waph-peddinry

WAPH-WEB APPLICATION PROGRAMMING AND HACKING

instructor : Dr.Phu Phung

Name: Narendra Reddy Peddireddy **Email**: peddirny@mail.uc.edu

Repository's URL: https://github.com/narendra2105/waph-peddinry.git

Narendra Reddy Peddireddy uses this private repository to house all of the course's code. The following is how this repository is organized.

Lab's

-lab 1:Foundations of the Web ### Overview of Lab1

The two lab sections are labeled "The Web and HTTP Protocol" and "The Web-Application Programming" Use of the Wireshark program is stressed in the "The Web and HTTP Protocol" sections in order to examine the protocol.

###Hands-on hacking exercises ### Part I: The Web and HTTP Protocol ### Task1: Through this assignment, I became familiar with the potent network protocol analyzer Wireshark. Enter "sudo wireshark &" in the terminal to start the Wireshark monitoring process. Click the 'any' interface to move on to step 2 and start capturing.Step 3 involves opening a web and typing "http://example.com/index.html." Going to Wireshark, disable the filter.Learn about the features of Wireshark by selecting the relevant request and answer. Part 1.1 contains screenshots of this process.The Wireshark program, accessible via the terminal, facilitates thorough observation of HTTP protocols. Screenshots demonstrate the tool's efficacy in monitoring network activity by displaying the captured requests, responses, and HTTP streams.

Wireshark accessing HTTP requests

Wireshark accessing HTTP responses

Both HTTP request and responses

Task2:Understanding HTTP using telnet and Wireshark

Telnet can be used to initiate requests and responses by using the terminal with the command "$telnet example.com 80." Once the server is connected, type "GET/HTTP/1.1" and then "host: example.com" and hit enter two times to produce request and message requests response that can be seen and examined in Wireshark.

Figure 1: Narendra Reddy Peddireddy Headshot

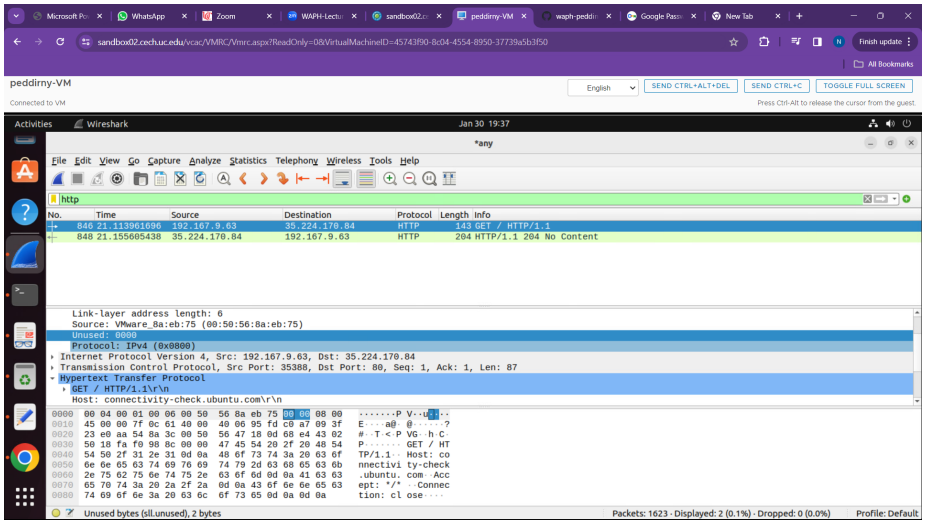


Figure 2: Wireshark accessing HTTP requests

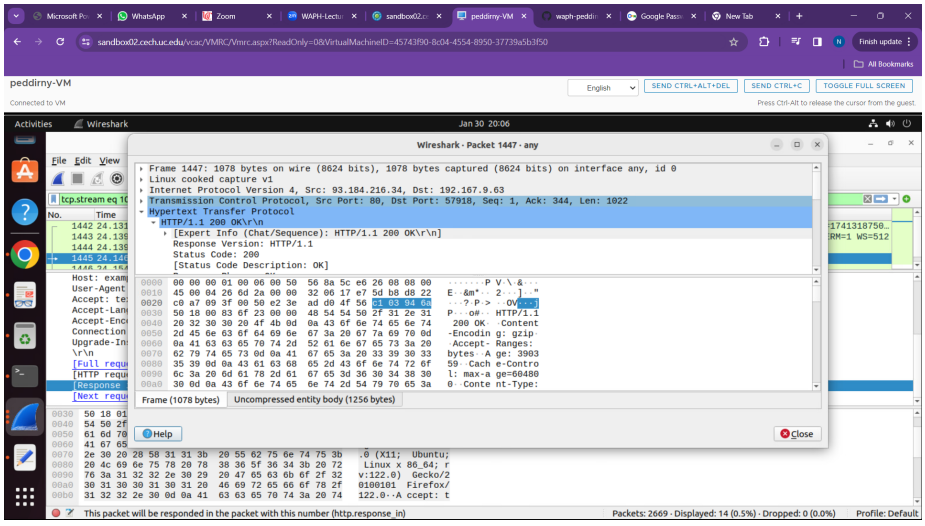


Figure 3: Wireshark accessing HTTP responses

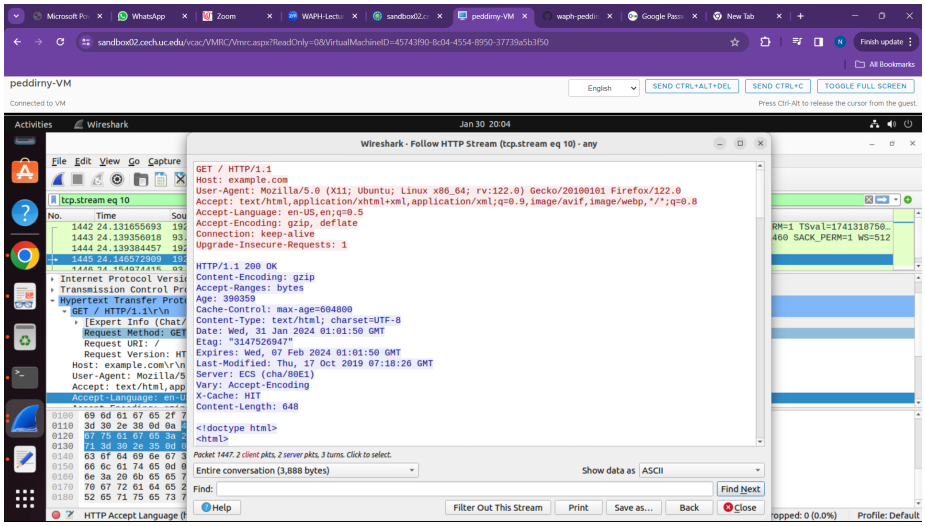


Figure 4: Wireshark accessing HTTP responses

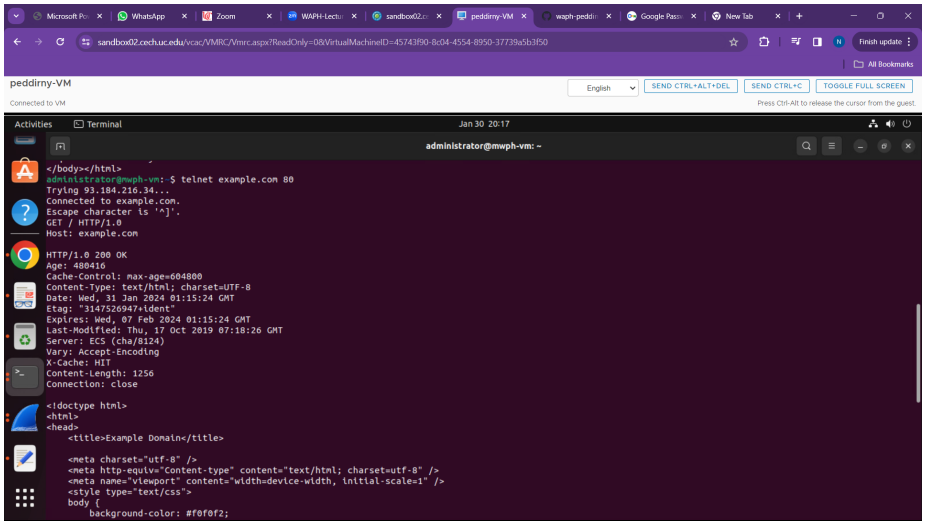


Figure 5: Screenshot4

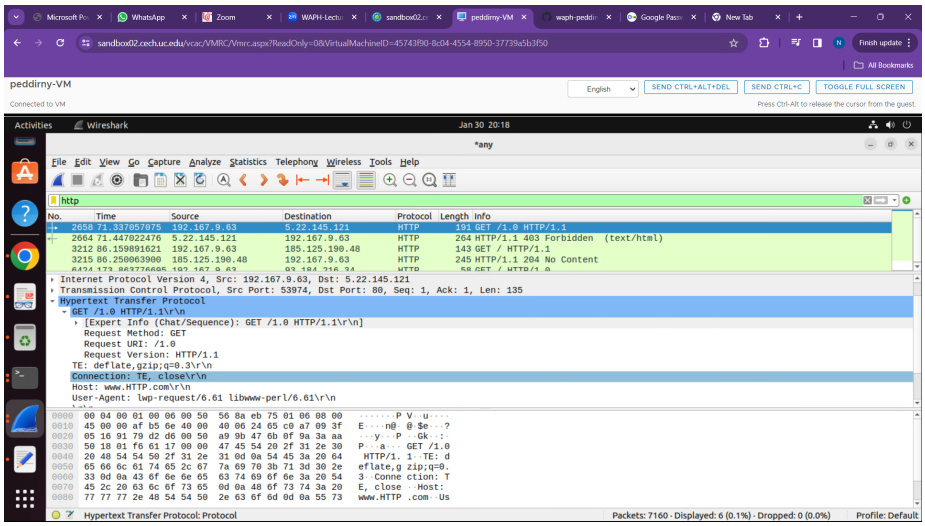


Figure 6: Screenshot5

Telent HTTP Request

HTTP request and responses

Part TWO: "Basic Web Application Programming"

Task1 A :"CGI Web applications in C"

Using the Sublime editor, I wrote a C programming file to install gcc. Then, I used sudo to compile it into a CGI file by running "gcc helloworld.c -o helloworld.cgi." By transferring the final CGI file to the /usr/lib/cgi-bin/ directory, it may be executed on the server for localhost access and therefore be deployed.

```
c#include <stdio.h> int main(void) { printf("Content-Type: text/plain;
charset=utf-8\n\n"); printf("Hello World CGI! From Narendra,
WAPH\n\n"); return 0; }
```

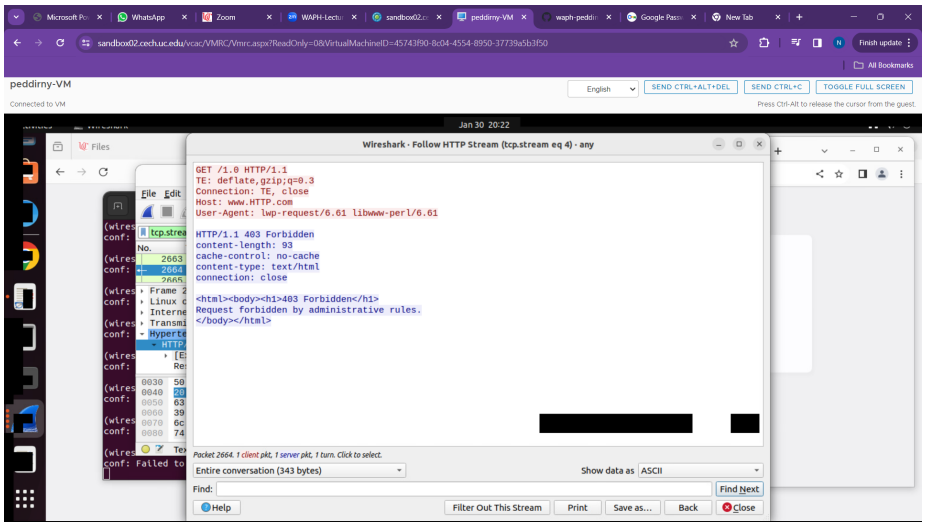
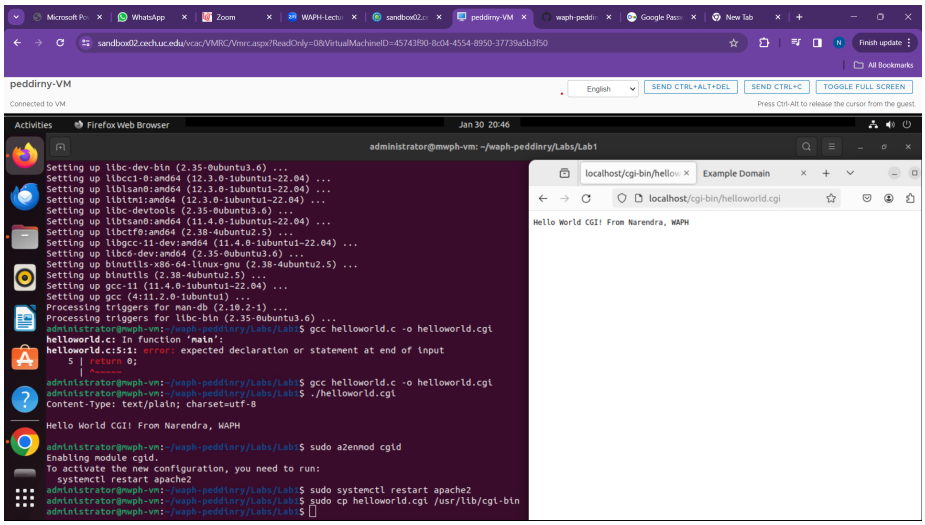


Figure 7: Screenshot6



B CGI program and deploy it with a simple HTML template

In this case, I created a copy of the original C programming file and added HTML information to the new copy. To smoothly integrate the HTML data and changed the modified C file into a CGI file and ran it on the server.

Html Code

```
c#include<stdio.h> int main(void){ printf("<html><body><h1>
Welcome to Web Applications </h1><p>Hello World CGI! From Narendra,
```

```
WAPH</p></body></html>); return 0; }
```

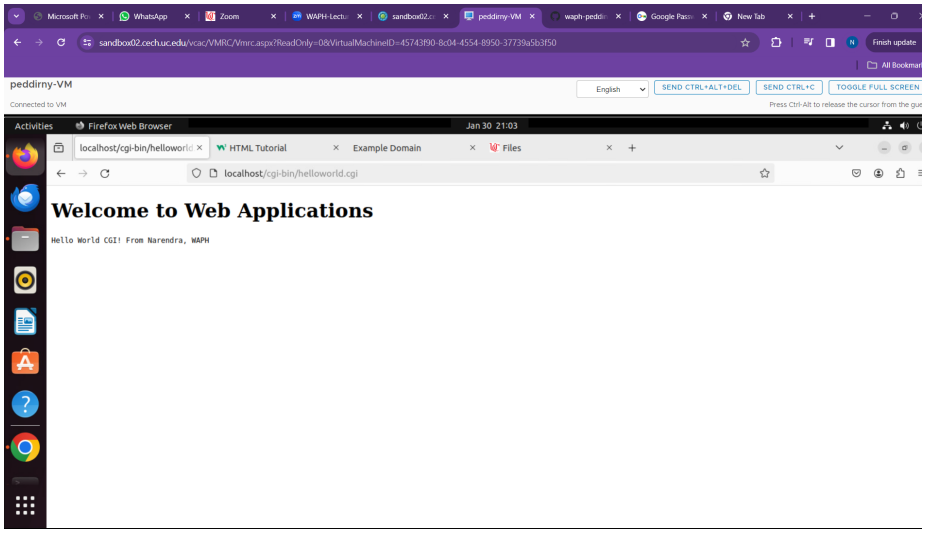


Figure 8: Screenshot8

Task 2: "A simple PHP Web Application with user input"

A: Writing PHP code is the first step in creating a helloworld.php file. PHP commands such as "sudo cp helloworld.php /var/www/html" are then executed to transmit the file to localhost. and Search for the file in the browser.

B: In the "Echo program" section, a PHP script is presented where sensitive data exploitation and susceptibility to injection attacks are identified as security risks. Additionally, a PHP application named "Helloworld.php" is created, and a separate PHP code snippet ("Echo.php") is developed for requesting data from a URL, monitored using Wireshark to observe generated requests and responses on the server.

Task 3: "Understanding HTTP POST AND GET requests"

HTTP GET

A:Launch the application, start the data collection process, then run the echo.php file in the web browser. Put an end to the Wireshark capture, select the relevant HTTP request and response files, and apply the "http" filter. As seen in the attached image, you can follow the HTTP stream in a new window.

HTTP POST

B: Firstly, I acquired Curl libraries through the terminal, followed by utilizing the "curl -X post" command to transmit data to the server. Subsequently, I

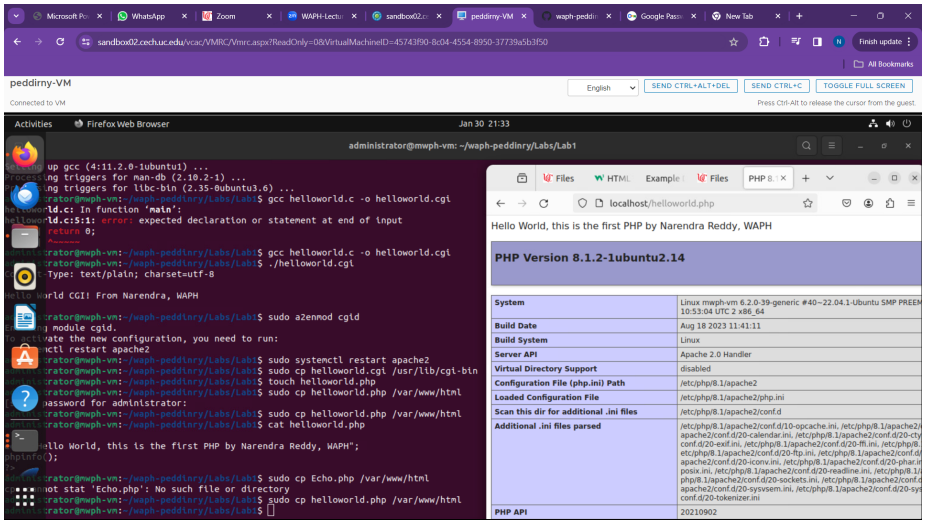


Figure 9: Screenshot9

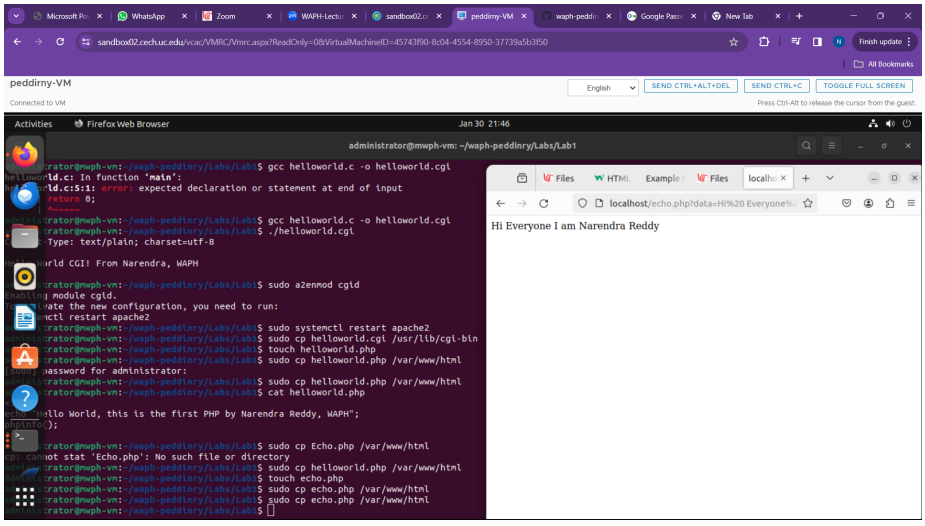


Figure 10: Screenshot10

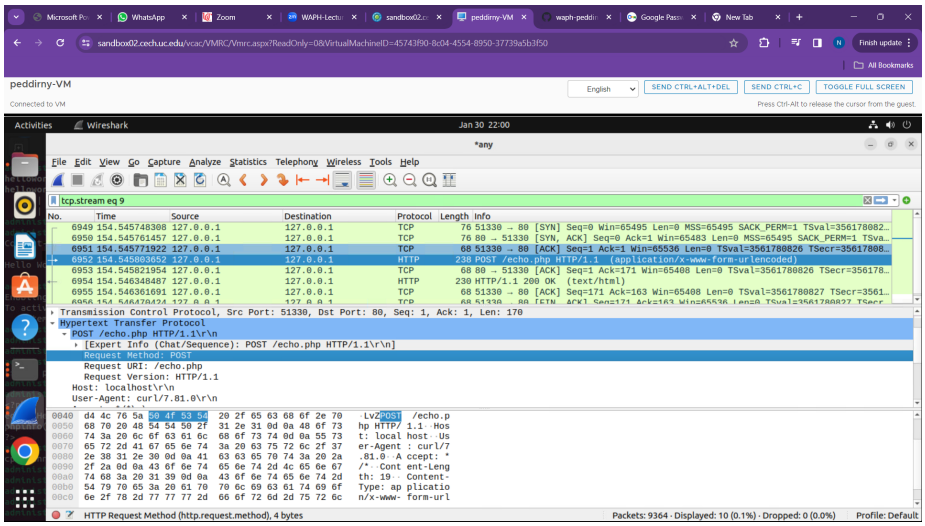


Figure 11: Screenshot11

observed the generated requests and responses in Wireshark.

C:The HTTP protocol has two methods that are frequently used to facilitate communication between servers and clients: HTTP POST and HTTP GET requests.

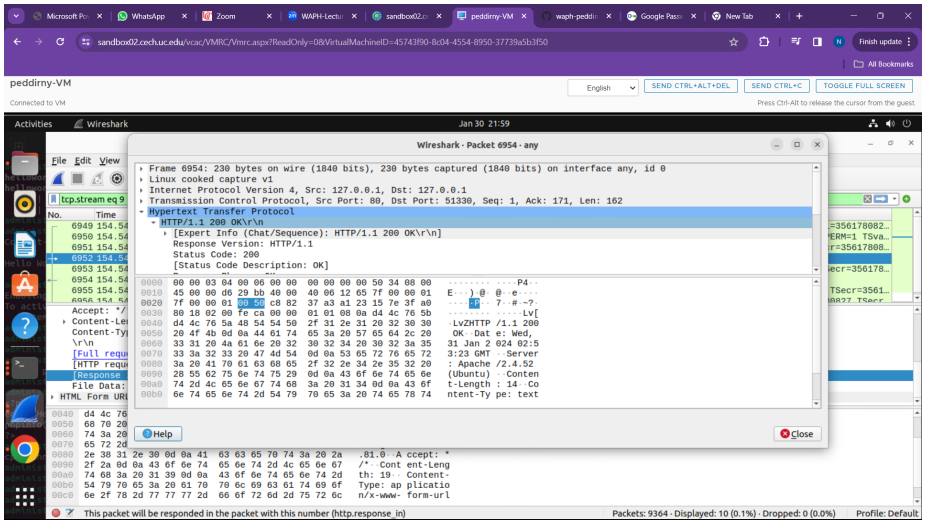


Figure 12: Screenshot12

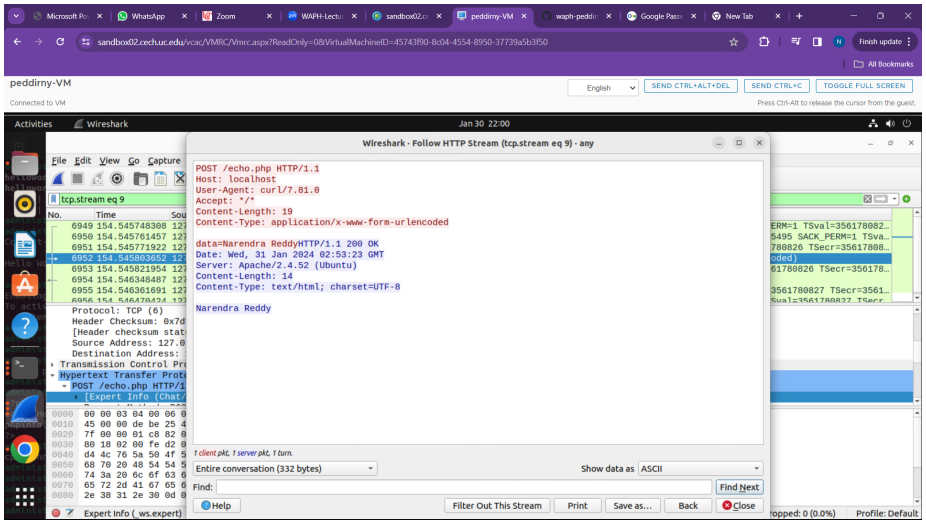


Figure 13: Screenshot13