

## INTODRUCTION

This project had helped me to have a better understanding how the network server is a unification of works. When creating and designing the Server from Scratch, without knowledge and without using any software that provides all these applications that I used such as (XAMPP, Wamp, WordPress...), I spent a lot of time on what every little command or step would do and in how everything that I was setting up would work. Implementing everything and installing all the software, wasn't too big of an issue, but when it came to configure and make everything work, that was a new experience. Connecting a database with PHP and HTML using apache2 can be testing, even when the individual pieces work on their own. Setting up of apache2 in Ubuntu taught me about configuration of apache2, Ubuntu, bash, and Linux in general, and how parts like the website of the server, can be complex when appear simple.

This program also taught me the importance of setting up the configuration files for these programs and the configuration of the ports. When adding MYSQL database to the code in PHP I failed, and had to restart again. The PHP server wasn't allowed to enter the server through localhost, which I realized in the first try that I wasn't the localhost. I was connecting from SSH using ubuntu Server. Due to multiple fails. I restarted the project from scratch, installed, and set up everything with more precision.

In the second part of the project, I proved myself that I could be a black hacker and get in my own system breaking down some of the errors I had. When I started scanning and looking for vulnerabilities, I realized that I made the server really secured and that it wasn't going to be easy to get in it. At the end, and through some information that I got on the website and scanning, I was able to get into the server using brute force attacks; I connected through the SSH port, being able to modify the html files, making the website login through an SMPT server to send me emails about every registration that the website would made.

## WHAT DID YOU LEARN FROM THIS PROJECT?

- Using apache2 and configuring the server, permission of users for the web, use of DNS and website addresses, ports connections and the creation and configuration of my own SSL certificate.
- HTML to edit, create websites, encrypt passwords and use of other methods and actions.
- PHP for formatting and getting data to send to the database, use of different classes.
- MYSQL and uses of databases on PhpMyAdmin.
- SMTP server to send automatics emails with data obtained from PHP.
- Use of different exploits for software.
- Bash scripting to create scripts that help with scanning process.
- Improve of the use of some tools such as WFUZZ, NMAP, SSH-Client, etc....
- Escalating Permissions and privileges of users in Linux distribution.

## COMMANDS FOR THE PROJECT

Here, there is a list with some of the commands I used when setting the Ubuntu Server up:

- Sudo apt update
- Sudo apt upgrade
- Sudo apt install apache2
- Sudo ufw app list
- Sudo ufw allow 'Apache'
- Sudo service ufw status
- Sudo service apache2 status
- Localhost on website to 10.0.2.15
- Sudo apt install mysql-server mysql-client php-mysql
- Sudo apt install php libapache2-mod-php php-mysql
- Sudo nano /var/www/html/info.php -> phpinfo();
- Sudo apt install phpmyadmin ( user phpMyAdmin , password 12341234)
- Sudo nano /etc/apache2/apache2.conf - > Include /etc/phpmyadmin/apache.conf

- Sudo a2enmod rewrite
- Sudo nano /etc/apache2/sites-available/000-default.conf -> add ( RewriteEngine On  
, RewriteCond %{SERVER\_PORT} 80 , RewriteRule ^(.\*)\$  
https://alvaroleon1234.com%1 [R,L]
- Sudo mkdir /etc/ownership - > cd /etc/ownership
- Sudo Openssl genrsa -des3 -out server.key 4096 ( password 12341234)
- Sudo openssl req -new -key server.key -out server.csr ( challenge password  
alvaroleon1234)
- Sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
- Sudo openssl rsa -in server.key -out server.key.insecure
- Sudo Mv server.key server.key.secure
- Sudo mv server.key.insecure server.key
- Sudo chmod 000 \* and sudo chown root:root \*
- Sudo nano /etc/apache2/ports.conf ( edit port 443 to listen)
- Sudo a2enmod vhost\_alias
- Sudo nano /etc/apache2/sites-available/000-default.conf ( ServerName  
www.alvaroleon1234.com)
- Sudo nano /etc/hosts ( 127.0.0.1 alvaroleon1234.com)
- Sudo nano /etc/apache2/sites-available/default-ssl.conf ( change the routes for ssl  
certificate and add alias and name of website )
- Sudo a2ensite default-ssl.conf
- Sudo a2enmod ssl - > sudo service apache2 restart