**A picture containing drawing

Description automatically generated**

**IE3092**

**Information Security Project**

**3rd Year, 2nd Semester**

**MID\_REVIEW**

IT18166934 – Hussain M.H IT18036626 - Liyanage U.I.D

**Introduction**

Our CTF box is initially a web based box and eventually we will be moving to a shell based CTF. The box is being made using an older Ubuntu server version (12.04). The CTF hunter should be in the same ip range and hunt for the ip address of the server to move forward.

**Audience/Theme**

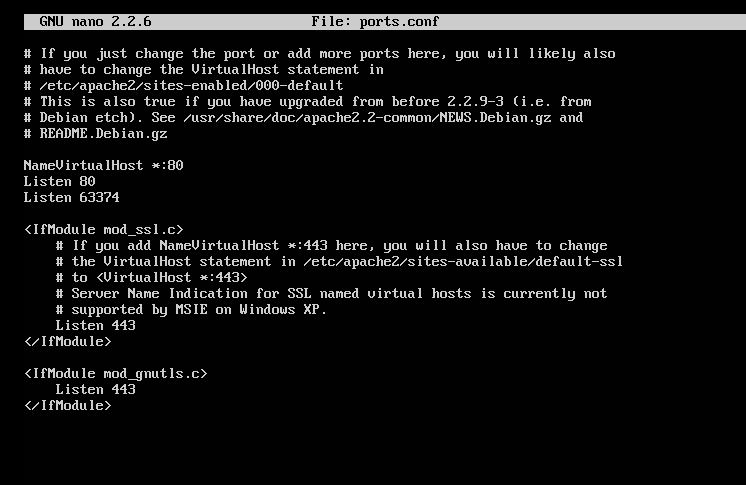
We will be focusing on software engineers in software companies with the motive of keeping them aware of vulnerabilities which could be in web sites and servers while development.

**Implementation**

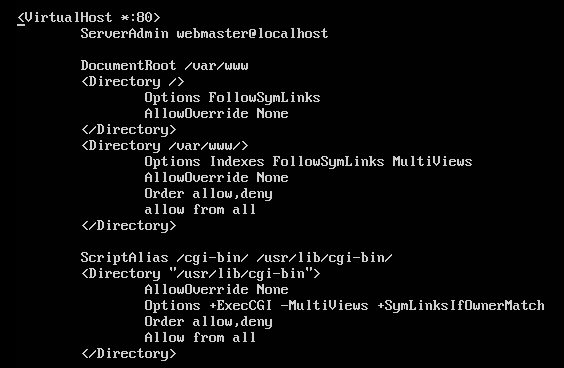
We have used an Ubuntu server version to host our CTF box and there are two apache instances installed in the same Ubuntu server.

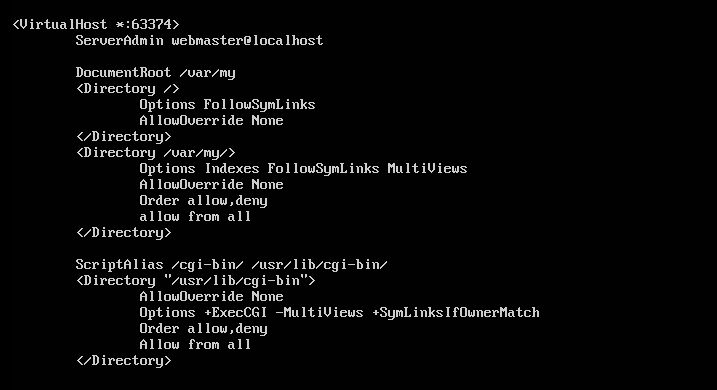
Inside the /etc/apache2 configuration files we have done the needed modification to create the two instances.

* Added the uncommon port along with the default port in the apache2.conf



* Created a new path for the uncommon port in /sites-available/default





At one apache instance, we have assigned the first apache instance to a default and common port and the other apache instance is assigned to an uncommon port.

**First apache instance**

**FLAG#1**

* In the first apache instance we have created two web pages and on the first web page, it is assigned to the default page. And to make things easier at the beginning the CTF player could find the first flag if he/she refers the page source of that web page.

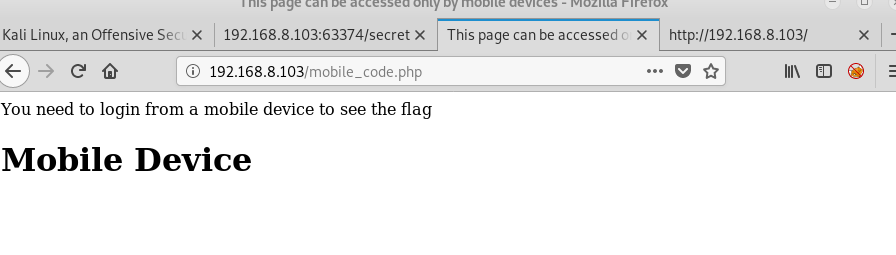
This flag is encrypted using base64, therefore to get the original format, it should be decrypted using base64.



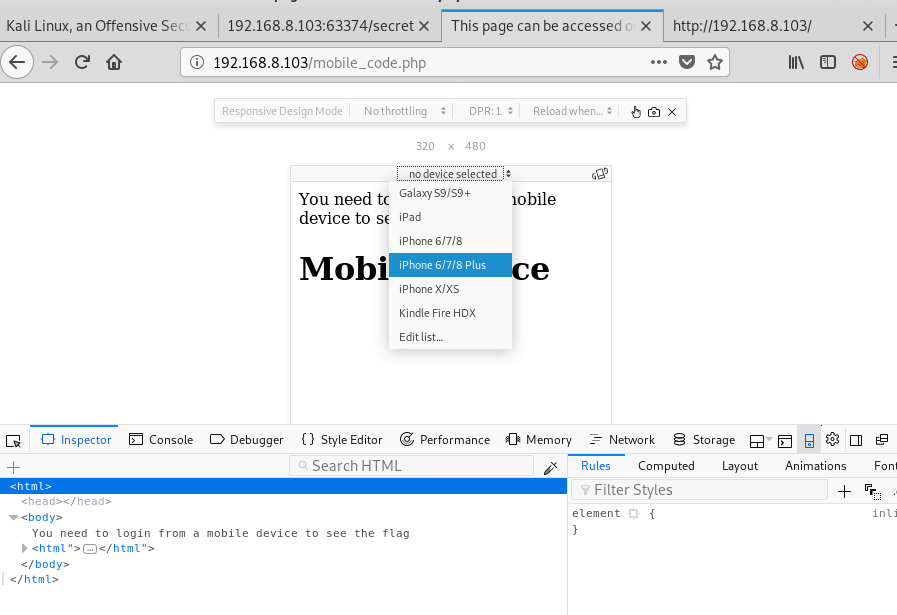
* And there is a second web page in this apache instance and this page is kept hidden.
  + To find out this hidden page, we have to use a directory directory bruteforce. We can use the following command to do so.
    - **dirsearch –u http://<IP>:<port> -e php,html –l <wordlist path>**
  + But the page cannot be found using the default kali directory wordlist
  + The CTF player have to guess the robot.txt file and then he/she could find a random folder in it which the customised wordlist is found.
  + Using that wordlist the hidden page could be found using directory bruteforce.

**FLAG #2**

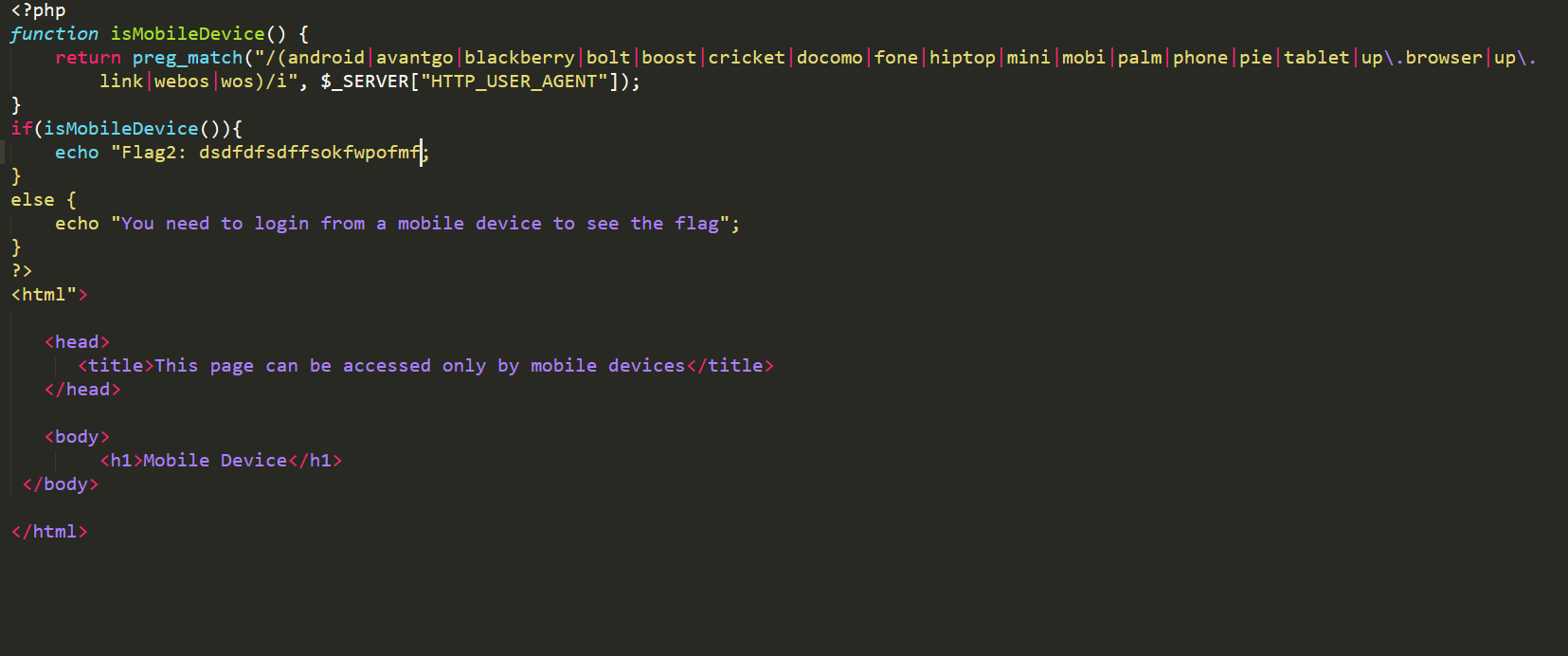
After finding the hidden page, and when trying to access that particular web page, it will show that the page could be accessed only by mobile devices.



Therefore the page should be modified to be responsive to any device and when it is accessed by mobile device, the FLAG#2 will be received.



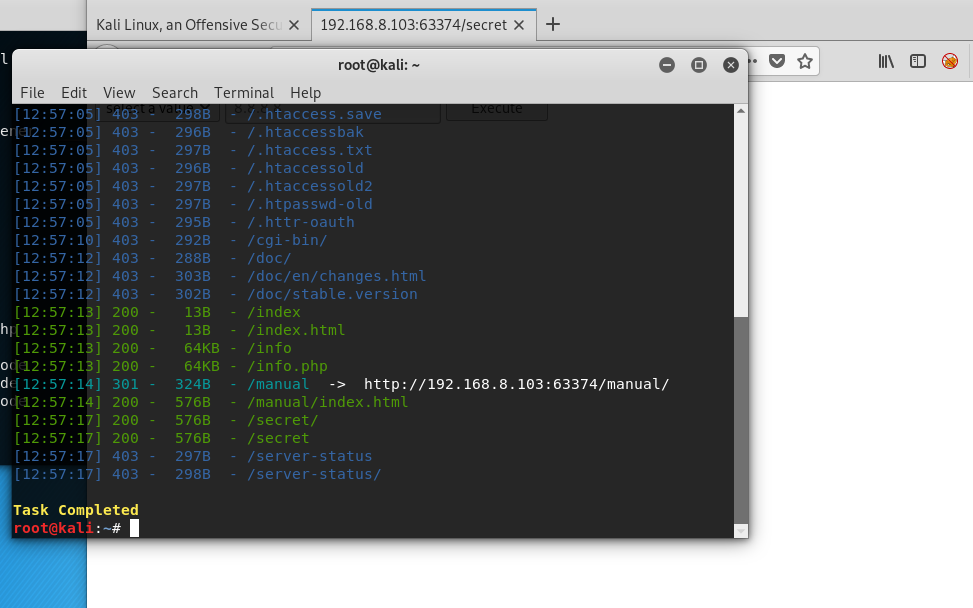
Following is the source code for mobile responsiveness implementation.



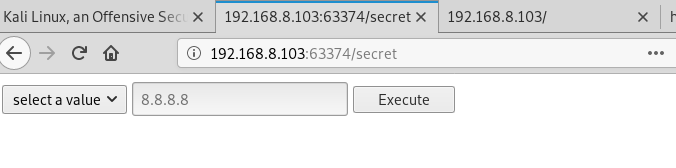
And that’s all with the apache instance one.

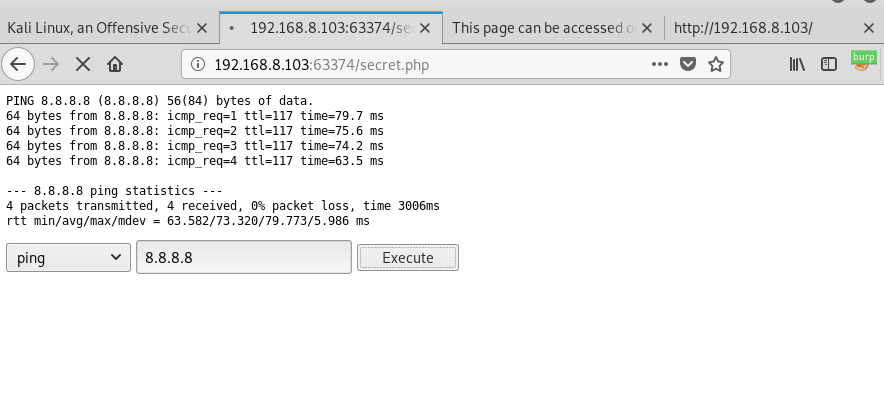
**Second Apache instance**

As in the first apache instance, there are two web pages but only one will be used. The default page has no functions whereas the other hidden page which could be found using the same directory list found on the first instance is what we will be using. The following diagram shows the results of the directory bruteforce and we could observe a page named “secret” and that’s the page we r looking for.



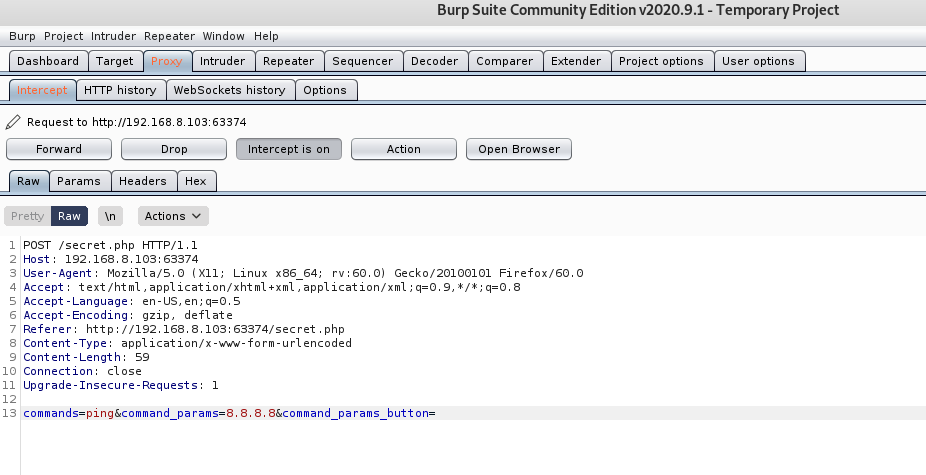
The hidden web page is created with a **command-injection vulnerability**.



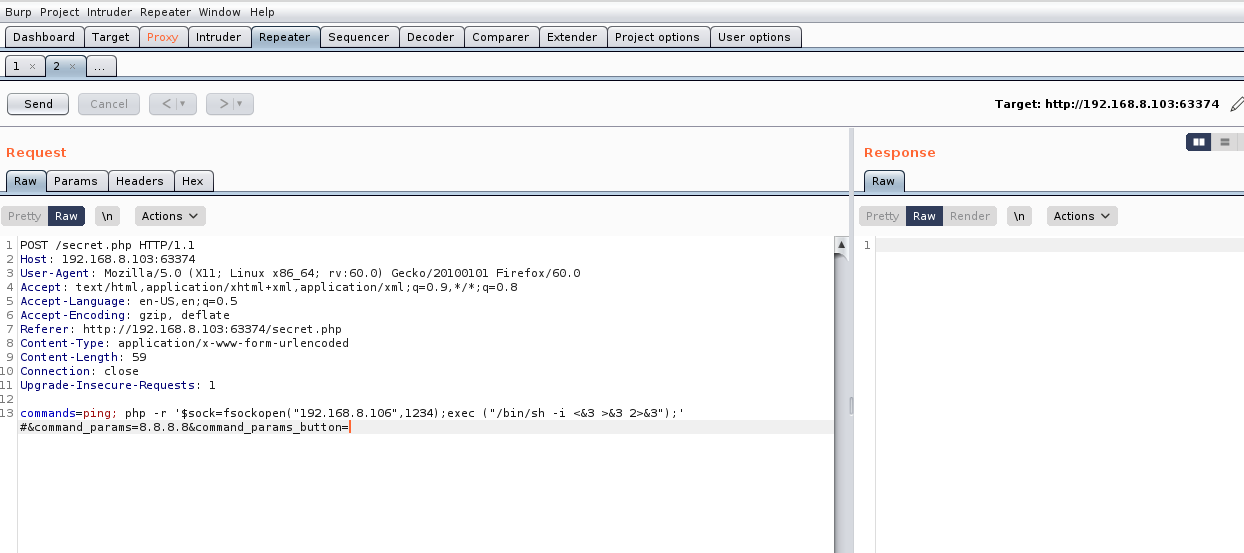


Since this page is vulnerable to command injection, the CTF hunter could capture the request from burp and try to modify the request to obtain a reverse shell for a low privileged user. In order to obtain the reverse shell the hunter should inject a payload which will satisfy the needs.

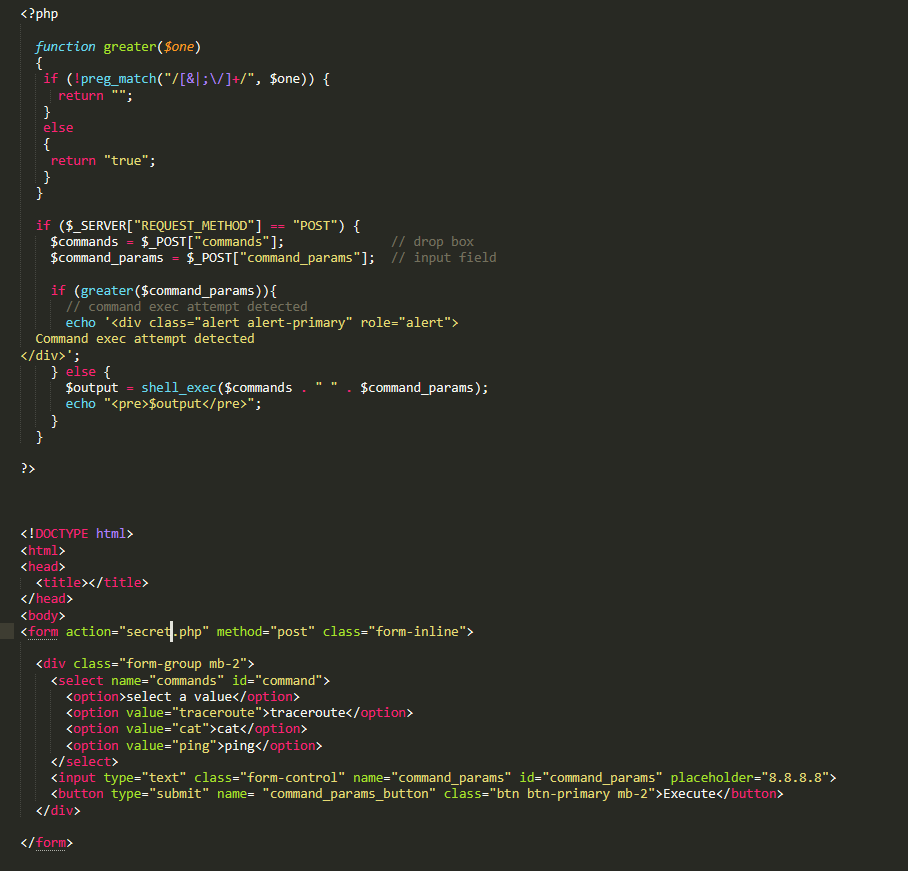
Observe the below diagram



Once captured the request, send it to repeater inject payload.

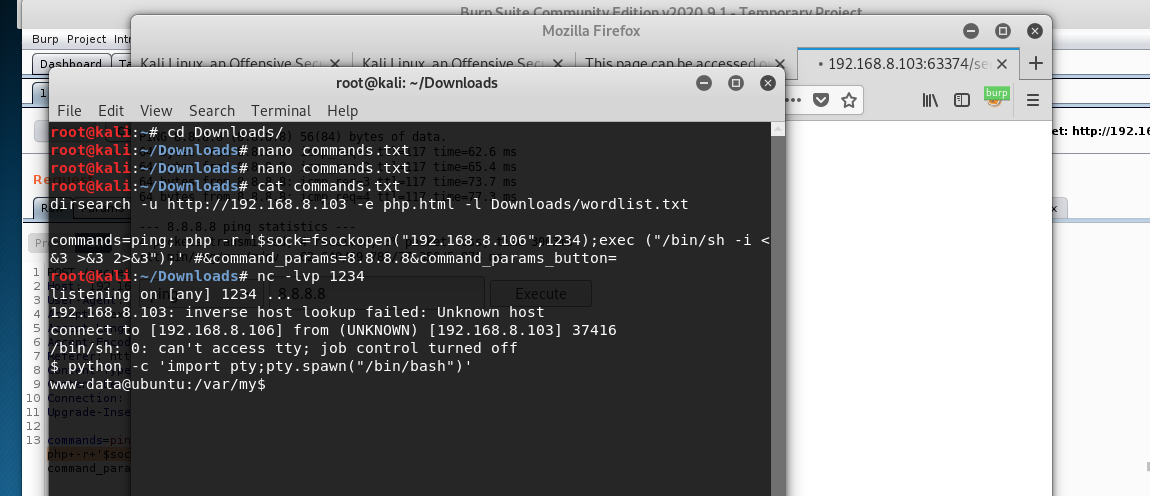


Following is the source code for the command injection vulnerability.



Once the payload has been injected successfully, the low privilege user (www-data) could be obtained by opening the listener.

This has been successfully exploited to get the low privilege user.



That is the progress of our CTF so far.

**Future progress plans**

We have planned to implement a poorly configured FTP server locally and once obtained the www-data it can be exploited and we can get to user account.