# 

**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 cuộc tấn công bằng Snort

**Hà Nội - 2025**

# MỤC LỤC

[MỤC LỤC 1](#_Toc152579358)

[DANH MỤC CÁC HÌNH VẼ VÀ BẢNG 2](#_Toc152579359)

[1. Nội dung và hướng dẫn thực hiện bài thực hành 3](#_Toc152579360)

[1.1. Mục đích 3](#_Toc152579361)

[1.2. Yêu cầu đối với sinh viên 3](#_Toc152579362)

[1.3. Nội dung thực hành 3](#_Toc152579363)

[2. Phân tích, thiết kế bài thực hành 5](#_Toc152579364)

[2.1. Phân tích yêu cầu bài thực hành 5](#_Toc152579365)

[2.2. Thiết kế bài thực hành 5](#_Toc152579366)

[3. Cài đặt và cấu hình các máy ảo 9](#_Toc152579367)

# DANH MỤC CÁC HÌNH VẼ VÀ BẢNG

[Hình 1.1. Thư mục triển khai và cấu hình bài thực hành 10](#_Toc152856182)

[Hình 1.2. Cấu hình bài máy ảo client 10](#_Toc152856183)

[Hình 1.3. Cấu hình bài máy ảo server 11](#_Toc152856184)

[Hình 1.4. Cấu hình bài máy ảo firewall 12](#_Toc152856185)

[Hình 1.5. File treataslocal client 12](#_Toc152856186)

[Hình 1.6. Sinh viên thực hiện quét cổng bằng công cụ nmap. 13](#_Toc152856187)

[Hình 1.7. Sinh viên thực hiện tạo file user bằng crunch. 13](#_Toc152856188)

[Hình 1.8. Sinh viên tạo file password bằng crunch. 13](#_Toc152856189)

[Hình 1.9. Sinh viên cấu hình snort. 14](#_Toc152856190)

[Hình 1.10. Sinh viên thực hiện viết luật để phát hiện tấn công brute force. 14](#_Toc152856191)

[Hình 1.11. Sinh viên thực hiện tấn công bằng hydra. 14](#_Toc152856192)

[Hình 1.12. Sinh viên thực hiện ssh 15](#_Toc152856193)

[Hình 1.13. Snort phát hiện tấn công và cảnh báo. 15](#_Toc152856194)

[Hình 1.14. Sinh viên cấu hình iptables chặn lưu lượng cổng 22. 16](#_Toc152856195)

[Hình 1.15. Luật snort phát hiện tấn công SQLinj 16](#_Toc152856196)

[Hình 1.16. Tấn công SQL Injection. 16](#_Toc152856197)

[Hình 1.17. Snort phát hiện và cảnh báo khi bị tấn công SQL Injection. 16](#_Toc152856198)

[Hình 1.18. Luật snort phát hiện tấn công XSS 17](#_Toc152856199)

[Hình 1.19. Tấn công XSS. 17](#_Toc152856200)

[Hình 1.20. Tấn công XSS. 17](#_Toc152856201)

[Hình 1.21. Snort phát hiện và cảnh báo. 17](#_Toc152856202)

[Hình 1.22. Kết quả sau khi hoàn thành bài lab. 18](#_Toc152856203)

[Bảng 1.1. Các tiêu chí chấm điểm 6](#_Toc152783132)

[Bảng 1.2. Kết quả chấm điểm 8](#_Toc152783133)

[Bảng 1.3. Kết quả chấm điểm 9](#_Toc152783134)

[Hình 4.1.Xác định Registry 18](#_Toc152783135)

[Hình 4.3. Đẩy image lên Dockerhub 19](#_Toc152783136)

[Hình 4.5 Tạo file IModule tar chứa bài thực hành 20](#_Toc152783137)

# Nội dung và hướng dẫn thực hiện bài thực hành

## 1.1. Mục đích

- Giúp sinh viên hiểu về công cụ snort

- Tìm hiểu về bảo mật mạng

- Thực hành với công cụ nmap, crunch, snort.

- Nâng cao kỹ năng phân tích và giải quyết vấn đề

- Tăng cường ý thức về trách nhiệm và đạo đức

## 1.2. Yêu cầu đối với sinh viên

* Có kiến thức cơ bản về hệ điều hành Linux, mô hình mạng khách/chủ.

## 1.3. Nội dung thực hành

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

*labtainer -r ptit-snort-detect*

*(chú ý: sinh viên sử dụng email stu.ptit.edu.vn 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 ba terminal ảo sẽ xuất hiện, các terminal này đại diện cho máy ảo để sinh viên tiến hành thực hiện trên đó.

* Sinh viên tiến hành quét cổng mở của máy server bằng công cụ nmap:

*sudo nmap 172.25.0.3 -sS*

* Sinh viên tạo ra hai file, một file trong đó là danh sách các tài khoản thông thường được sử dụng và một file là danh sách các mật khẩu hay được dùng:

*crunch <số lượng ít nhất của tài khoản> <số lượng kí tự nhiều nhất> -o <tên file>*

*crunch <số lượng ít nhất của mật khẩu> <số lượng kí tự nhiều nhất> -o <tên file>*

* Sinh viên thực hiện tạo file luật và thêm luật snort trên máy server để phát hiện các cuộc tấn công brute force theo đường dẫn */etc/snort/rules/<tentapluat>*
* Sinh viên tiến hành tấn công bằng hydra trên máy client

*hydra -L <tên file tài khoản> -P <tên file mật khẩu> 172.25.0.3 ssh*

* Sau khi bruteforce thành công có được tài khoản, mật khẩu từ máy client sinh viên thực hiện ssh vào máy server:

*ssh <username>@172.25.0.3*

* Sau khi ssh thành công sinh viên thực hiện đọc file:

*cat /root/filetoview.txt*

* Sinh viên cấu hình iptables trên máy firewall để chặn lưu lượng cổng 22
* Sinh viên thực hiện tạo file luật snort trên máy server để phát hiện cuộc tấn công XSS theo đường dẫn */etc/snort/rules*
* Sinh viên thêm luật snort trên máy server để phát hiện cuộc tấn công SQLinjection theo đường dẫn thư mục */etc/snort/rules/<tentapluat>*
* Sinh viên truy cập vào website theo địa chỉ http://*172.25.0.3/dvwa* và tiến hành tấn công SQLInjection
* Sau khi tấn công snort sẽ phát hiện và đưa ra cảnh báo
* Sinh viên thực hiện tạo file luật trên máy server để phát hiện cuộc tấn công XSS theo đường dẫn */etc/snort/rules*
* Sinh viên thực hiện thêm luật snort trên máy server để phát hiện cuộc tấn công XSS theo đường dẫn thư mục */etc/snort/rules/<tentapluat>*
* Sinh viên truy cập vào website theo địa chỉ http://*172.25.0.3/dvwa* và tiến hành tấn công XSS
* Sau khi tấn công snort sẽ phát hiện và đưa ra cảnh báo

# Phân tích, thiết kế bài thực hành

## 2.1. Phân tích yêu cầu bài thực hành

Bài thực hành cần có ba máy tính để sinh viên thực hành, một máy làm máy client, một máy làm server, một máy làm tưởng lửa.

Trên máy client cần cài đặt công cụ nmap, crunch để sinh viên làm bài. Để hoàn thành, sinh viên cần thực hiện các yêu cầu trên.

Trên máy server được cài đặt trang web chứa lỗi DVWA, snort để sinh viên có thể tiến hành tấn công và phát thông báo qua snort.

Hệ thống sẽ ghi lại các thao tác của sinh viên khi thực hành. Từ kết quả ghi lại đó sẽ đánh giá được sinh viên đã làm đúng hay chưa đúng so với yêu cầu của bài lab.

Để bắt đầu bài thực hành, sinh viên cần phải sử dụng các câu lệnh khởi tạo (labtainer -r <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

* Cấu hình docker gồm có:
  + client: lưu cấu hình cho máy khách, trong đó gồm có:
    - Tên máy : client
    - Cài đặt các công cụ cần thiết
  + server: lưu cấu hình cho máy khách, trong đó gồm có:
    - Tên máy : server
    - Cài đặt các công cụ cần thiết
  + firewall: lưu cấu hình cho máy khách, trong đó gồm có:
    - Tên máy : firewall
    - Cài đặt các công cụ cần thiết
  + config: lưu cấu hình hoạt động của hệ thống
  + dockerfiles: mô tả cấu hình
    - client: tên của máy và một số thông tin cơ bản
    - server: tên của máy, một số thông tin khác được đưa vào để cài đặt ví dụ như trang web DVWA, snort
    - filewall: tên của máy, một số thông tin khác được đưa vào để cài đặt ví dụ như iptables
  + docs: lưu phần mô tả hướng dẫn làm bài thực hành cho sinh viên.
    - Các nhiệm vụ cần phải thực hiện để thực hành thành công:
      * Dùng nmap quét các cổng đang mở
      * Tạo được file user và pasword
      * Thực hiện tấn công
      * Chặn lưu lượng qua cổng bằng iptables
      * Dùng snort để cảnh báo các cuộc tấn công brute force và XSS, SQLinjection
  + instr\_config: lưu cấu hình cho phần nhận kết quả và chấm điểm.
* 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, sự kiện được mô tả và cấu hình như bảng 1.1, 1.2, 1.3:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ports\_ok | iptables\_ok | creuserlist | crepasslist | hydra\_attack | sshview |

|  |  |  |
| --- | --- | --- |
| bruforSSH | SQLinjection | XSS |

Các tiêu chí chấm điểm

* port\_ok: quét thành công các cổng đang mở
* createuserlist: tạo thành công file user
* createpasslist: tạo thành công file password
* hydrra\_attack: thực hiện thành công tấn công
* sshview: tiến hành ssh đến máy server bằng tài khoản mật khẩu tìm được và cat ra file bí mật
* bruteforceSSH: thực hiện tấn công và snort đưa ra cảnh báo
* iptable\_ok: thực hiện chặn thành công lưu lượng qua cổng chỉ định
* SQLinjection: tấn công và snort phát hiện, đưa ra cảnh báo
* XSS: tấn công XSS và snort đưa ra cảnh báo

Bảng chấm điểm các task:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Result Tag** | **Container** | **File** | **Field Type** | **Field ID** | **Line Type** | **Line ID** |
| \_ssh\_open | **Client** | Nmap.stdout | TOKEN | 2 | STARTWITH | 22/tcp |
| \_telnet\_open | **Client** | Nmap.stdout | TOKEN | 2 | STARTWITH | 23/tcp |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Result Tag** | **Container** | **File** | **Field Type** | **Field ID** | **Timestamp Type** |
| Createuserlist | Client | Crunch.stdin | Contrains | User.txt | file |
| Createpasslist | Client | Crunch.stdin | Contrains | pass.txt | file |
| Hydra\_attack | Client | Hydra.stdout | Contrains | Successfully completed | File |
| BruteforceSSH | Server | Snort.stdout | Contrains | Failed SSH logins | File |
| SQLInjection | Server | Snort.stdout | Contrains | SQL Injection | File |
| XSS | Server | Snort.stdout | Contrains | XSS | file |

Kết quả chấm điểm

Nội dung Goals của bài lab

\_ssh matchany string\_equal \_ssh\_open Literal open

\_telnet matchany string\_equal \_telnet\_open Literal open

Ports\_ok boolean (\_ssh and \_telnet)

Iptables\_ok boolean (\_telnet and\_not \_ssh)

* 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 (xem bảng 1.3).

Kết quả chấm điểm

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| port\_ok | iptables\_ok | creuserlist | crepasslist | hydra\_attack | shhview | bruforSSH | SQLInjection | XSS |
| Y | Y | Y | Y | Y | Y | Y | Y | Y |

# Cài đặt và cấu hình các máy ảo

Trình bày nội dung các file cấu hình của hệ thống tại đây, trong thư mục

A screenshot of a computer

Description automatically generated

Thư mục triển khai và cấu hình bài thực hành

Cấu hình file máy ảo:

A screenshot of a computer

Description automatically generated

Cấu hình bài máy ảo client

A screenshot of a computer program

Description automatically generated

Cấu hình bài máy ảo server

A screenshot of a computer program

Description automatically generated

Cấu hình bài máy ảo firewall

Tạo file treataslocal

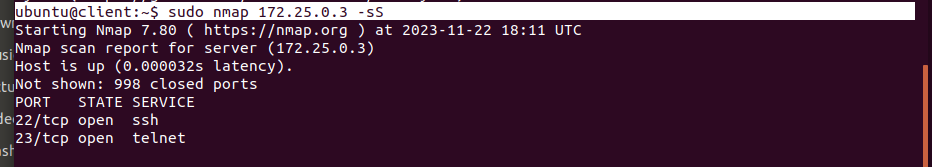
A screenshot of a computer

Description automatically generated

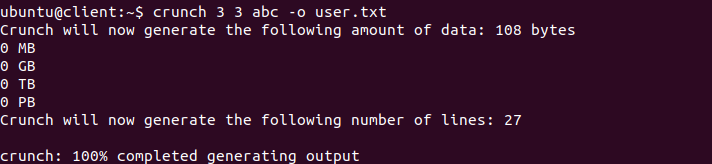
File treataslocal client

**Thử nghiệm và đánh giá**

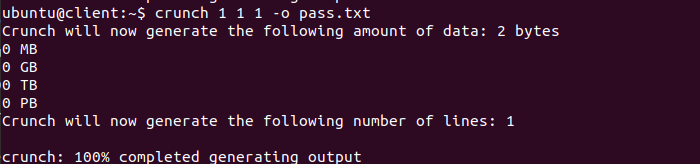
Chúng tôi đã xây dựng thành công bài thực hành, dưới đây là các hình ảnh minh họa về bài thực hành:



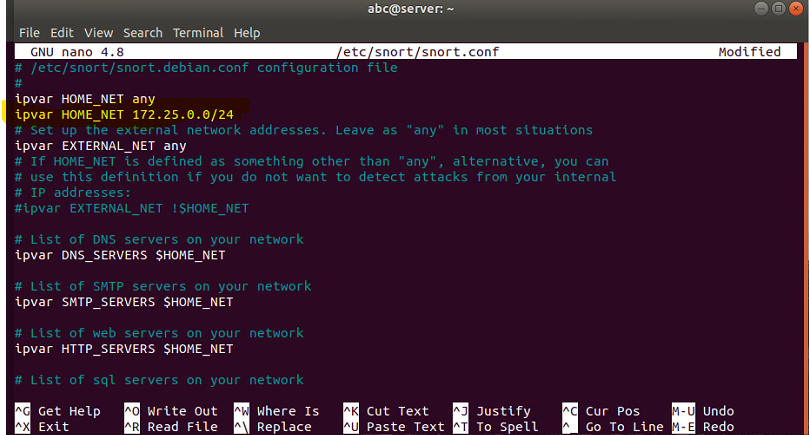
Sinh viên thực hiện quét cổng bằng công cụ nmap.



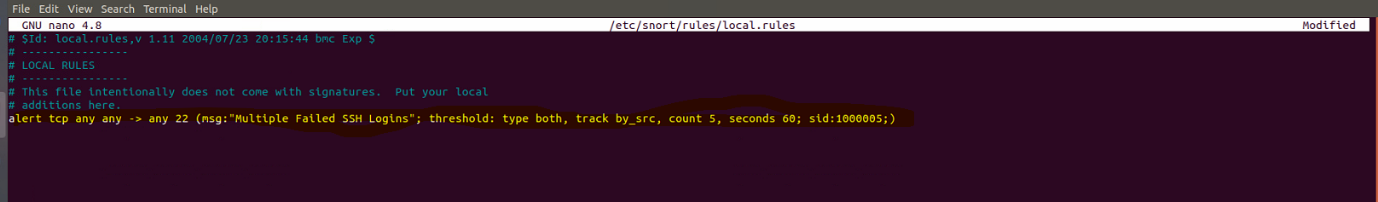
Sinh viên thực hiện tạo file user bằng crunch.



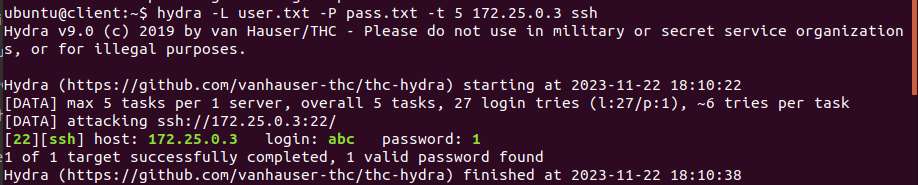
Sinh viên tạo file password bằng crunch.



Sinh viên cấu hình snort.



Sinh viên thực hiện viết luật để phát hiện tấn công brute force.

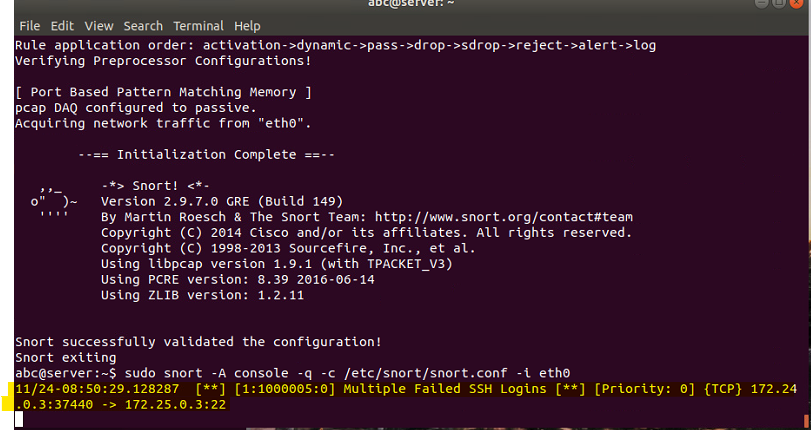


Sinh viên thực hiện tấn công bằng hydra.

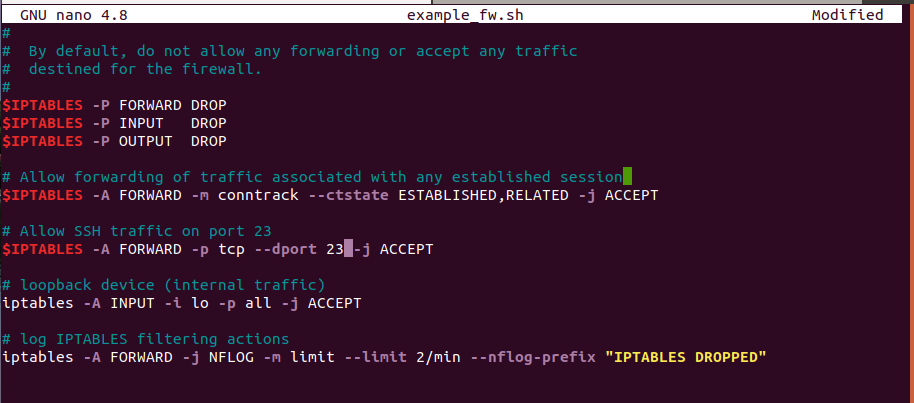
A computer screen shot of a computer program

Description automatically generated

Sinh viên thực hiện ssh



Snort phát hiện tấn công và cảnh báo.



Sinh viên cấu hình iptables chặn lưu lượng cổng 22.

A screenshot of a computer

Description automatically generated

Luật snort phát hiện tấn công SQLinj

A screenshot of a computer

Description automatically generated

Tấn công SQL Injection.



Snort phát hiện và cảnh báo khi bị tấn công SQL Injection.

A screenshot of a computer

Description automatically generated

Luật snort phát hiện tấn công XSS

A screenshot of a computer

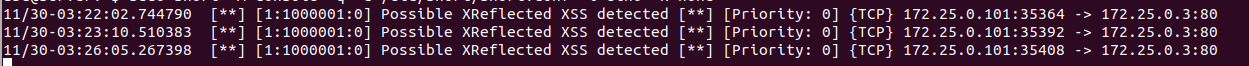
Description automatically generated

Tấn công XSS.

A screenshot of a computer

Description automatically generated

Tấn công XSS.



Snort phát hiện và cảnh báo.



Kết quả sau khi hoàn thành bài lab.

# Triển khai bài lab

* 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

Hình 4.1.Xác định Registry

Labtainer/trunk/labs git config --global user.name "Quandna"

git config --global user.email "quanndptit@gmail.com"

* Thêm bài thực hành vào git

A screenshot of a computer screen

Description automatically generated

*Hình 4.2.* Thêm bài thực hành vào git

* Vào thư mục trunk/distrib/ và sử dụng lệnh ./publish.py -d -l ptit-snort-detect 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 ptit-snort-detect

A computer screen shot of a computer program

Description automatically generated

Hình 4.3. Đẩy image lên Dockerhub

* Sau khi hoàn tất, image của các vùng chứa (container) bao gồm attacker và server  
  sẽ được đẩy lên https://hub.docker.com/repositories/quanndptit

A screenshot of a computer

Description automatically generated

Hình 4.4.Các image của bài lab được lưu trữ trên DockerHub sẽ tự động pull về khi tải và khởi động bài lab lần đầu

* Tạo file IModule tar chứa bài thực hành

A computer screen shot of a program

Description automatically generated

*Hình 4.5* Tạo file IModule tar chứa bài thực hành

File này nằm ở Labtainer/trunk/Imodule

Kéo file này ra Imodule.tar hoặc copy ra máy thật .

Trên git hub tạo một kho lữu trữ mới và Tạo Phát hành và đẩy file tar này lên phần phát hành

A screenshot of a computer

Description automatically generated

Hình 4.6: file tar được đẩy lên git

* Tạo file IModule tar chứa bài thực hành

A screenshot of a computer

Description automatically generated

*Hình 4.7:* Tạo file IModule tar chứa bài thực hành

* Thực hiện đẩy lên github lưu trữ bài lab để sinh viên có thể tải về thực hiện

A screenshot of a computer

Description automatically generated

Hình 4.8.Thực hiện đẩy lab lên github

* Sinh viên sẽ sử dụng lệnh:
* 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 white text

Description automatically generated

Hình 4.9. Hệ thống tự động pull image được lưu trữ trên docker về để chạy lab