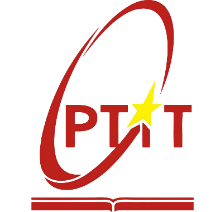
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**

-------🙞🙜🕮🙞🙜-------



**Báo cáo bài thực hành**

**TÌM HIỂU BÀI THỰC HÀNH**

**PTIT – PEN\_ATTK\_MITRE\_T1014**

Giảng viên: Nguyễn Ngọc Điệp

Môn học: Đồ án tốt nghiệp

Sinh viên: Đinh Việt Anh

Mã sinh viên: B20DCAT005

**Hà Nội - 2025**

Mục lục

[Bài thực hành: Tìm hiểu về Rootkit (pen\_attk\_mitre\_t1014) 6](#_Toc187418652)

[1. Giới thiệu chung về bài thực hành: 6](#_Toc187418653)

[2. Nội dung và hướng dẫn bài thực hành 7](#_Toc187418654)

[3. Phân tích yêu cầu bài thực hành 9](#_Toc187418655)

[4. Thiết kế bài thực hành 9](#_Toc187418656)

[5. Cài đặt và cấu hình các máy ảo 11](#_Toc187418657)

[6. Tích hợp và triển khai 12](#_Toc187418658)

[7. Thử nghiệm và đánh giá 17](#_Toc187418659)

DANH MỤC CÁC HÌNH VẼ

[Hình 1. Giao diện Labedit của bài lab 11](#_Toc186664003)

[Hình 2. Cài đặt phần Result 11](#_Toc186664004)

[Hình 3. Dockerfiles của máy victim 12](#_Toc186664005)

[Hình 4. Đăng nhập trong Labtainer 13](#_Toc186664006)

[Hình 5. Khởi tạo quá trình lưu trữ trên dockerhub 13](#_Toc186664007)

[Hình 6. Khai báo thông tin tài khoản 14](#_Toc186664008)

[Hình 7. Thêm bài lab muốn lưu trữ 14](#_Toc186664009)

[Hình 8. Đăng nhập tài khoản dockerhub 14](#_Toc186664010)

[Hình 9. Quá trình tải bài lab lên dockerhub 15](#_Toc186664011)

[Hình 10. Lưu trữ bài lab trên dockerhub 16](#_Toc186664012)

[Hình 11. Lưu trữ bài lab trên github 16](#_Toc186664013)

[Hình 12. Tải bài lab về 17](#_Toc186664014)

[Hình 13. Chạy lab từ imodule 17](#_Toc186664015)

[Hình 13. Sửa đổi code giấu file được yêu cầu 18](#_Toc186664016)

[Hình 14. Tạo file cần giấu theo yêu cầu 18](#_Toc186664017)

[Hình 15. Biên dịch code 19](#_Toc186664018)

[Hình 16. Chèn module vào kernel 19](#_Toc186664019)

[Hình 17. Chèn module thành công 19](#_Toc186664020)

[Hình 18. Kiểm tra hoạt động của rootkit 19](#_Toc186664021)

[Hình 19. Gỡ bỏ rootkit và đánh giá kết quả bài thực hành 20](#_Toc186664022)

DANH MỤC BẢNG BIỂU

[Bảng 1. Bảng Result 9](#_Toc185894910)

Bài thực hành: Tìm hiểu về Rootkit (pen\_attk\_mitre\_t1014)

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

**Bài thực hành "Tìm hiểu về Rootkit"** được thiết kế nhằm giúp sinh viên hiểu rõ cách các kẻ tấn công có thể sử dụng rootkit để ẩn các hoạt động độc hại trên hệ thống Linux. Đây là một khía cạnh quan trọng trong bảo mật hệ thống, giúp sinh viên làm quen với các phương pháp phát hiện và ngăn chặn các hành vi xâm phạm ẩn trong hệ điều hành.

Trong bài thực hành này, sinh viên sẽ làm việc trong một môi trường thực hành chuyên biệt, bao gồm một hệ thống mục tiêu và môi trường kiểm thử được cấu hình sẵn. Thông qua các bước thực hành, sinh viên sẽ học cách tìm kiếm, phân tích và khai thác rootkit, cũng như cách phát triển các mô-đun kernel độc hại cơ bản để hiểu rõ cơ chế hoạt động của chúng.

**Các kỹ năng mà sinh viên sẽ học được bao gồm:**

* Phân tích và nhận diện các hành vi ẩn của rootkit trong hệ thống Linux.
* Hiểu cách rootkit thao túng các thành phần như bảng syscall, các hàm kernel, và cấu trúc hệ điều hành.
* Phát triển các đoạn mã đơn giản để ẩn tệp, tiến trình hoặc các hoạt động khác trên hệ thống Linux.
* Hiểu rõ rủi ro bảo mật khi hệ thống bị nhiễm rootkit và cách các nhà quản trị có thể phòng chống.

**Sau bài thực hành này, sinh viên sẽ nắm vững:**

* Cách hoạt động của rootkit trên hệ thống Linux, đặc biệt trong việc thao túng nhân (kernel).
* Cách phát hiện các dấu hiệu cho thấy hệ thống đã bị nhiễm rootkit.
* Các biện pháp cơ bản để phòng ngừa và phát hiện rootkit trong môi trường thực tế.

Thông qua bài thực hành, sinh viên sẽ nâng cao kỹ năng đánh giá bảo mật hệ thống Linux và có nhận thức rõ ràng hơn về các mối nguy hiểm tiềm ẩn khi hệ thống bị xâm phạm. Đây là bước đệm quan trọng giúp sinh viên phát triển kiến thức chuyên sâu về an toàn thông tin, đặc biệt là trong lĩnh vực phân tích mã độc và các kỹ thuật bảo mật kernel.

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

2.1 Mục đích

Bài thực hành giúp sinh viên tìm hiểu cách các **rootkit** được sử dụng để thao túng và ẩn các hoạt động độc hại trên hệ thống Linux. Sinh viên sẽ xây dựng và thử nghiệm một rootkit đơn giản, từ đó hiểu rõ cơ chế hoạt động của rootkit, bao gồm việc thao tác với nhân (kernel), che giấu tệp, tiến trình và kết nối mạng.

Thông qua bài thực hành, sinh viên sẽ nhận thức được các rủi ro bảo mật nghiêm trọng khi hệ thống bị nhiễm rootkit và hiểu cách các kẻ tấn công tận dụng rootkit để che giấu hành vi của chúng. Đồng thời, bài thực hành cũng cung cấp cái nhìn sâu sắc về cách phát hiện và ngăn chặn rootkit trong thực tế.

* 1. Yêu cầu đối với sinh viên
* Sinh viên cần có kiến thức cơ bản về:
* **Hệ điều hành Linux** và cách quản lý tệp, tiến trình, quyền người dùng.
* **Kernel module**: Hiểu khái niệm mô-đun nhân và cách thêm/gỡ mô-đun trên Linux.
* **Ngôn ngữ lập trình C**: Cơ bản về đọc, viết mã trong C, đặc biệt liên quan đến kernel programming.
* **Lệnh và công cụ bảo mật**: Sử dụng các lệnh như lsmod, insmod, rmmod, và công cụ kiểm tra nhật ký hệ thống như dmesg.
* **Quy trình phát hiện rootkit**: Phân tích dấu hiệu nhiễm mã độc và cách xử lý.
* Bài thực hành cung cấp cơ hội để sinh viên trải nghiệm thực tế cách rootkit thao tác hệ điều hành, đồng thời trang bị kiến thức quan trọng trong việc phòng chống và phát hiện mã độc trong môi trường an toàn thông tin.
  1. Nội dung thực hành

Khởi động bài lab:

Vào terminal, gõ:

*Labtainer -r pen\_attk\_mitre\_t1014*

*(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 terminal ảo sẽ xuất hiện đại diện cho máy ***victim***.

Sinh viên cần thay thay đổi đoạn code để rootkit có thể ẩn đúng file được yêu cầu.

Sau khi thay đổi xong, lưu file và thực hiện biên dịch

*Make*

Sau khi biên dịch thành công, 1 file **.*ko*** sẽ xuất hiện.

Sinh viên tạo 1 file được yêu cầu ẩn đi với nội dung bất kỳ

*nano file\_name*

Thực hiện chèn module vào kernel

*Sudo insmod lkmdemo.ko*

Nếu đã chèn thành công, kết quả sẽ hiển thị module lkmdemo đang chạy

Thực hiện kiểm tra đã giấu file được yêu cầu thành công hay chưa, module đã được chèn vào kernel thành công hay chưa

*ls && cat nano file\_name && lsmod | grep lkmdemo > output*

Kết quả lệnh list file không hiển thị file được yêu cầu giấu nhưng vẫn có thể đọc được file đó, rootkit đã hoạt động.

Xóa module khỏi kernel

sudo rmmod lkmdemo.ko

Kết thúc bài lab:

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

*stoplab pen\_attk\_mitre\_t1014*

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 pen\_attk\_mitre\_t1014*

1. Phân tích yêu cầu bài thực hành

Bài thực hành được thiết kế chỉ 1 máy ảo mô phỏng ***victim*** được build dựa trên package .kali tích hợp công cụ kmod, code rootkit che giấu file và các thư viện cần thiết để biên dịch module và chèn module vào kernel. Để hoàn thành bài thực hành sinh viên cần sử dụng code rootkit có sẵn, thay đổi code để giấu file được yêu cầu, biên dịch code và chèn module vào kernel, sau đó thực hiện kiểm tra file đã được che giấu chưa, module đã được chèn vào kernel thành công, hệ thống hoạt động bình thường hay không. Cuối cùng xóa bỏ rootkit khỏi kernel

1. Thiết kế bài thực hành

Trên môi trường máy ảo Ubuntu được cung cấp, sử dụng docker tạo ra 1 container: mang tên “user” đóng vai trò máy nạn nhân.

Cấu hình docker gồm có:

user: lưu cấu hình cho máy nạn nhân, trong đó gồm có:

Tên máy: user

config: lưu cấu hình hoạt động của hệ thống

dockerfiles: mô tả cấu hình của container: user, trong đó:

user: sử dụng các thư viện mặc định hệ thống và package .kali tích hợp công cụ kmod, code rootkit che giấu file và các thư viện cần thiết để biên dịch module và chèn module vào kernel.

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:

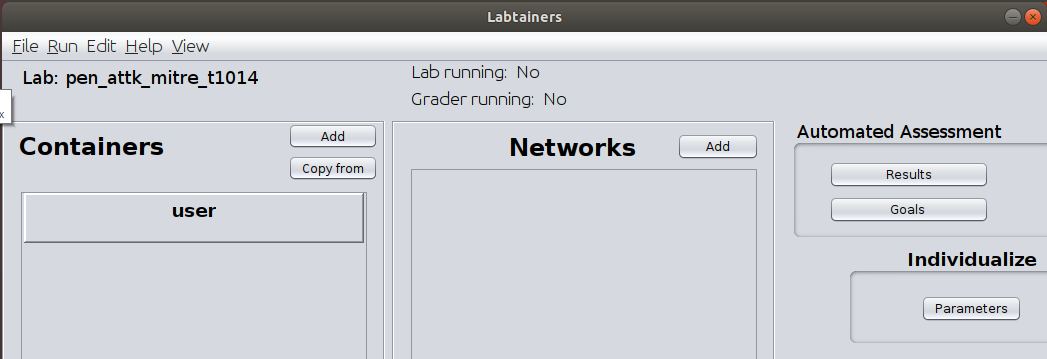
* Tạo 1 file được yêu cầu
* Chỉnh sửa file code để giấu file đó
* Biên dịch code sau khi đã chỉnh sửa
* Sau khi biên dịch thành công, chèn module vào kernel
* Kiểm tra module đã được chèn thành công hay chưa.
* Kiểm tra rootkit có hoạt động che giấu đúng file hay không
* Kết thúc bài lab và đóng gói 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:

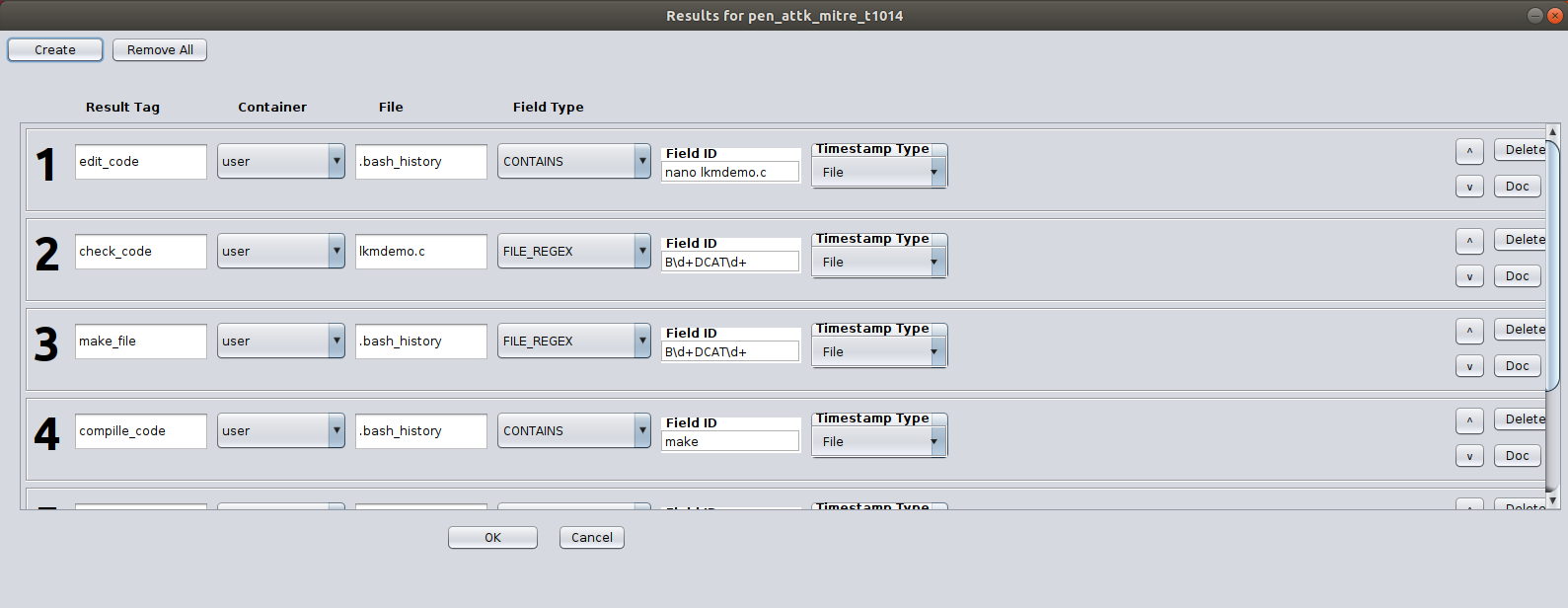
Bảng 1. Bảng Result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Result Tag | Container | File | Field Type | Field ID | Timestamp Type |
| edit\_code | user | .bash\_history | CONTAINS | nano lkmdemo.c | File |
| check\_code | user | lkmdemo.c | FILE\_REGEX | B\d+DCAT\d+ | File |
| make\_file | user | .bash\_history | FILE\_REGEX | B\d+DCAT\d+ | File |
| compille\_code | user | .bash\_history | CONTAINS | make | File |
| load\_the\_kernel\_module | user | .bash\_history | CONTAINS | sudo insmod lkmdemo.ko | File |
| check\_module | user | output | FILE\_REGEX | lkmdemo\s+\d+ | File |

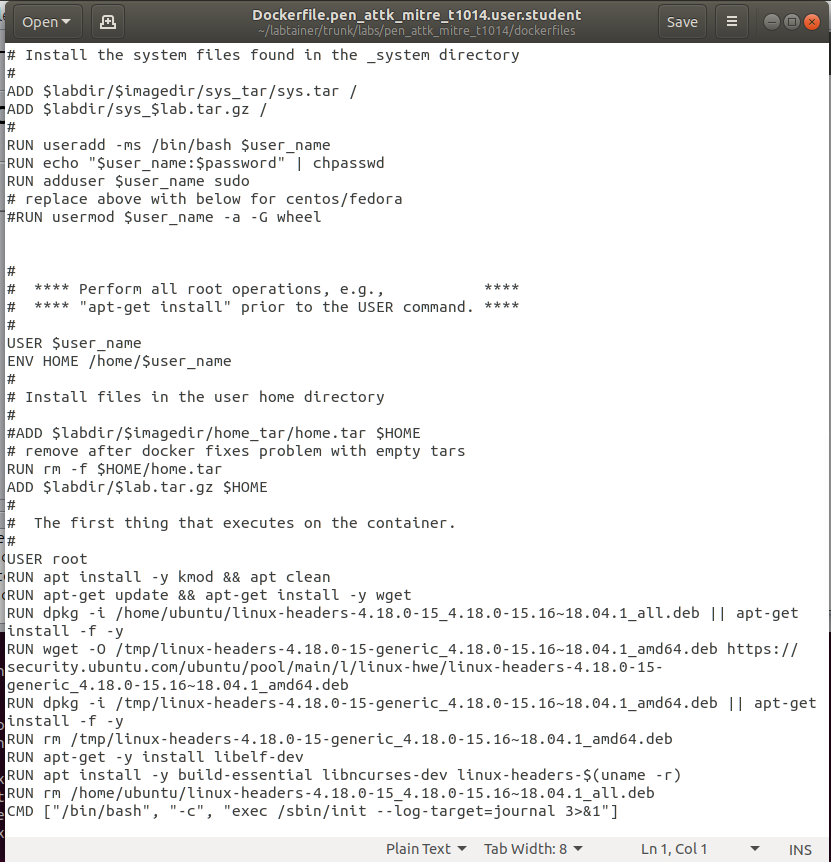
1. Cài đặt và cấu hình các máy ảo



Hình 1. Giao diện Labedit của bài lab



Hình 2. Cài đặt phần Result



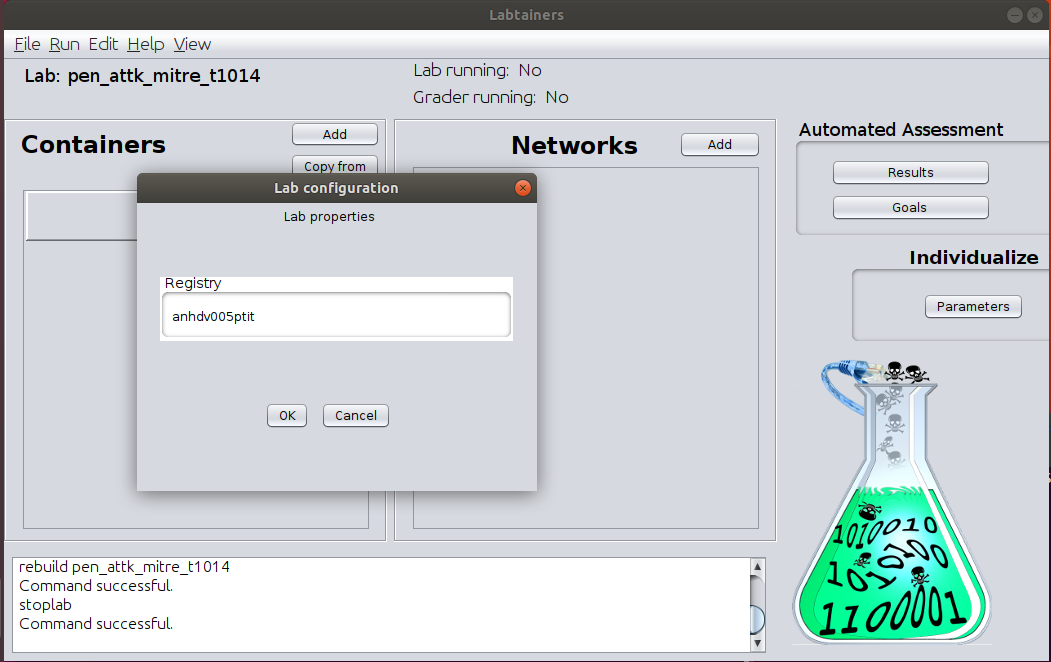
Hình 3. Dockerfiles của máy victim

1. Tích hợp và triển khai

Bài thực hành đã được triển khai như sau:

* + - * 1. Docker Hub

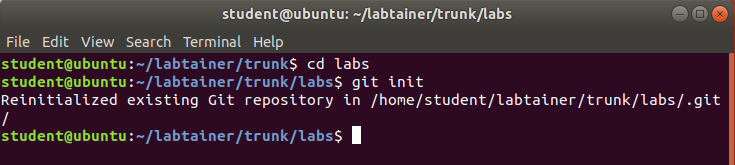
Tạo tài khoản trên trang https://hub.docker.com trên giao diện labtainer chọn mục Edit/Config (registry) đănh nhập tên tài khoản hiển thị trên dockerhub và Build only lại.



Hình 4. 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 5. Khởi tạo quá trình lưu trữ trên dockerhub

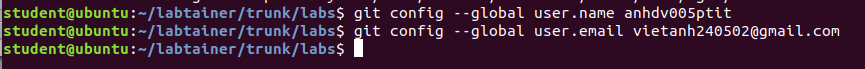
Khai báo các thông số để kết nối đến dockerhub với lệnh:

***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)

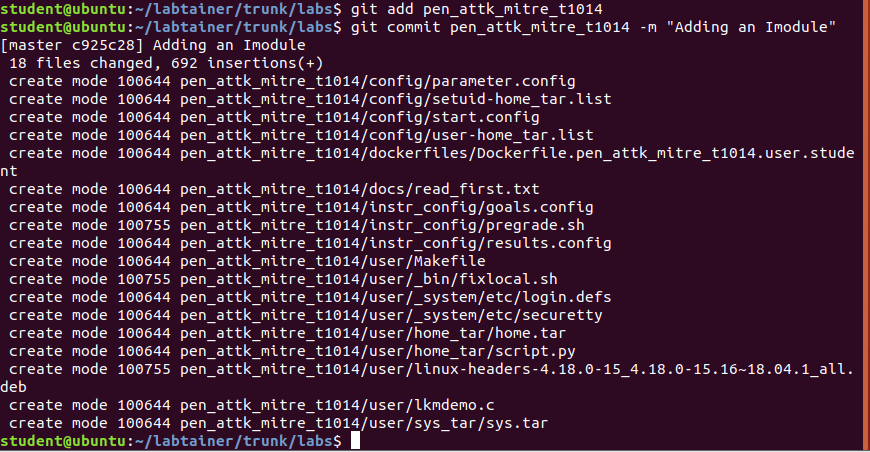


Hình 6. Khai báo thông tin tài khoản

Thêm bài lab muốn lưu trữ vào git theo lệnh:

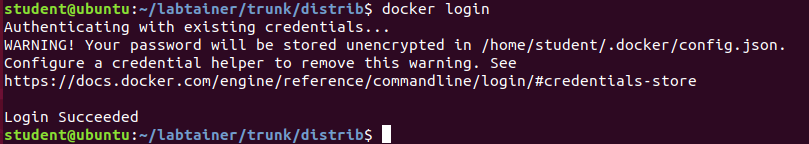
**git add <tên bài lab>**

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

****

Hình 7. Thêm bài lab muốn lưu trữ

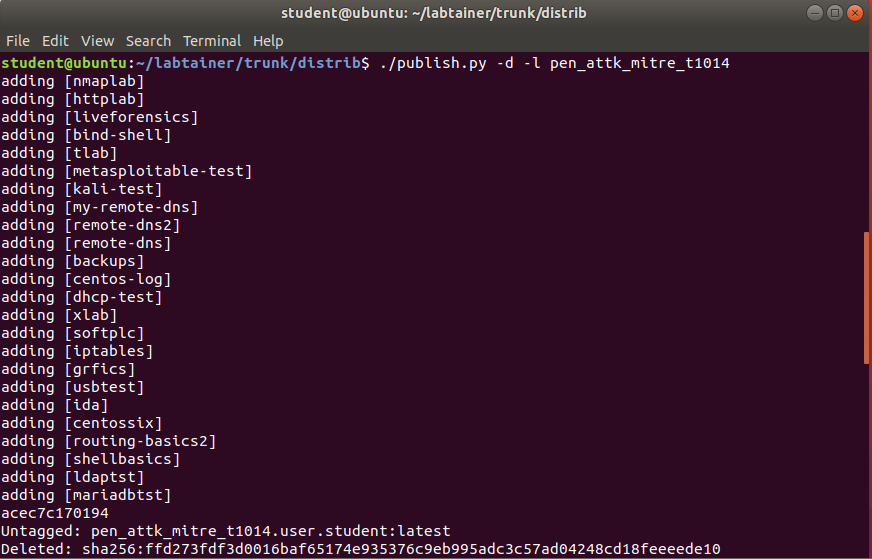
Chuyển đến thư mục ~trunk/distrib, sử dụng lệnh docker login để đăng nhập docker trên Labtainer



Hình 8. Đăng nhập tài khoản dockerhub

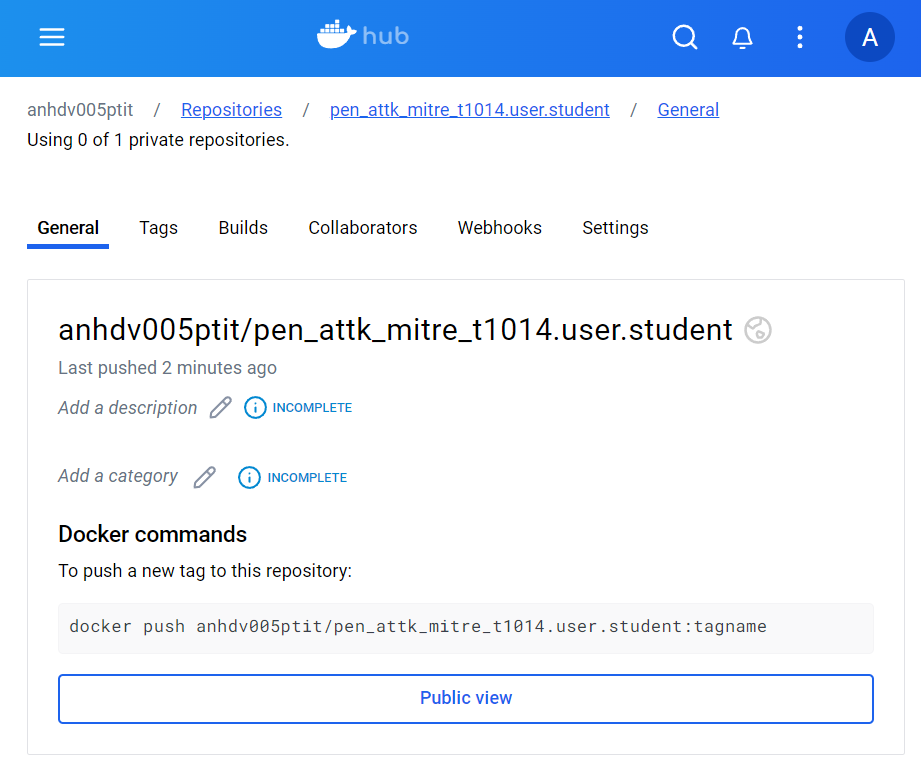
Đẩy bài lab lên dockerhub bằng cách sử dụng lệnh:

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

******

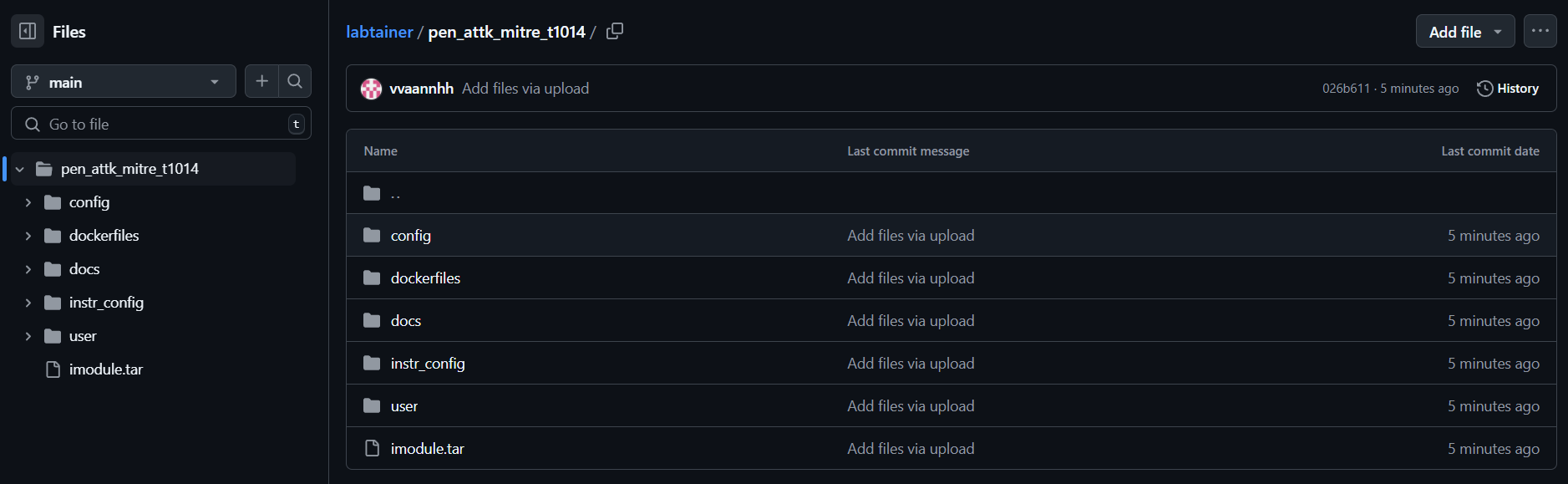
Hình 9. Quá trình tải bài lab lên dockerhub

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



Hình 10. Lưu trữ bài lab trên dockerhub

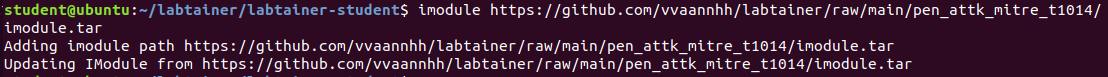
* + - * 1. Github



Hình 11. Lưu trữ bài lab trên github

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

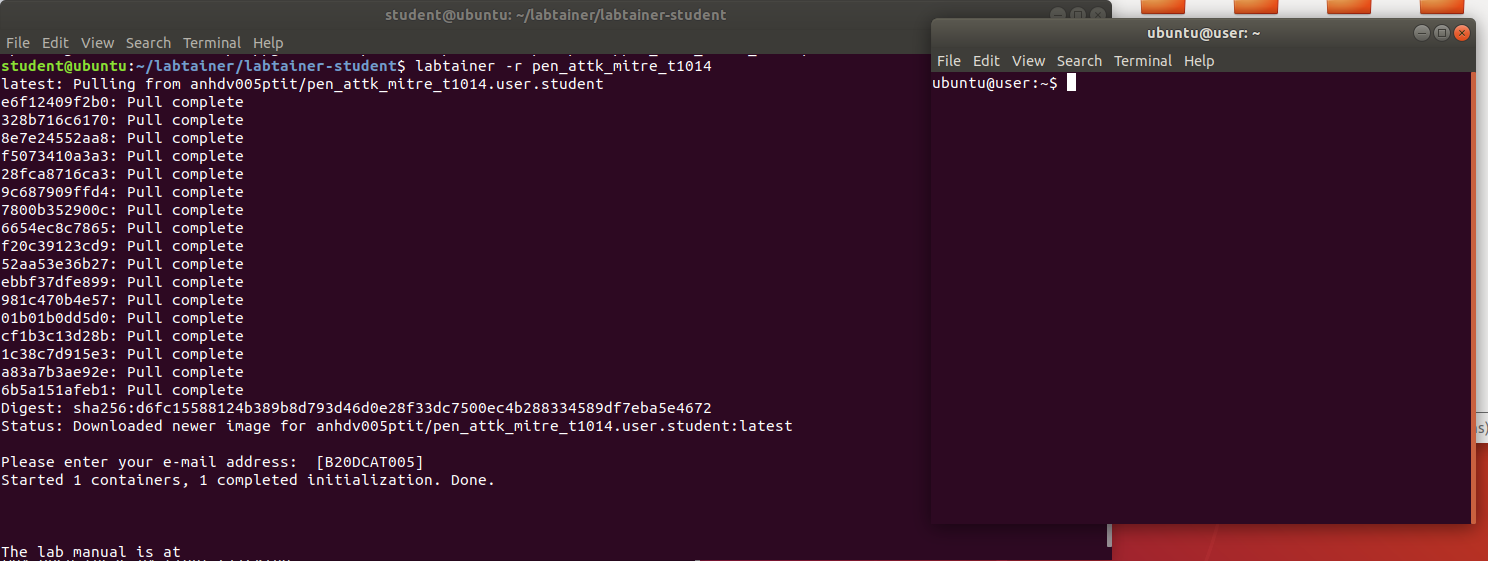
***imodule <đường dẫn>***



Hình 12. Tải bài lab về

Load bài lab từ imodule

***labtainer -r <tên bài lab>***



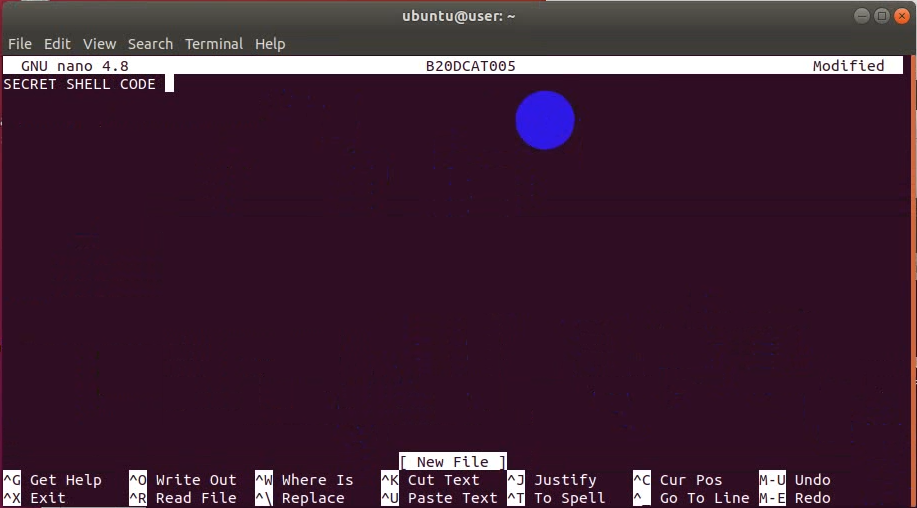
Hình 13. Chạy lab từ imodule

1. Thử nghiệm và đánh giá

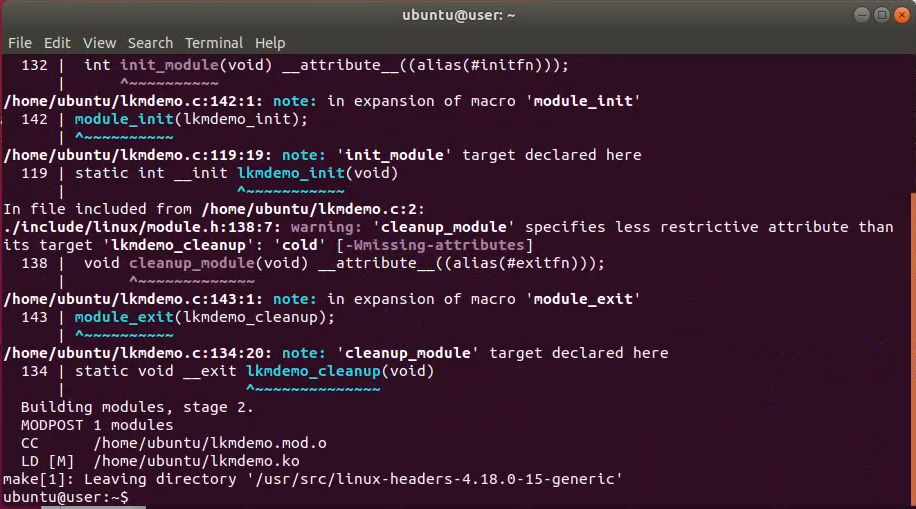
Bài thực hành đã được xây dựng thành công, dưới đây là hình ảnh minh họa về bài thực hành:



Hình 13. Sửa đổi code giấu file được yêu cầu



Hình 14. Tạo file cần giấu theo yêu cầu



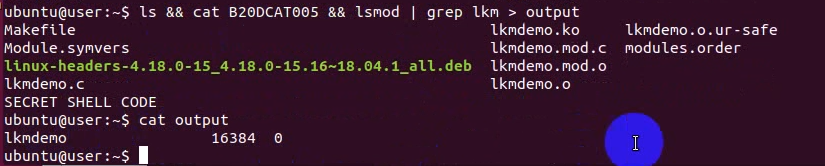
Hình 15. Biên dịch code



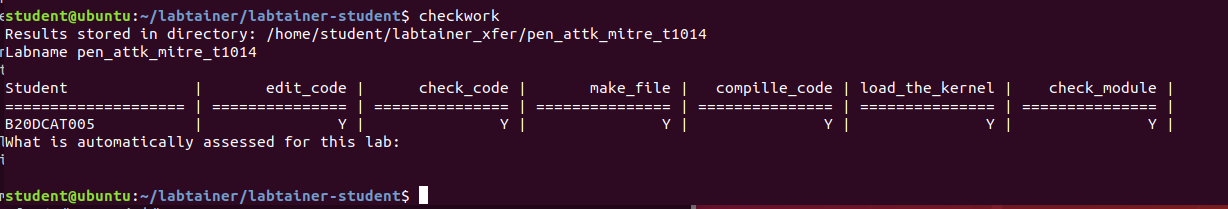
Hình 16. Chèn module vào kernel



Hình 17. Chèn module thành công



Hình 18. Kiểm tra hoạt động của rootkit



Hình 19. Gỡ bỏ rootkit và đánh giá kết quả bài thực hành