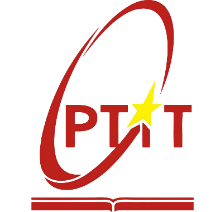
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-------🙞🙜🕮🙞🙜-------



**Báo cáo bài thực hành**

**TÌM HIỂU BÀI THỰC HÀNH**

**PTIT – PEN\_TOOL\_MITRE\_** **T1105**

Giảng viên: Nguyễn Ngọc Điệp

Môn học: Đồ án tốt nghiệp

Sinh viên: Đinh Việt Anh

Mã sinh viên: B20DCAT005

**Hà Nội - 2024**

Mục lục

[Bài thực hành: Tìm hiểu về Remote File Copy (pen\_tool\_mitre\_t1105) 5](#_Toc186723953)

[1. Giới thiệu chung về bài thực hành: 5](#_Toc186723954)

[2. Nội dung và hướng dẫn bài thực hành 5](#_Toc186723955)

[3. Phân tích yêu cầu bài thực hành 7](#_Toc186723956)

[4. Thiết kế bài thực hành 7](#_Toc186723957)

[5. Cài đặt và cấu hình các máy ảo 9](#_Toc186723958)

[6. Tích hợp và triển khai 12](#_Toc186723959)

[7. Thử nghiệm và đánh giá 16](#_Toc186723960)

DANH MỤC CÁC HÌNH VẼ

[Hình 1. Giao diện Labedit của bài lab 10](#_Toc186723961)

[Hình 2. Cài đặt phần Result 11](#_Toc186723962)

[Hình 3. Dockerfiles của máy victim 12](#_Toc186723963)

[Hình 4. Dockerfiles của attacker 13](#_Toc186723964)

[Hình 5. Đăng nhập trong Labtainer 14](#_Toc186723965)

[Hình 6. Khởi tạo quá trình lưu trữ trên dockerhub 14](#_Toc186723966)

[Hình 7. Khai báo thông tin tài khoản 15](#_Toc186723967)

[Hình 8. Thêm bài lab muốn lưu trữ 15](#_Toc186723968)

[Hình 9. Đăng nhập tài khoản dockerhub 15](#_Toc186723969)

[Hình 10. Quá trình tải bài lab lên dockerhub 16](#_Toc186723970)

[Hình 11. Lưu trữ bài lab trên dockerhub 16](#_Toc186723971)

[Hình 12. Lưu trữ bài lab trên github 16](#_Toc186723972)

[Hình 13. Tải bài lab về 17](#_Toc186723973)

[Hình 14. Chạy lab từ imodule 17](#_Toc186723974)

[Hình 15. Kết nối đến máy victim và kiểm tra file cần sao chép 18](#_Toc186723975)

[Hình 16. Sao chép file từ máy victim về máy attacker 18](#_Toc186723976)

[Hình 17. Kiểm tra kết quả sao chép file trên máy attacker 18](#_Toc186723977)

[Hình 18. Đánh giá kết quả bài thực hành. 18](#_Toc186723978)

DANH MỤC BẢNG BIỂU

[Bảng 1. Bảng Result 9](#_Toc186723979)

[Bảng 2. Bảng Goal 10](#_Toc186723980)

DANH MỤC CÁC TỪ VIẾT TẮT

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

Bài thực hành: Tìm hiểu về Remote File Copy (pen\_tool\_mitre\_t1105)

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

Bài thực hành **"Tìm hiểu về Remote File Copy thông qua công cụ rsync"** được thiết kế nhằm giúp sinh viên hiểu rõ cách thức các tác nhân độc hại có thể lợi dụng công cụ truyền tải tệp từ xa như **rsync** để sao chép dữ liệu giữa máy mục tiêu và máy tấn công trong các kịch bản xâm nhập. Đây là một bước khởi đầu quan trọng để sinh viên làm quen với kỹ thuật **Remote File Copy** trong hệ thống Linux và các biện pháp phòng chống tương ứng.

Trong bài thực hành này, sinh viên sẽ làm việc trong một môi trường giả lập gồm hai container:

* **Victim** (hệ thống mục tiêu): Máy nạn nhân mà kẻ tấn công đã truy cập được qua SSH.
* **Attacker** (hệ thống tấn công): Được sử dụng để kết nối và sao chép tệp từ máy Victim thông qua công cụ rsync.

Thông qua các bước thực hành, sinh viên sẽ hiểu rõ cách công cụ **rsync** có thể được khai thác để sao chép tệp từ xa, từ đó mở rộng kiến thức về kỹ thuật truyền tải dữ liệu trong các kịch bản tấn công và phòng thủ.

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

2.1 Mục đích

Bài thực hành giúp sinh viên tìm hiểu cách công cụ **rsync** có thể bị khai thác để thực hiện **sao chép tệp từ xa** giữa hệ thống mục tiêu (Victim) và hệ thống tấn công (Attacker). Sinh viên sẽ thực hiện việc kết nối SSH và sử dụng **rsync** để sao chép tệp từ máy mục tiêu về máy tấn công, qua đó hiểu rõ cơ chế hoạt động của **rsync** và các rủi ro bảo mật tiềm ẩn khi cấu hình không an toàn.

Bài thực hành cung cấp kiến thức thực tế về cách thức các tác nhân độc hại lợi dụng **rsync** để chiếm đoạt dữ liệu hoặc duy trì kết nối với máy mục tiêu, đồng thời giúp sinh viên nắm vững kỹ thuật này trong quá trình tấn công và phòng thủ hệ thống.

* 1. Yêu cầu đối với sinh viên

Sinh viên cần có kiến thức cơ bản về:

* Hệ điều hành **Linux** và quản lý hệ thống.
* **SSH** để kết nối từ xa giữa hai hệ thống.
* Công cụ **rsync** và cách sử dụng lệnh để sao chép tệp từ xa.
* Kiến thức cơ bản về kiểm tra tệp/thư mục và xác minh tính toàn vẹn dữ liệu.
  1. Nội dung thực hành

Khởi động bài lab:

Vào terminal, gõ:

*Labtainer -r <tên bài lab>*

*(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: ***attacker***, một cái là đại diện cho máy nạn nhân: ***victim***.

Trên terminal ***attacker*** thực hiện kết nối SSH đến máy ***victim*** với password 123456:

*ssh ubuntu@172.0.0.2*

Sau khi ***attacker*** dùng ssh kết nối thành công đến máy ***victim,*** thực hiện các bước tiếp theo***.***

Kiểm tra nội dung file cần sao chép

*cat secret*

Sau khi xác định file cần sao chép, ***attacker*** sử dụng rsync sao chép file về máy ***attacker***

*Rsync ubuntu@172.0.0.2:/home/ubuntu/secret /home/ubuntu*

Sau khi sao chép thành công, ***attacker*** kiểm tra file đã sao chép được.

Kết thúc bài lab:

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

*stoplab <tên bài lab>*

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 <tên bài lab>*

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 được build dựa trên package .kali tích hợp công cụ rsync. Để hoàn thành bài thực hành sinh viên cần sử dụng ssh để kết nối tới máy victim, kiểm tra file cần sao chép và thực hiện sao chép về máy attacker.

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 “attacker” đóng vai trò máy tấn công và 1 container mang tên “victim” đóng vai trò máy nạn nhân.

Tạo mạng LAN có cấu hình: 172.0.0.0/24 và gateway: 172.0.0.101

Cấu hình docker gồm có:

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

Tên máy: attacker

Địa chỉ trong mạng LAN: 172.0.0.3

Gateway: 172.0.0.101

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

Tên máy: victim

Địa chỉ trong mạng LAN: 172.0.0.2

Gateway: 172.0.0.101

config: lưu cấu hình hoạt động của hệ thống

dockerfiles: mô tả cấu hình của 2 container: attacker và victim, trong đó:

attacker: sử dụng các thư viện mặc định hệ thống và package .kali tích hợp sẵn rsync giúp thực hiện các lệnh và chức năng của tool này.

victim: sử dụng các thư viện mặc định hệ thống và package .kali tích hợp sẵn rsync giúp thực hiện các lệnh và chức năng của tool này.

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:

* ***Attacker*** kết nối thành công đến máy victim
* ***Attacker*** kiểm tra file cần sao chép
* ***Attacker*** sao chép file từ máy victim về máy ***attacker***
* ***Attacker*** kiểm tra file đã sao chép thành công.
* Kết thúc bài lab và đóng gói kết quả.

.

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 1,2,3:

Bảng 1. Bảng Result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Result Tag | Container | File | Field Type | Field ID | Timestamp Type | LINE ID |
| check\_connection | attacker | .bash\_history | CONTAINS | ssh ubuntu@172.0.0.2 | File |  |
| check\_file | victim | .bash\_history | CONTAINS | cat secret | File |  |
| copy\_file | attacker | .bash\_history | CONTAINS | ubuntu@172.0.0.2 | File |  |
| check\_result | attacker | .bash\_history | CONTAINS | cat secret | File |  |

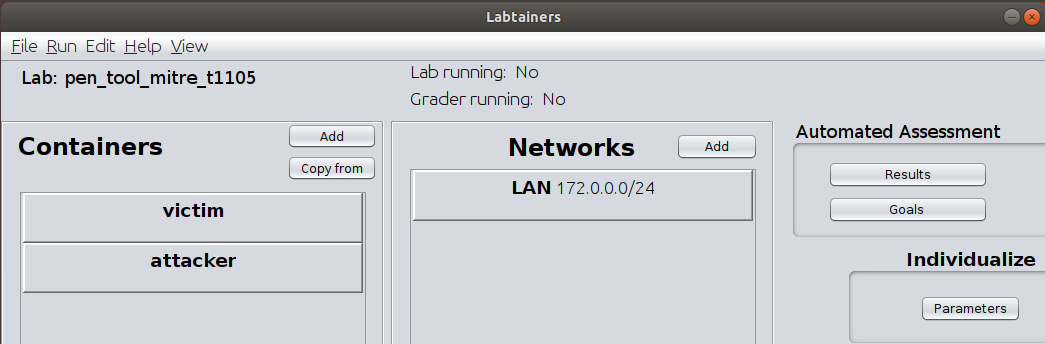
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Goal ID | Goal Type | Operator | Result Tag | Answer Type | Parameter |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Bảng 2. Bảng Goal

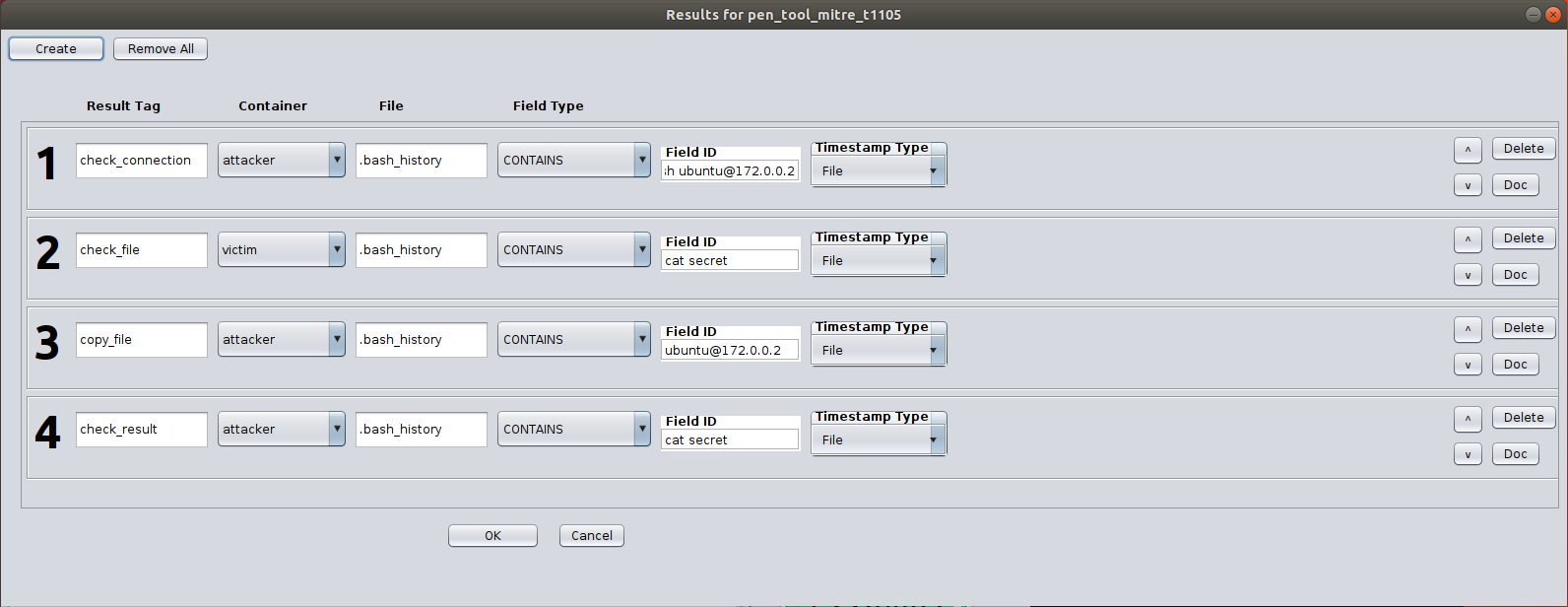
Bảng 3. Bảng Parameter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Param ID | Operator | File name | Symbol | Step | Hashstring |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

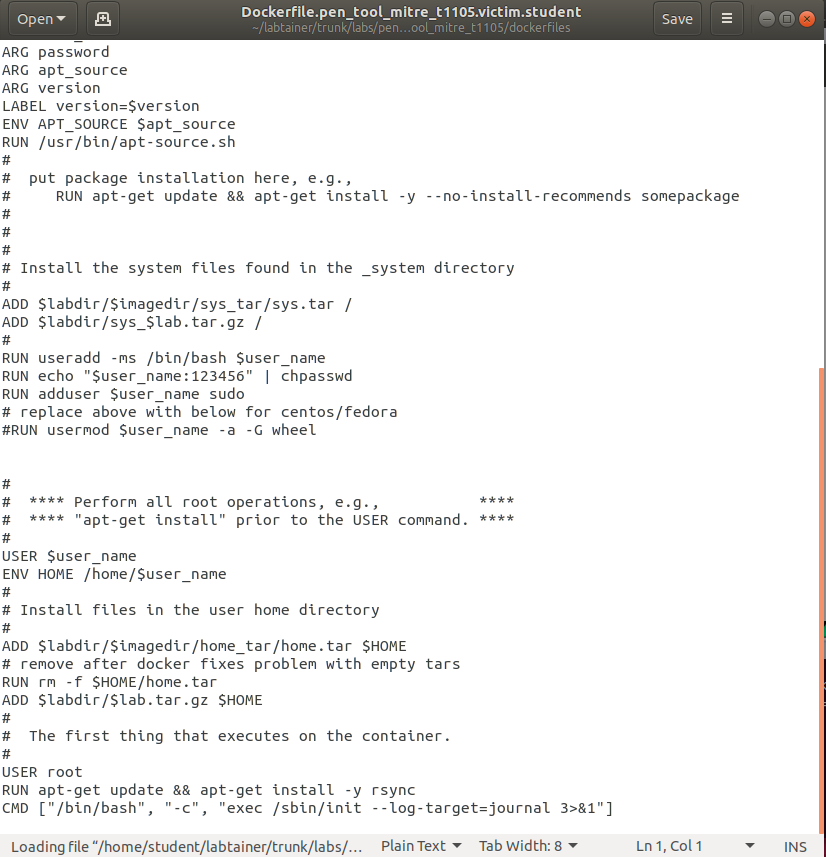
1. Cài đặt và cấu hình các máy ảo



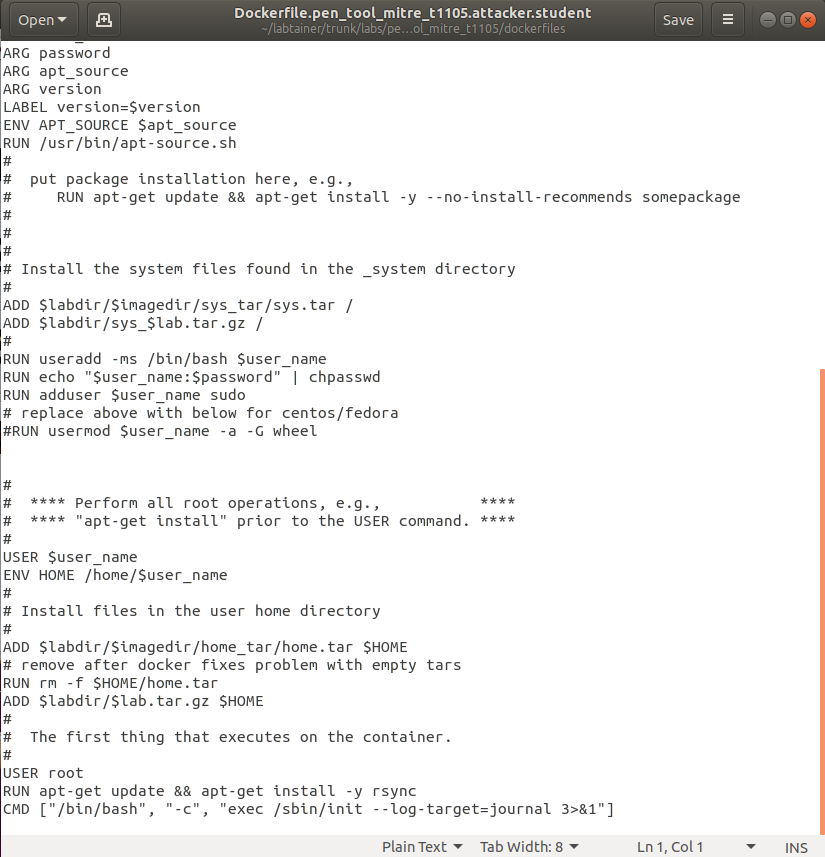
Hình 1. Giao diện Labedit của bài lab



Hình 2. Cài đặt phần Result



Hình 3. Dockerfiles của máy victim



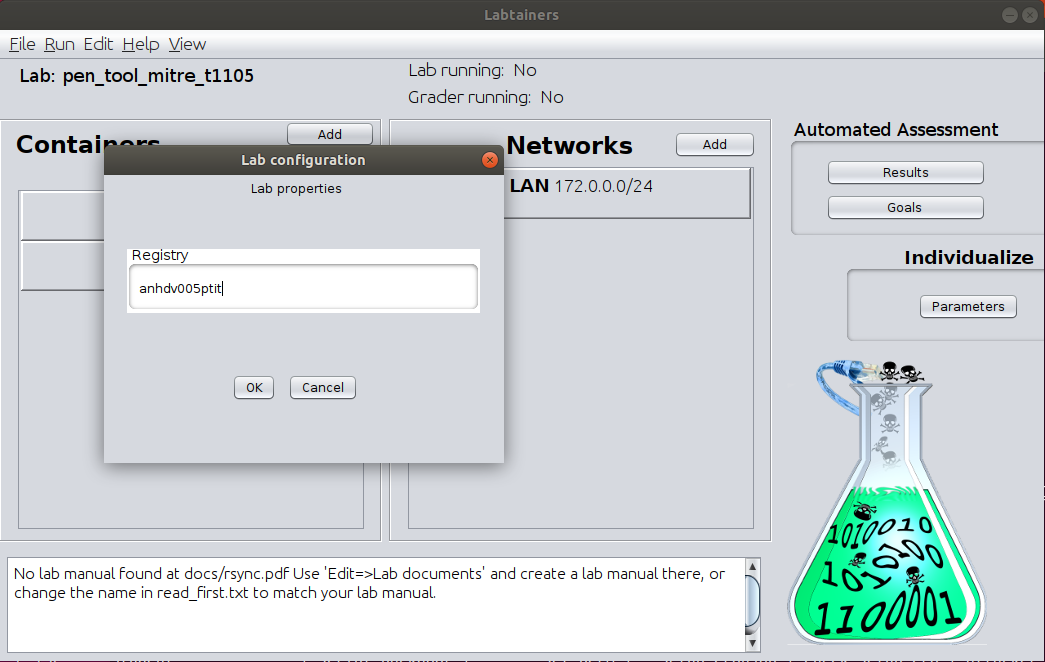
Hình 4. Dockerfiles của attacker

1. Tích hợp và triển khai

Bài thực hành đã được triển khai như sau:

* + - * 1. Docker Hub

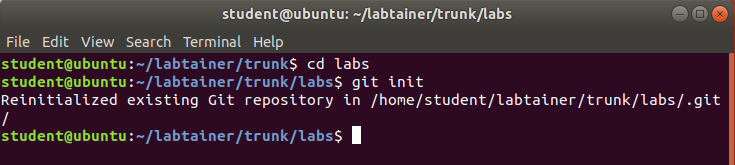
Tạo tài khoản trên trang https://hub.docker.com trên giao diện labtainer chọn mục Edit/Config (registry) đănh nhập tên tài khoản hiển thị trên dockerhub và Build only lại.



Hình 5. Đăng nhập trong Labtainer

Khởi tạo quá trình lưu trữ trên dockerhub từ đường dẫn ~labtainer/trunk/labs với lệnh:

***git init*** (chỉ nhập 1 lần duy nhất)



Hình 6. Khởi tạo quá trình lưu trữ trên dockerhub

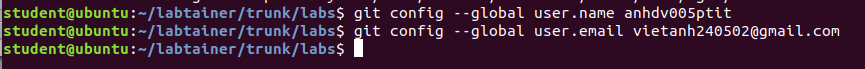
Khai báo các thông số để kết nối đến dockerhub với lệnh:

***git config --global user.name “your\_username”***

(với your\_username: tên hiển thị trên git)

***git config --global user.email your\_email@example.com***

(với your\_email@example.com: tài khoản đăng nhập dockerhub)

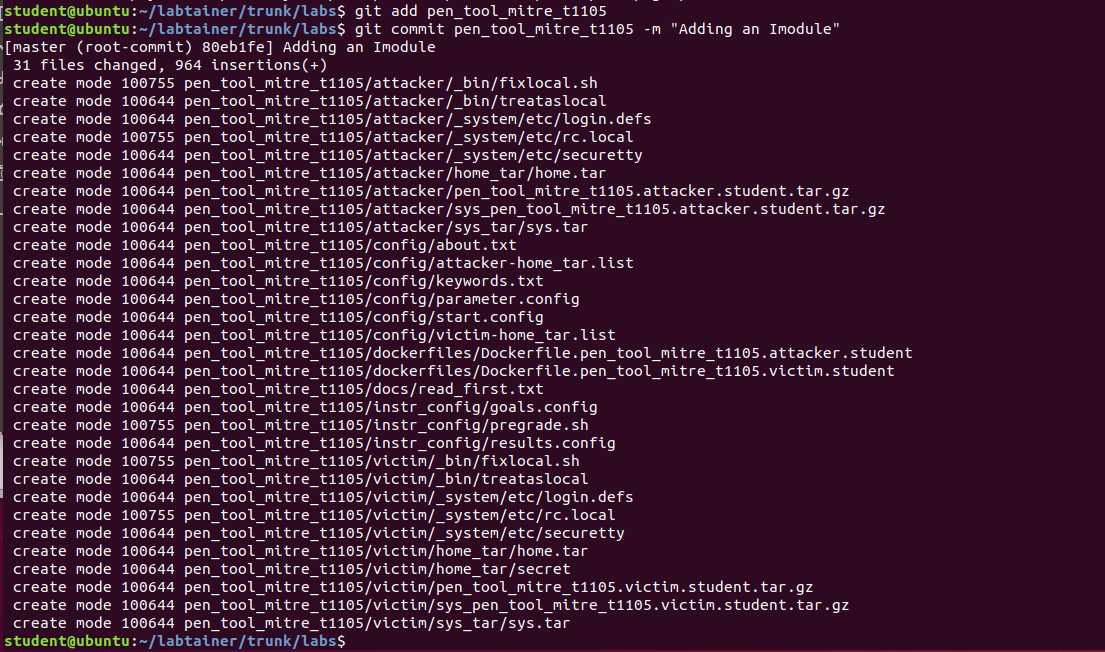


Hình 7. Khai báo thông tin tài khoản

Thêm bài lab muốn lưu trữ vào git theo lệnh:

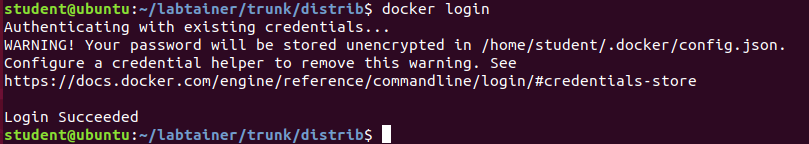
**git add <tên bài lab>**

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



Hình 8. Thêm bài lab muốn lưu trữ

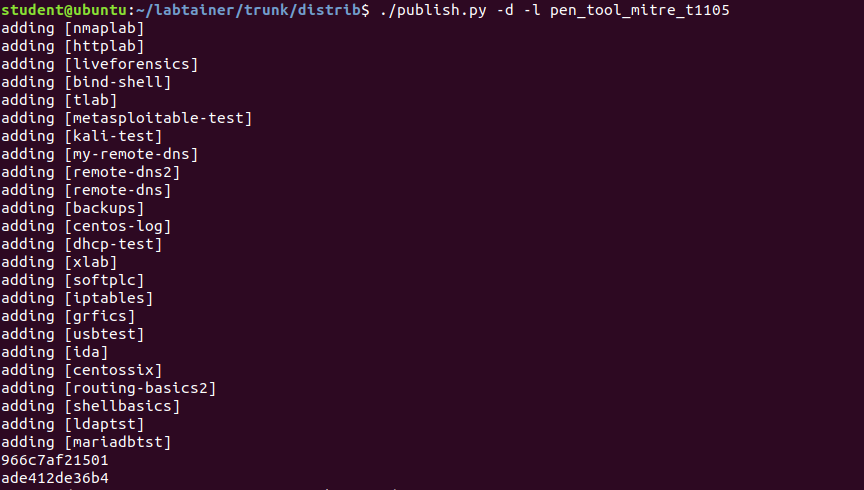
Chuyển đến thư mục ~trunk/distrib, sử dụng lệnh docker login để đăng nhập docker trên Labtainer



Hình 9. Đăng nhập tài khoản dockerhub

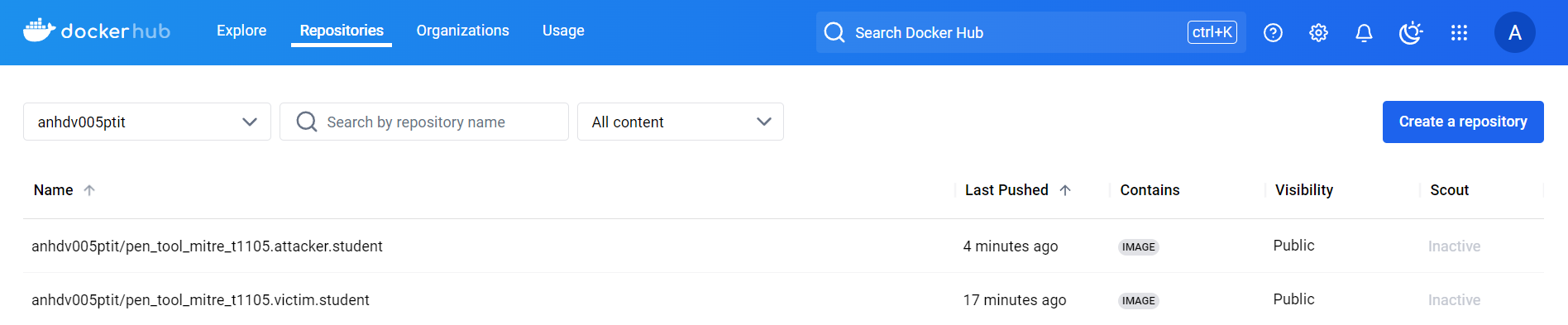
Đẩy bài lab lên dockerhub bằng cách sử dụng lệnh:

***./publish.py -d -l <tên bài lab>***



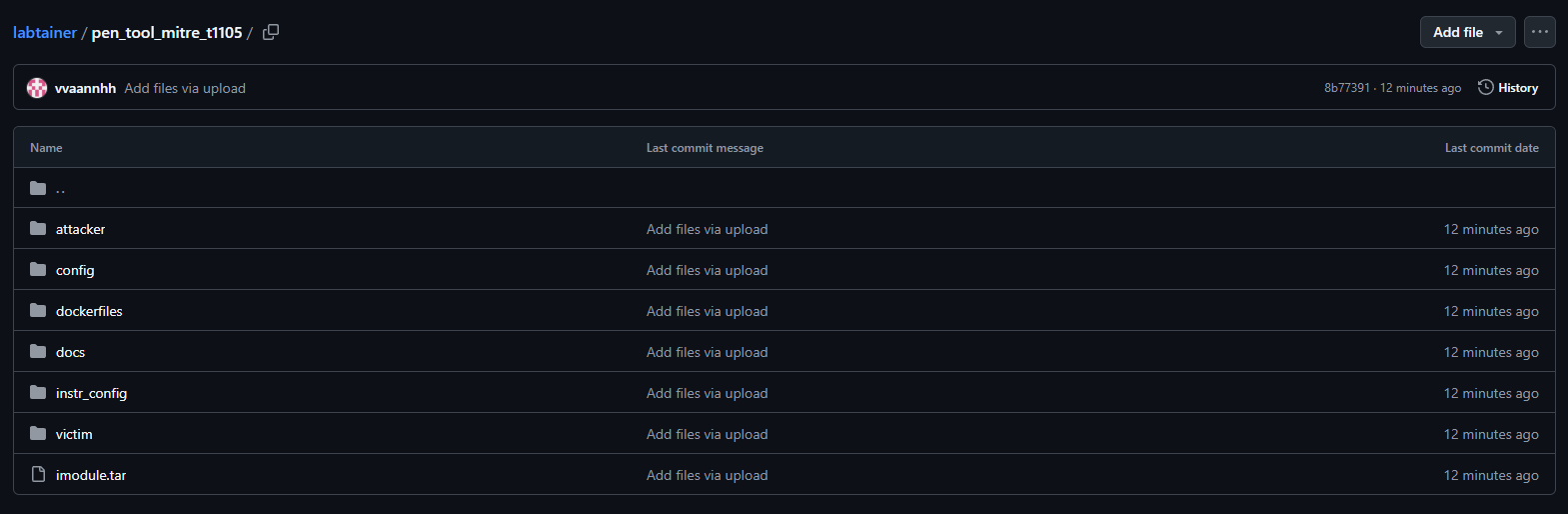
Hình 10. Quá trình tải bài lab lên dockerhub

Quá trình thành công các bài lab đã được đẩy lên quản lý trên docker hub



Hình 11. Lưu trữ bài lab trên dockerhub

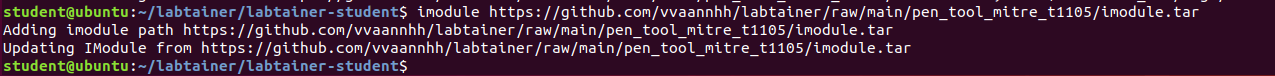
* + - * 1. Github



Hình 12. Lưu trữ bài lab trên github

Sinh viên tiến hành tải bài lab theo đường dẫn được cung cấp theo lệnh

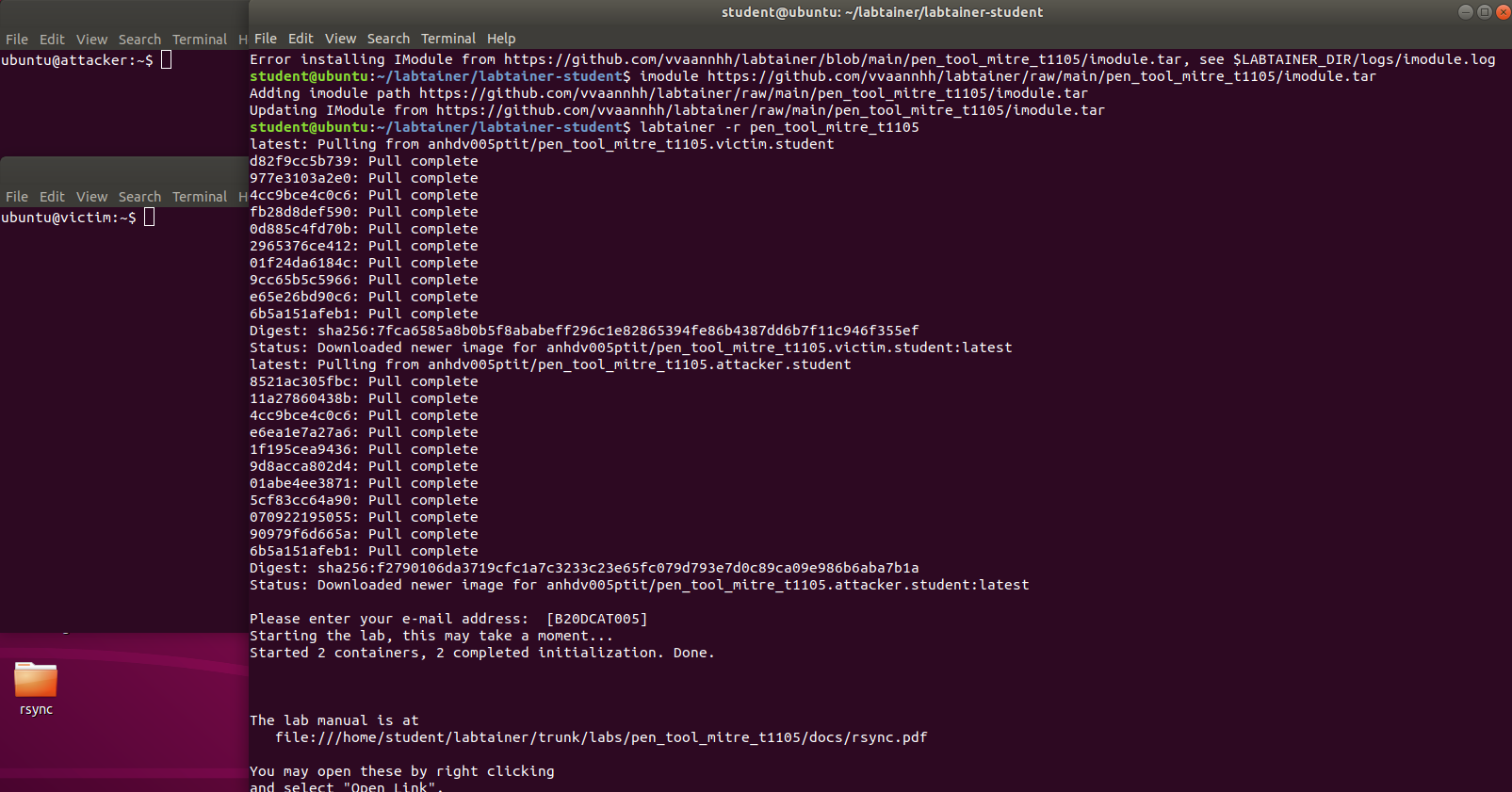
***imodule <đường dẫn>***



Hình 13. Tải bài lab về

Load bài lab từ imodule

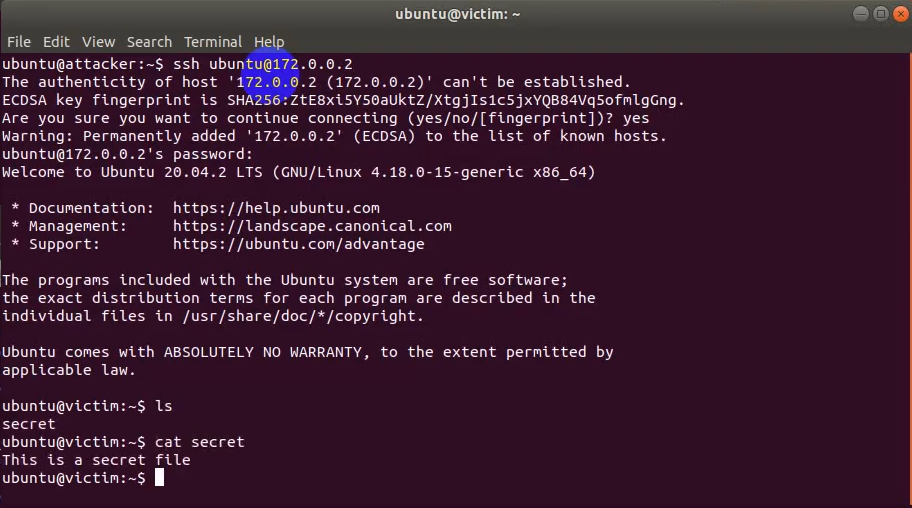
***labtainer -r <tên bài lab>***



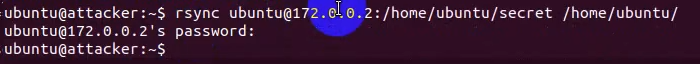
Hình 14. Chạy lab từ imodule

1. Thử nghiệm và đánh giá

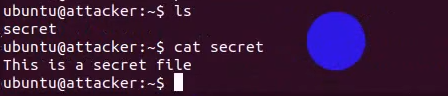
Bài thực hành đã được xây dựng thành công, dưới đây là hình ảnh minh họa về bài thực hành:



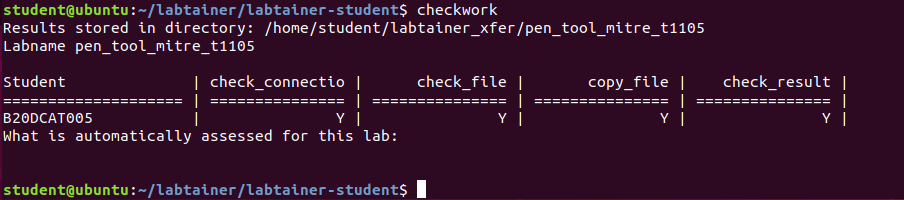
Hình 15. Kết nối đến máy victim và kiểm tra file cần sao chép



Hình 16. Sao chép file từ máy victim về máy attacker



Hình 17. Kiểm tra kết quả sao chép file trên máy attacker



Hình 18. Đánh giá kết quả bài thực hành.