

WAPH - Web Application Programming and Hacking

Instructor: Dr. Phu Phung

Student

Name: Vamshi Reddy Gummadi'

Email: gummadvy@mail.uc.edu

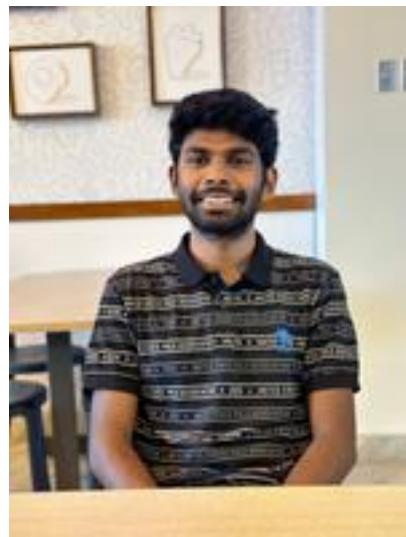


Figure 1: Headshot

Lab 1 Repository

Url for Lab 1: <https://github.com/gummadvy-uc/vamshi-reddy-gummadi>

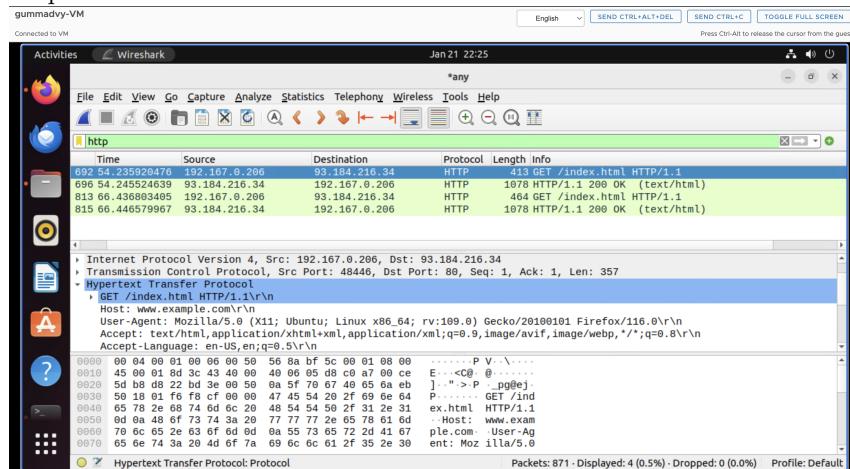
Lab 1 - Foundations of the Web

In this lab i familiar with wireshark tool and with the help of that tool i have i have captured the request and response in that tool.ALSO, applied http filter to check the request and response. With the help of telnet, I have checked the request and response. In the task2 i have learned some basic cgi application and also observed the output.Also, familiar with some php handson.

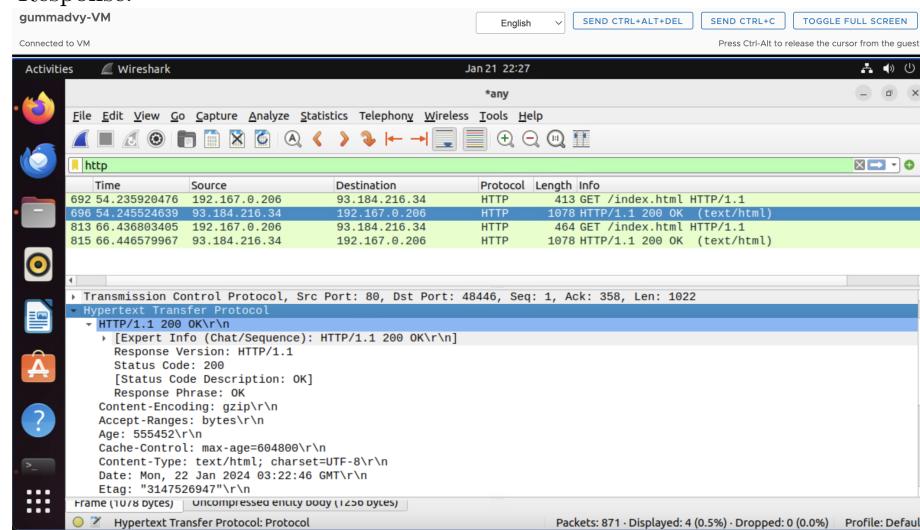
Part 1: The Web and HTTP protocol

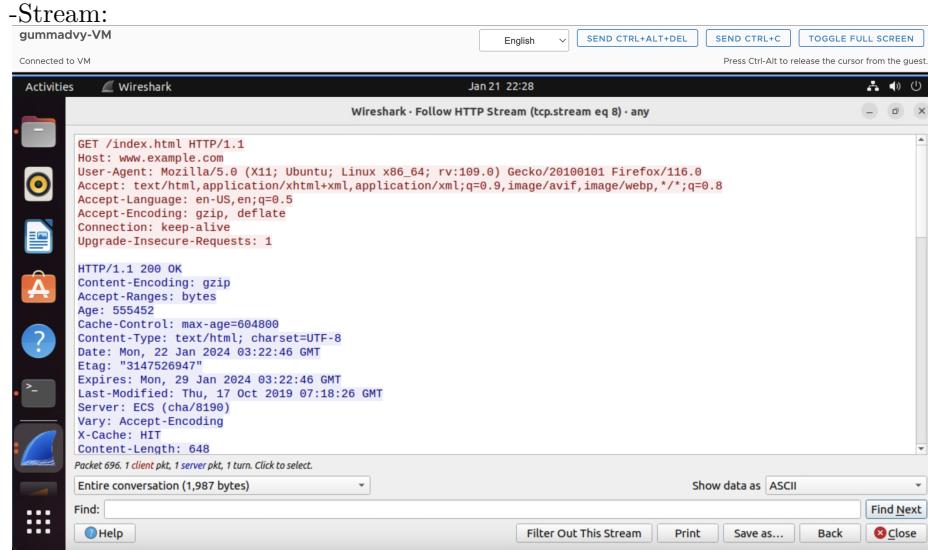
TASK 1: Familiar with Wireshark tool and HTTP protocol

- I have installed Wireshark tool using the sudo apt install Wireshark -qt command. And verified and run the application.
- I ran the Command sudo Wireshark & to open the Wireshark from the terminal.
- I clicked on the settings icon and selected any to capture all the interfaces, then clicked on start button.
- I opened a browser in a virtual machine and searched for the http://example.com/index.html , later returned to the Wireshark tool and stopped tracking.
- I applied a http in the filter bar to check the request and response of the example website. 6. clicked on the follow -> HTTP Stream to check the request and response.
- Request:



-Response:





TASK 2: Understanding HTTP using telnet and Wireshark

- I have clicked the start button in Wireshark to capture the http request and response.
- I have ran the command telnet example.com 80 in the terminal to connect to the web server. After success, I have written GET/index.html HTTP/1.0, clicked enter, written Host: example.com, clicked enter for two times.
- After clicking on the enter twice, I have received the output of the example.com . Both the request and response.
- I have compared the request of both browser and telnet. I can see , when I run via browser there are some additional field are there such as user-agent, accept, connection.
- I have compared the response of both browser and telnet. The only difference is content encoding is present when I run via browser.

gummadvy-VM

Connected to VM

Activities Terminal Jan 21 22:36

administrator@mwpv-vm: ~

```
administrator@mwpv-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: 556194
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Mon, 22 Jan 2024 03:35:08 GMT
Etag: "3147526947+ident"
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (cha8190)
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" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<style type="text/css">
```

gummadvy-VM

Connected to VM

Activities Wireshark Jan 21 22:54

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

No. Time Source Destination Protocol Length Info

1	185.48.580142153	192.167.0.206	93.184.216.34	HTTP	58	GET /index.html HTTP/1.0
2	187.48.589219140	93.184.216.34	192.167.0.206	HTTP	1666	HTTP/1.0 200 OK (text/html)

Linux Cooked capture V1
Internet Protocol Version 4, Src: 192.167.0.206, Dst: 93.184.216.34
Transmission Control Protocol, Src Port: 58436, Dst Port: 80, Seq: 46, Ack: 1, Len: 2
[3 Reassembled TCP Segments (47 bytes): #147(26), #183(19), #185(2)]
Hypertext Transfer Protocol
[GET /index.html HTTP/1.0\r\n]
[Expert Info (Chat/Sequence): GET /index.html HTTP/1.0\r\n]
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 frame: 187]
[Response in frame: 187]

Packets: 237 · Displayed: 2 (0.8%) Profile: Default

gummadvy-VM

Connected to VM

Activities Wireshark Jan 21 22:55

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

No. Time Source Destination Protocol Length Info

1	185.48.580142153	192.167.0.206	93.184.216.34	HTTP	58	GET /index.html HTTP/1.0
2	187.48.589219140	93.184.216.34	192.167.0.206	HTTP	1666	HTTP/1.0 200 OK (text/html)

Internet Protocol Version 4, Src: 93.184.216.34, Dst: 192.167.0.206
Transmission Control Protocol, Src Port: 80, Dst Port: 58436, Seq: 1, Ack: 48, Len: 1610
Hypertext Transfer Protocol
HTTP/1.0 200 OK\r\n
[Expert Info (Chat/Sequence): HTTP/1.0 200 OK\r\n]
[Request Version: HTTP/1.0]
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
Age: 557311\r\n
Cache-Control: max-age=604800\r\n
Content-Type: text/html; charset=UTF-8\r\n
Date: Mon, 22 Jan 2024 03:53:45 GMT\r\n
Etag: "3147526947+ident"\r\n
Expires: Mon, 29 Jan 2024 03:53:45 GMT\r\n

Packets: 237 · Displayed: 2 (0.8%) - Dropped: 0 (0.0%) Profile: Default

Part 2:

Task1:

a)

- I have created a HelloWorld.c program in sublime and written the code which was given by the professor.
 - By default CGI daemon is not enabled in Apache2, I have ran the command sudo a2enmod cgi and restarted apache server using the sudo systemctl restart apache2.
 - I have installed the gcc by using the sudo apt install gcc command
 - To compile the HelloWorld.c file I have used the command gcc HelloWorld.c -o HelloWorld.cgi clicked enter.
 - I have copied the HelloWorld.cgi to /usr/lib/cgi-bin by using sudo cp HelloWorld.cgi /usr/lib/cgi-bin.
 - I have run the link in the browser(<http://localhost/cgi-bin>HelloWorld.cgi>) and got the response as expected.

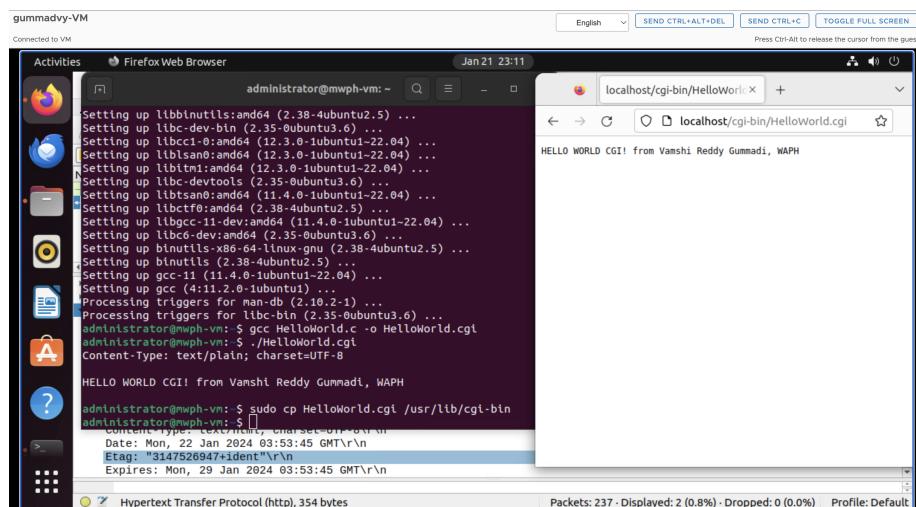


Figure 2: Screenshot7

b)

- Code for Index.c:

```
#include<stdio.h>
int main(void){

    printf("Content-Type: text/html; charset=UTF-8\n\n");
    printf("<!DOCTYPE html> \n <html>\n <head>\n <title> Vamshi Reddy Gummadi </title>\n </head>\n <body>\n </body>\n </html>");
}
```

```

        return 0;
}

```

- I have used the same commands for the index.c as above and run the link <http://localhost/cgi-bin/index.cgi>
- Received the output as expected.

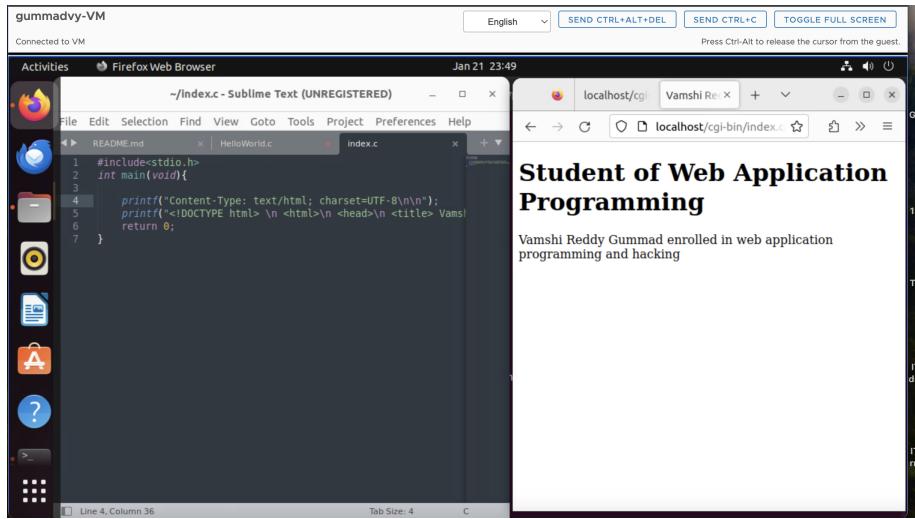


Figure 3: Screenshot8

Task 2:

a)

- I have installed PHP using the command `sudo apt-get install php libapache2-mod-php -y`.
- I have created the hello world.php in lab1 folder.
- Deployed the code to the web server by going into the directory of the file and run the command to copy that file into the server `sudo cp helloworld.php /var/www/html`.
- I have used the `localhost/helloworld.php` to check if the output is coming.

b):

- I have created a echo.php file in lab1 folder.
- Source code:
- Copied the code into `/var/www/html` and run the `http://localhost/echo.php?data=Hello`

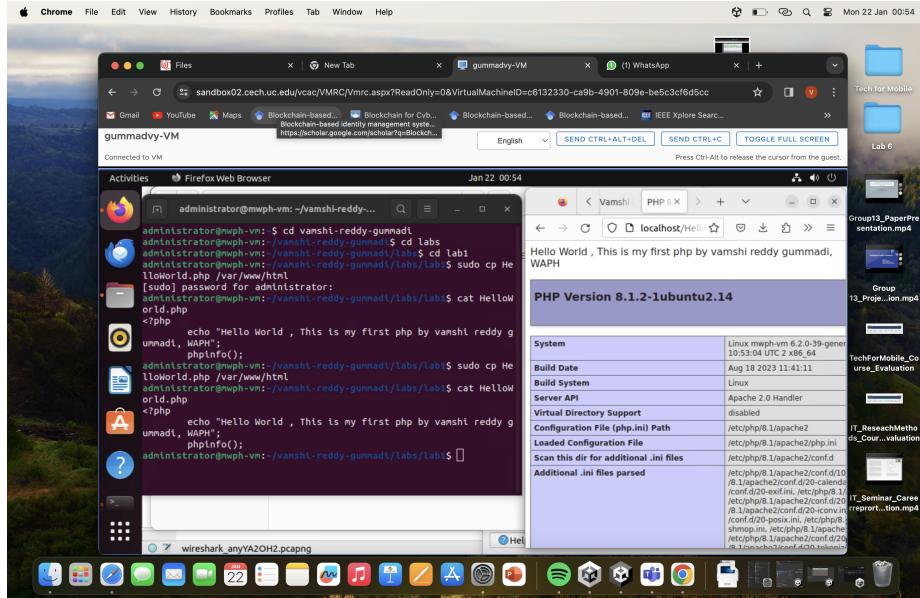
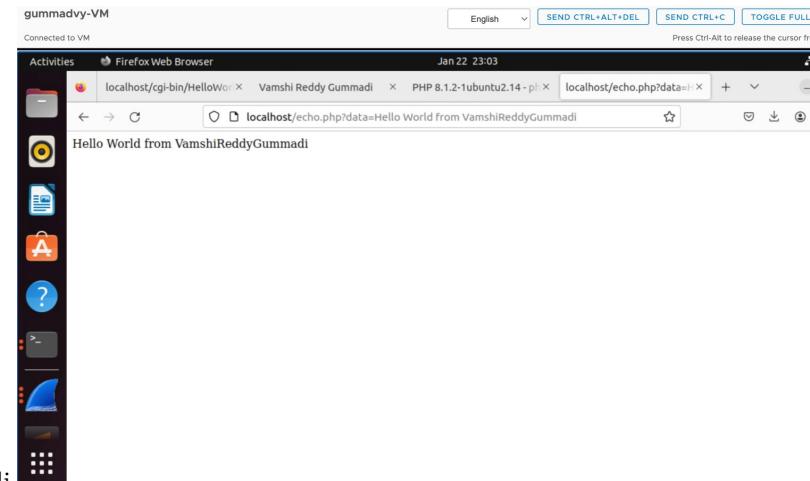


Figure 4: Screenshot9



World From VamshiReddyGummadi

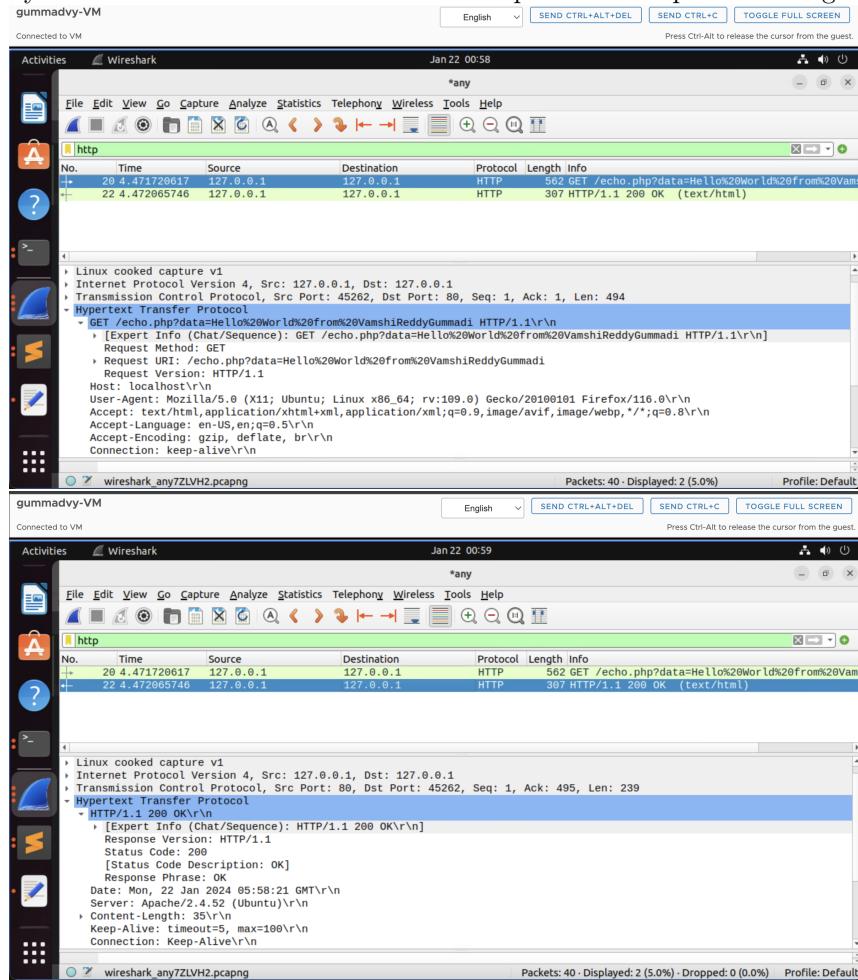
c:

- If the input for the request has not been properly checked then it could lead to some security issues.
- Always validate the data for request.
- If the data is not handled correctly, it may cause compromising the confidentiality like personal information.

Task 3:

a)

- I have started the Wireshark.
- Run the `http://localhost/echo.php?data=Hello World` From VamshiReddyGummadi in browser. checked the request and response messages



b):

- I have started the Wireshark.
- I have used the command to `curl -X POST http://localhost/echo.php -d "Hello world, from VamshiReddy Gummadi"` and opened http stream and observed the output.

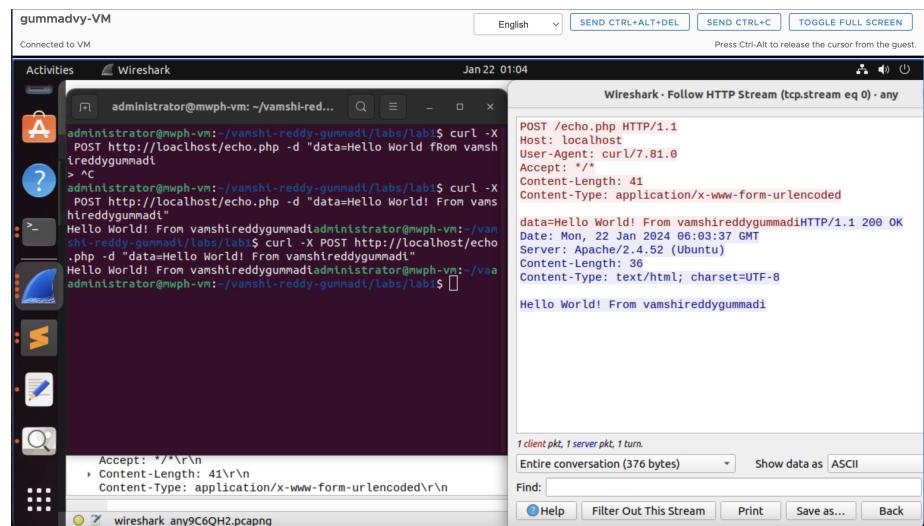


Figure 5: Screenshot12