# 

# a

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

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

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



**BÁO CÁO CHUYÊN ĐỀ**

**TÌM HIỂU BÀI THỰC HÀNH**

**ptit-findfile-decode**

Thực hiện bởi: Nhóm 5

Môn học: Chuyên đề an ninh mạng

Giảng viên hướng dẫn: Đinh Trường Duy

**Hà Nội - 2023**

**Mục lục**

[I. Nội dung và hướng dẫn thực hiện bài thực hành 1](#_Toc912278657)

[1. Mục đích 1](#_Toc1781632266)

[2. Yêu cầu đối với sinh viên 1](#_Toc1529004541)

[3. Nội dung thực hành 1](#_Toc1190000585)

[II. Phân tích , thiết kế bài thực hành 2](#_Toc824627584)

[1. Phân tích yêu cầu bài thực hành 2](#_Toc1803830197)

[2. Thiết kế bài thực hành 3](#_Toc947476280)

[3. Cài đặt và cấu hình máy ảo 6](#_Toc642595455)

[4. Thử nghiệm và đánh giá 10](#_Toc406551422)

[5. Triển khai bài thực hành 12](#_Toc1764268447)

[5.1. Các bước thực hiện up imodule lên DockerHub 12](#_Toc1753074600)

[5.2. Các bước thực hiện up file bài lab lên github 14](#_Toc449165360)

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

Tìm và Giải Mã Nội Dung Tập Tin GIúp sinh viên tìm hiểu được cách thao tác các lệnh trong linux

1. **Yêu cầu đối với sinh viên**

Có kiến thức cơ bản về hệ điều hành Linux

1. **Nội dung thực hành**

Sinh viên thực hiện tải bài lab tại :

*imodule https://github.com/gcndev/ptit-findfile-decode/raw/main/imodule.tar*

* Lưu ý: nếu bị lỗi không chạy được cần update bằng lệnh : *update-labtainer.sh*
* Khởi động bài lab:
  + Vào terminal, gõ:

*labtainer -r ptit-findfile\_decode*

*( chú ý : sinh viên sử dụng mã sinh viên của mình để nhập thông tin 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 khách: ***client***, một cái là đại diện cho máy chủ: ***server.*** Biết rằng 2 máy nằm cùng mạng LAN

* Trên terminal ***server*** sử dụng lệnh “ifconfig”, xác định địa chỉ IP và địa chỉ mạng LAN.
* Sau khi xác định được IP máy server , ở terminal máy ***client*** thực hiện kết nối ssh tới máy ***server*** ( với password mặc định là **ubuntu** ):

*ssh ubuntu@<IP của Server>*

* Sau khi kết nối thành công tới máy ***server***  qua ssh thực hiện tìm file có định dạng sau :
  + - ***Có kích thước 17 byte***
    - ***Có quyền thực thi***
    - ***Là file ẩn***
    - ***Là file mà con người có thể đọc được***

Sử dụng *find* để thực hiện tìm file có đặc điểm như trên ( tham khảo lệnh bằng cách gõ *find -help* )

* Sau khi tìm được file dùng lệnh *cat*  để mở file chứa thông điệp cần tìm

*cat <file cần tìm>*

* Sau khi mở được file ta nhận được một thông điệp được mã hóa bằng base64 , thực hiện giải mã và lưu vào file key.txt ở thư mục decode

*base64 -d <file cần tìm> >> decode/key.txt*

* Sau khi truy cập giải mã được file , ta mở file key.txt trong thư mục decode bằng lệnh

*cat <nhập tên file>*

* Sau khi mở và ta sẽ thấy được thông điệp cần tìm
* Kết thúc bài lab :
  + Trên terminal đầu tiên sử dụng câu lênh sau để kết thúc bài lab:

*stoplab ptit-findfile\_decode*

* + 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 ptit-findfile\_decode*

1. **Phân tích , thiết kế bài thực hành**
2. **Phân tích yêu cầu bài thực hành**

Bài thực hành cần có hai máy tính trong cùng mạng LAN . Trong đó một máy làm máy khách , một máy chủ chứa tập tim mà sinh viên cần tìm ra . sinh viên cần sử dụng máy khách để truy cập vào máy chủ sử dụng dịch vụ ssh bằng cổng và địa chỉ IP đã tìm ra.

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 có thể thực hiện được việc tìm file chứa thông điệp thì hệ thống cần cung cấp khả năng thiết lập một mạng LAN giữa máy chủ và máy khách, đồng thời cung cấp các thư viện cần cho máy chủ và máy khách để sử dụng được dịch vụ ssh . Hệ thống cần ghi lại được các thao tác sử dụng các lệnh linux và ssh của sinh viên thông qua các câu lệnh để tạo ra được kết quả đánh giá . Hệ thống yêu cầu sinh viên nhập email gắn liền với danh tính của sinh viên, và ghi lại thao tác mở tệp phía máy chủ để thực hiện việc cá nhân hóa cho từng sinh viên.

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

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 vài trò máy khách và 1 container mang tên “ server ” đóng vai trò máy chủ , 2 máy đều được mở các cổng cần thiết .

* Tạo mạng LAN “ SOME\_NETWORKS ” có cấu hình : 172.20.0.0/24 và gateway : 172.20.0.1
* Câu hình docker gồm có :
* Client : Lưu cấu hình cho máy khách , trong đó gồm có :
  + Tên máy khách : client
  + Địa chỉ trong mạng LAN : 172.20.0.10
  + Gateway : 172.20.0.1
* Server: lưu cấu hình cho máy chủ, trong đó gồm có:
  + Tên máy khách : client
  + Địa chỉ trong mạng LAN : 172.20.0.20
  + Gateway : 172.20.0.1
* 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, chương trình
  + server : sử dụng các thư viện mặc định hệ thống cùng với cấu hình NETWORK trong đó đã cài sẵn các dịch vụ, chương trình: openssh-server
* docs: lưu phần mô tả hướng dẫn làm bài thực hành cho sinh viên.
  + Các nhiệm vụ cần phải thực hiện để thành công :
    - Sử dụng câu lệnh ssh để kết nối tới máy server , sau khi kết nối thành công hiện từ Welcome!
    - Trên máy khách , sau khi kết nối sử dụng các lệnh linux để tìm file cần tìm
    - Sau khi mở được file , thực hiện lệnh cat để đọc file
    - Sau khi mở được file thực hiện lệnh base64 để giải mã file và lưu vào file key.txt theo yêu cầu
    - Sau khi giải mã và lưu mở file bằng lệnh cat
    - Kết thúc bài lab và đóng gói kết quả
* Instr\_config : lưu cấu hình cho phần nhận kết quả và chấm điểm.
* Thiết lập hệ thống mạng sao cho máy chủ và máy khách cùng một mạng LAN.
* Các thư viện cần cho máy chủ và máy khách để sử dụng được dịch vụ ssh.
* Tạo ra tệp trên máy chủ mang tên key.txt có nội dung như dưới để thực hiện hash md5

# This file is modified when container is created

# The string below will e replaced with a keyed bash

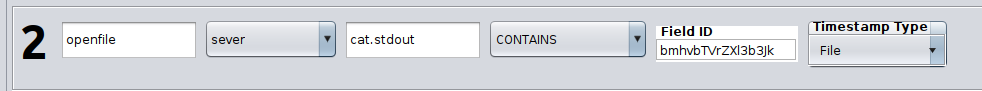
My string is: MY\_STRING

* Để hoàn thành bài thực hành , cần thực hiện các câu lệnh linux và sử dụng được dịch vụ ssh để truy cập máy chủ
* Sau khi hoàn thành bài thực hành, hệ thống cần tự động lưu lại kết quả vào 1 file.
* 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



Một số mục tiêu kết quả cần kiểm tra thể hiện trong hình trên :

* Ssh : thực hiện ssh thành công từ client tới server khi đó sẽ hiện từ “ Welcome ” , từ đó sẽ checkwork thành công

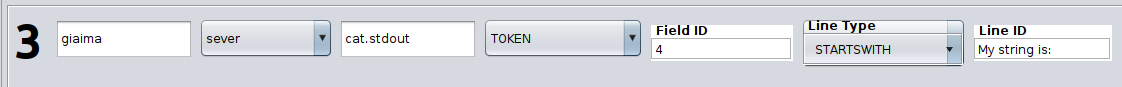


Một số mục tiêu kết quả cần kiểm tra thể hiện trong hình trên :

* Cat : cần thực hiện lệnh cat để mở file thành nhận được nội dung được mã hóa

A screenshot of a computer

Description automatically generated

Sử dụng parameters để tạo ngẫu nhiên mã hash khi nhập email hoặc mã sinh viên vào , thực hiện để cá nhân hóa bài lab , mỗi mã sinh viên sẽ được hash ngẫu nhiên 1 mã hash khác nhau , khi mở file key.txt bằng lệnh cat sẽ checkwork được

Và nhận được thông điệp giải mã

* Sau khi nhận được file đóng gói từ sinh viên, giảng viên sử dụng chức năng chấm điểm để xem kết quả được thiết kế dạng bảng trong đó ghi rõ email của sinh viên thực hiện, từng tiêu chí chấm điểm được ghi nhận ( nếu có chữ “Y” là đã hoàn thành, nếu không có là chưa hoàn thành) và kết luận là sinh viên đã hoàn thành bài thực hành đó hay chưa. Kiểm tra bài thực hành đúng do sinh viên làm bằng cách kiểm tra email

A black background with white text

Description automatically generated

1. **Cài đặt và cấu hình máy ảo**

Cấu hình file máy ảo

* Máy client :

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

* Cấu hình máy server :

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Với task1 chúng ta cài đặt cho Results : Cài đặt Field ID là IP của server để khi client kết nối ssh qua IP của Server ta sẽ bắt được để checkwork

Task 2 : Ở phần Field ID ta sẽ để nội dung của file mà cần tìm , như hình đó là nội dung trong file mà mở

Ở đây để cho việc tìm file khó , ta tạo nhiều file có kích thước giống nhau để làm nhiễu

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Như mô tả , file cần tìm là file ẩn có những đặc điểm

*human-readable*

*17 bytes in size*

*Executable*

Để tạo file ẩn ta sử dụng lệnh : *touch .filename*

File này sẽ không hiện thị ở folder ( nên không thể tìm bằng mắt ) và không thể tìm bằng cách sử dụng lệnh đơn giản như ls

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generatedFile nặng 17 byte , để làm nhiễu đặc điểm này , sử dụng lệnh *: truncate -s 17 filename* để tạo các file nặng 17 btye

A computer screen shot of a black background

Description automatically generated

File có thể thực thi , ở đây dùng lệnh *chmod a+rx filename* để phân quyền đọc và thực thi cho file

A screen shot of a computer

Description automatically generated

File con người có thể đọc được , vậy ta sẽ tạo các file không phải ASCII Text nặng 17 byte và phân quyền đọc và thực thi cho file bằng lệnh

*dd if=/dev/urandom of=mybinaryfile bs=1 count=17*

A computer screen shot of a computer program

Description automatically generated

1. **Thử nghiệm và đánh giá**

Chúng tôi đã xây dựng thành công bài thực hành, dưới đây là các hình ảnh minh họa về bài thực hành

Bước 1 : check ip máy server

A computer screen shot of a program

Description automatically generated

Bước 2 : truy cập ssh từ client tới server

A screenshot of a computer

Description automatically generated

Bước 3 : thực hiện tìm file theo đặc điểm mà bài đưa ra

Những file ẩn , có kích thước 17 byte , có thể thực thi

A screen shot of a computer

Description automatically generated

Loại được khá nhiều tiếp đến tìm theo file dạng con người có thể đọc được ( File dạng ASCII Text )

A screenshot of a computer screen

Description automatically generated

Từ 2 lệnh ta thấy file cần tìm là file *.results* trong thư mục *task8*

Thực hiện mở bằng lệnh *cat*

A close up of white text

Description automatically generated

Bước 4 : Thực hiện giải mã vào chèn thêm vào file key.txt trong thư mục decode và sử dụng lệnh cat để mở file

A screenshot of a computer code

Description automatically generated

Thực hiện checkwork :

A black background with white text

Description automatically generated

1. **Triển khai bài thực hành**

**5.1. Các bước thực hiện up imodule lên DockerHub**

Xác định Registry (tên DockerHub để thiết lập push và tự động pull image khi tải về và sử dụng lab ) bằng cách chọn Edit / Config (registry)

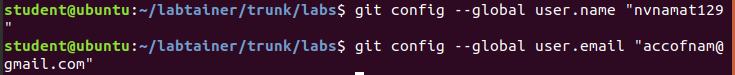
A screenshot of a computer

Description automatically generated

* Chuyển tới thư mục chứa các bài thực hành: **labtainer/trunk/labs**
* Khởi tạo git: **git init** (chỉ khởi tạo một lần, không lặp lại với mỗi lần).
* Sử dụng lệnh để config :

*git config --global user.name "username"*

*git config --global user.email "email"*



* Thêm bài thực hành vào git

git add <tên bài lab>

git commit <tên bài lab> -m “Adding an IModule”

A screen shot of a computer code

Description automatically generated

* Vào thư mục trunk/distrib/ và sử dụng lệnh ./publish.py -d -l <tên bài lab> để đẩy  
  image lên DockerHub
* Lưu ý cần tạo 1 tài khoản dockerhub( Mình sử dụng đăng nhập bằng tài khoản github)
* Sử dụng lệnh *docker login*

Đăng nhập Docker sau đấy sử dụng lệnh

*./publish.py -d -l <tên bài lab>*

*A screenshot of a computer

Description automatically generated*

Sau khi up thành công kết quả sẽ hiển thị trên web : *https://hub.docker.com/repositories/nvnamat129*

A screenshot of a computer

Description automatically generated

* Tạo file IModule tar chứa bài thực hành:

*create-imodules.sh*

Sau đó, copy và lưu lại file imodule.tar và chia sẻ đường dẫn URL tới sinh viên để thực hiện bài thực hành.

**5.2. Các bước thực hiện up file bài lab lên github**

* Thực hiện tạo repository

A screenshot of a computer

Description automatically generated

* Thực hiện upload file Imodule.tar lên github

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Sau khi thực hiện uploads gửi link cho sinh viên : *https://github.com/gcndev/ptit-findfile-decode*
* Thực hiện tải và giải nén bài lab. Di chuyển bài lab sang thư mục /labtainer/trunk/labs
* Sử dụng lệnh khởi động bài lab để thực hiện pull image của bài lab được lưu trữ lên docker

A computer screen shot of a computer code

Description automatically generated