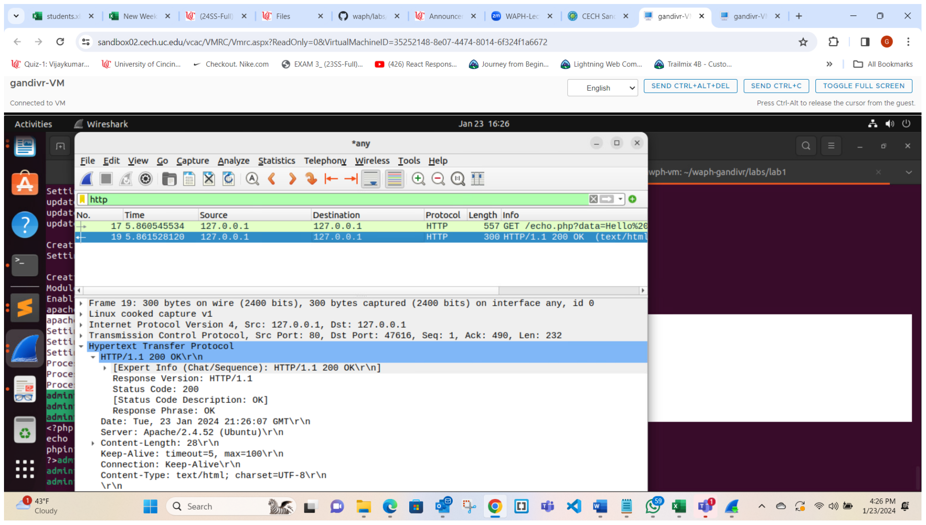
Output:

Output:



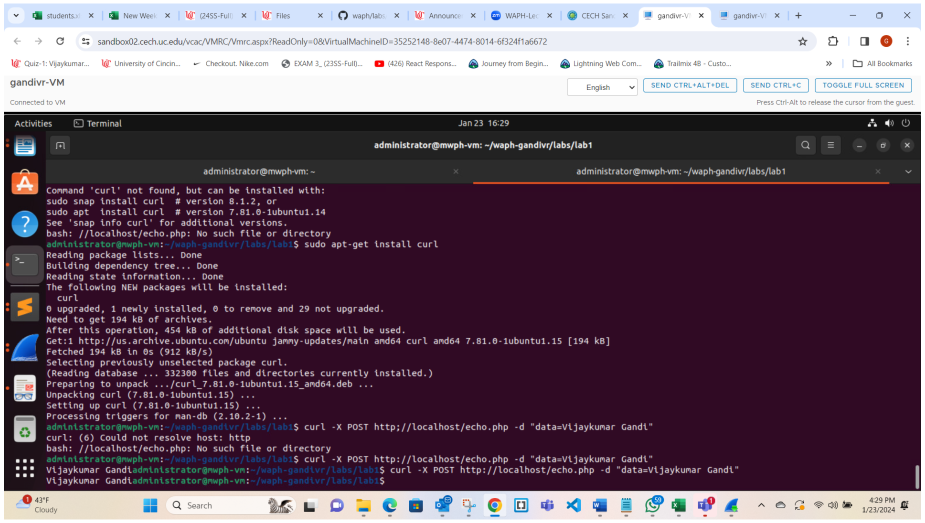
```
No.  Time         Source     Destination  Protocol  Length  Info
  17 5.860545534  127.0.0.1  127.0.0.1    HTTP         557  GET /echo.php?data=Hello%20
  19 5.861528120  127.0.0.1  127.0.0.1    HTTP         300  HTTP/1.1 200 OK  (text/html

Frame 17: 557 bytes on wire (4456 bits), 557 bytes captured (4456 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 47616, Dst Port: 80, Seq: 1, Ack: 1, Len: 489
Hypertext Transfer Protocol
  GET /echo.php?data=Hello%20World%20from%20Gandi%20Vijay HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /echo.php?data=Hello%20World%20from%20Gandi%20Vijay HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /echo.php?data=Hello%20World%20from%20Gandi%20Vijay
    Request Version: HTTP/1.1
  Host: localhost\r\n
  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:121.0) Gecko/20100101 Firefox/121.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate, br\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  Sec-Fetch-Dest: document\r\n
```

Figure 14: Screen-15



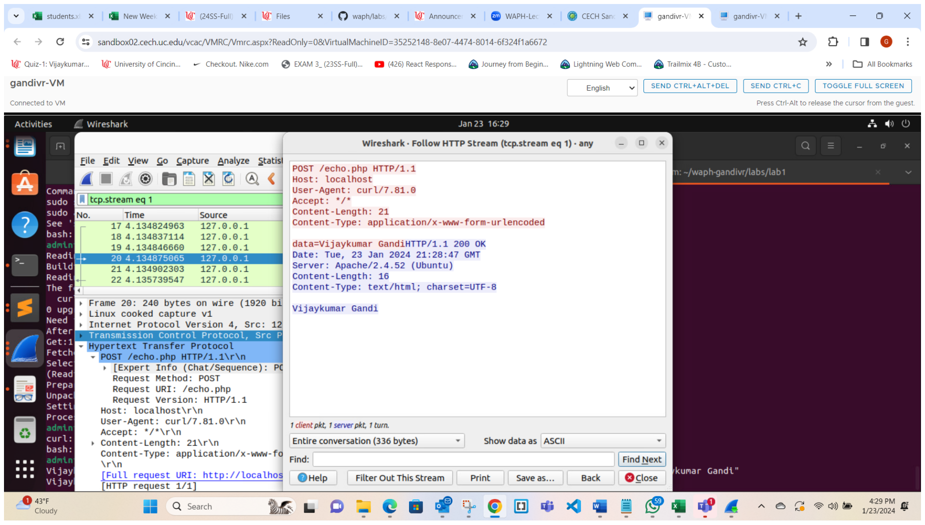
```
No.  Time         Source     Destination  Protocol  Length  Info
  17 5.860545534  127.0.0.1  127.0.0.1    HTTP         557  GET /echo.php?data=Hello%20
  19 5.861528120  127.0.0.1  127.0.0.1    HTTP         300  HTTP/1.1 200 OK  (text/html

Frame 19: 300 bytes on wire (2400 bits), 300 bytes captured (2400 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 80, Dst Port: 47616, Seq: 1, Ack: 490, Len: 232
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Date: Tue, 23 Jan 2024 21:26:07 GMT\r\n
  Server: Apache/2.4.52 (Ubuntu)\r\n
  Content-Length: 28\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  \r\n
```

Figure 15: Screen-16



```
Command 'curl' not found, but can be installed with:
sudo snap install curl  # version 8.1.2, or
sudo apt  install curl  # version 7.81.0-1ubuntu1.14
See 'snap info curl' for additional versions.
bash: //localhost/echo.php: No such file or directory
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ sudo apt-get install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 29 not upgraded.
Need to get 194 kB of archives.
After this operation, 454 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.15 [194 kB]
Fetched 194 kB in 0s (912 kB/s)
Selecting previously unselected package curl.
(Reading database ... 332300 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.15_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.15) ...
Setting up curl (7.81.0-1ubuntu1.15) ...
Processing triggers for man-db (2.10.2-1) ...
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ curl -X POST http;//localhost/echo.php -d "data=Vijaykumar Gandi"
curl: (6) Could not resolve host: http
bash: //localhost/echo.php: No such file or directory
administrator@mwph-vm:~/waph-gandivr/labs/lab1$ curl -X POST http://localhost/echo.php -d "data=Vijaykumar Gandi"
Vijaykumar Gandiadministrator@mwph-vm:~/waph-gandivr/labs/lab1$ curl -X POST http://localhost/echo.php -d "data=Vijaykumar Gandi"
Vijaykumar Gandiadministrator@mwph-vm:~/waph-gandivr/labs/lab1$
```

Figure 16: Screen-17



```
tcp.stream eq 1
No.  Time         Source
17 4.134824963  127.0.0.1
18 4.134837114  127.0.0.1
19 4.134846660  127.0.0.1
20 4.134875065  127.0.0.1
21 4.134902303  127.0.0.1
22 4.135739547  127.0.0.1

Frame 20: 240 bytes on wire (1920 bi
Linux cooked capture v1
Internet Protocol Version 4, Src: 12
Transmission Control Protocol, Src P
Hypertext Transfer Protocol
  POST /echo.php HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): PO
    Request Method: POST
    Request URI: /echo.php
    Request Version: HTTP/1.1
  Host: localhost\r\n
  User-Agent: curl/7.81.0\r\n
  Accept: */*\r\n
  Content-Length: 21\r\n
  Content-Type: application/x-www-fo
  \r\n
  [Full request URI: http://localhos
  [HTTP request 1/1]
```

Wireshark · Follow HTTP Stream (tcp.stream eq 1) · any

```
POST /echo.php HTTP/1.1
Host: localhost
User-Agent: curl/7.81.0
Accept: */*
Content-Length: 21
Content-Type: application/x-www-form-urlencoded

data=Vijaykumar GandiHTTP/1.1 200 OK
Date: Tue, 23 Jan 2024 21:28:47 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 16
Content-Type: text/html; charset=UTF-8

Vijaykumar Gandi
```

1 client pkt, 1 server pkt, 1 turn.

Entire conversation (336 bytes) · Show data as: ASCII

Figure 17: Screen-19

Comparison of GET and POST Requests:

I have Compared request messages in Wireshark.Noted that although the response message is the same, GET embeds data in the URL, while POST sends data in the header.