

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



BÁO CÁO LAB CUỐI KỲ

**ĐỀ TÀI ỨNG DỤNG MOD SECURITY ĐỂ CHẶN CÁC TẤN
CÔNG VÀO WEB (DÙNG DVWA)**

Môn: AN TOÀN MẠNG

Giảng viên hướng dẫn: ThS. Trần Thị Dung

Nhóm 03

Sinh viên thực hiện:	Lương Minh Tiến	–	N18DCAT069
	Trần Văn Tư	–	N18DCAT081
	Phạm Thạch	–	N18DCAT082

TPHCM - Tháng 10, 2021

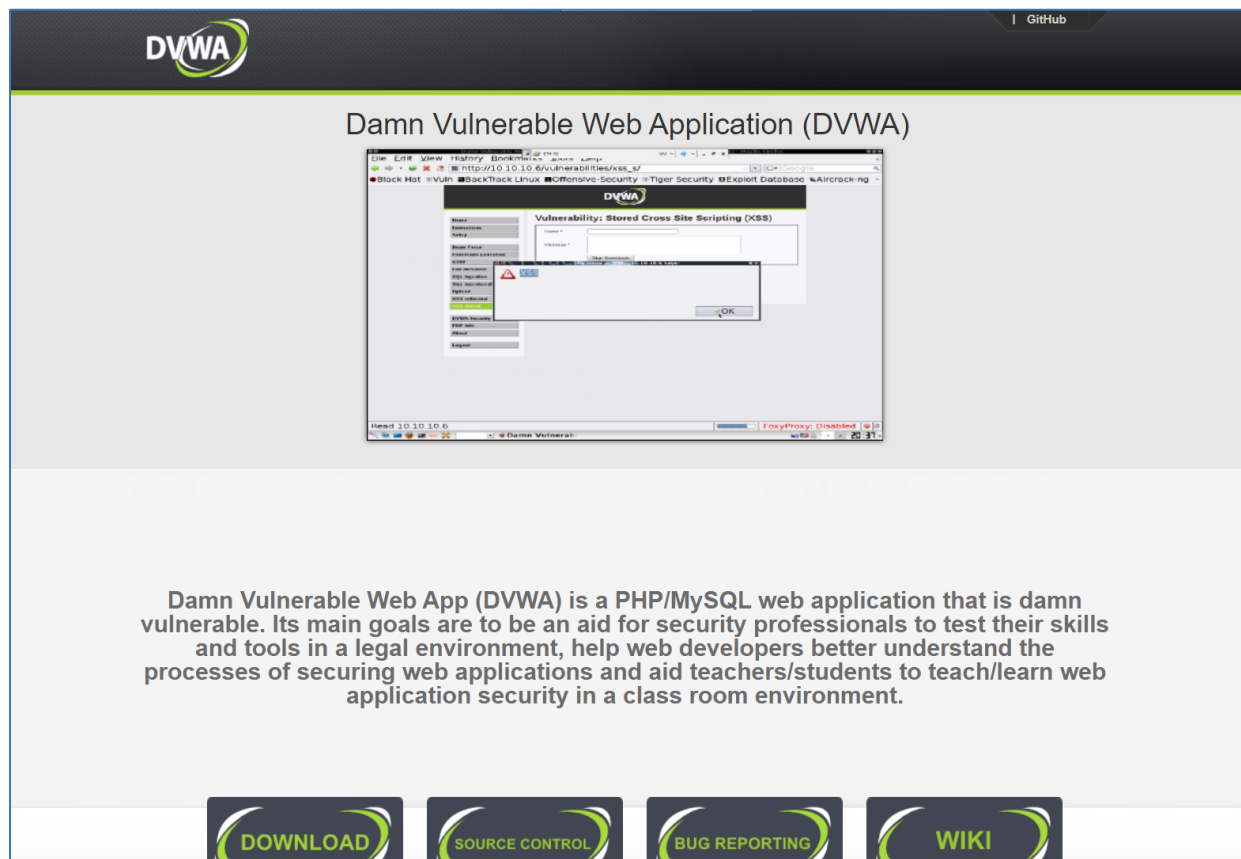
Mục lục

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	3
1.1. DVWA	3
1.2. ModSecurity	3
CHƯƠNG 2. YÊU CẦU CƠ BẢN	4
2.1. Cài đặt và cấu hình DVWA	4
2.2 Cách sử dụng DVWA	7
2.3 Phân tích cách phòng thủ các tấn công XSS trong DVWA	10
2.3.1 Reflected XSS	10
2.3.2 Stored XSS	10
2.3.3 DOM XSS	12
2.4 Cách cài đặt và cấu hình ModSecurity	13
2.5 Tự viết một bộ rule cho ModSecurity	19
2.5.1 Cấu trúc của một rule	19
2.5.2 Một số ví dụ	21
2.5.3 Chặn các tấn công XSS ở mức low	23
CHƯƠNG 3. YÊU CẦU NÂNG CAO	23
3.1 Mô hình	23
3.2 Cài đặt và cấu hình DVWA	23
3.3 Cài đặt Apache2 và cấu hình ReverseProxy	26
3.4 Cài đặt và cấu hình ModSecurity	28
3.5 Kiểm thử ModSecurity (nhóm em sẽ test trên web server 2)	30
CHƯƠNG 4. KẾT LUẬN	35
CHƯƠNG 5. BẢNG PHÂN CHIA CÔNG VIỆC	35

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. DVWA

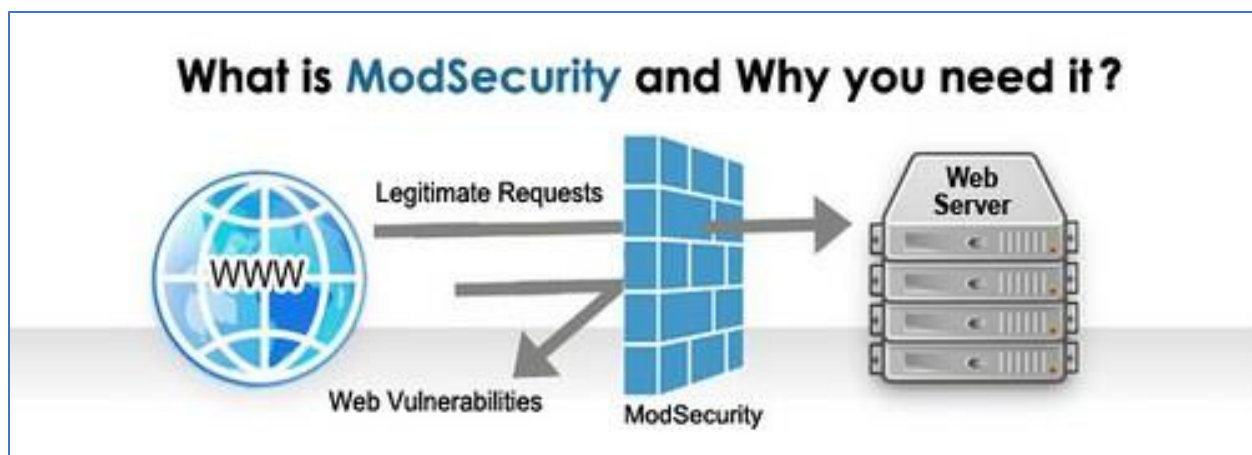
DVWA là 1 ứng dụng web (web application) miễn phí tích hợp PHP/MySQL dễ bị tấn công và nó là một môi trường thích hợp để người dùng sử dụng bằng cách lên trên này và thực hiện các tấn công để rèn luyện kỹ năng bảo mật hoặc thử nghiệm công cụ nào đó. DVWA giúp các nhà lập trình web hiểu rõ hơn về cách thức bảo mật web và nó cũng nhắm đến đối tượng là học sinh, sinh viên và giảng viên trong việc học tập về bảo mật ứng dụng web trong một môi trường học tập có thể kiểm soát được (nói dễ hiểu là cung cấp môi trường để cho học sinh, sinh viên khỏi đi phá lung tung trên website của người khác). DVWA cung cấp 10 loại tấn công web phổ biến nhất hiện nay dựa theo OWASP. Ở đây nhóm sử dụng dvwa phiên bản 1.10



Hình 1.1.1 Trang chủ của DVWA

1.2. ModSecurity

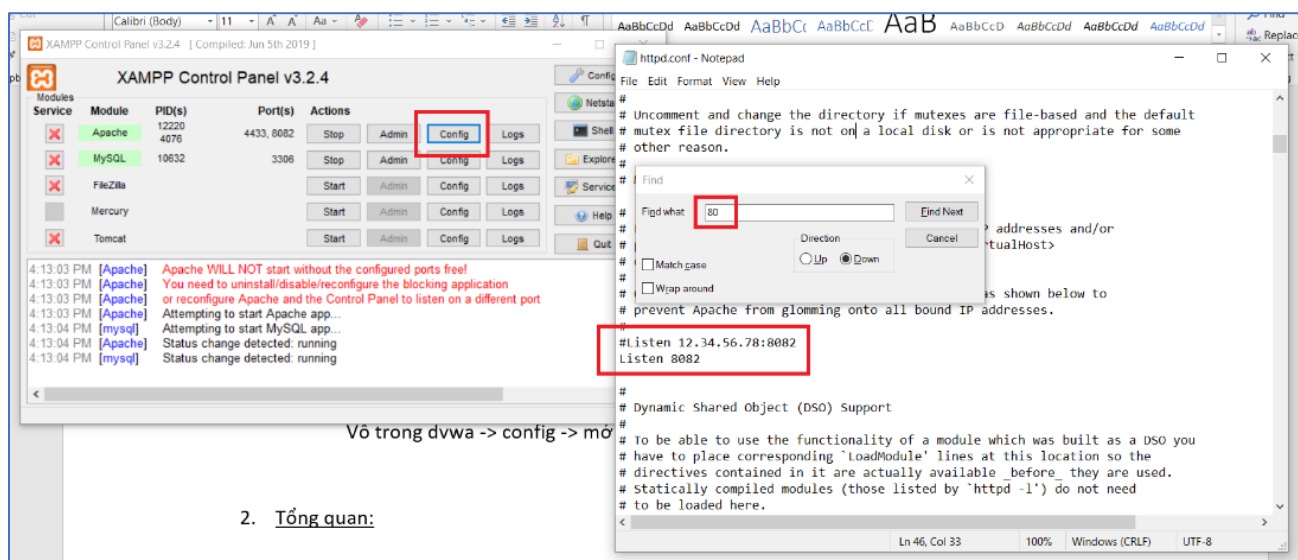
Mod security là một tường lửa open-source được tích hợp vào các web server như apache, IIS, Nginx ... nhằm bảo vệ các ứng dụng web khỏi các cuộc tấn công web phổ biến. Nó đóng vai trò như một tường lửa để ngăn chặn các dữ liệu độc hại đi vào web server. Ở đây nhóm em dùng bản 2.9.3 win 64.



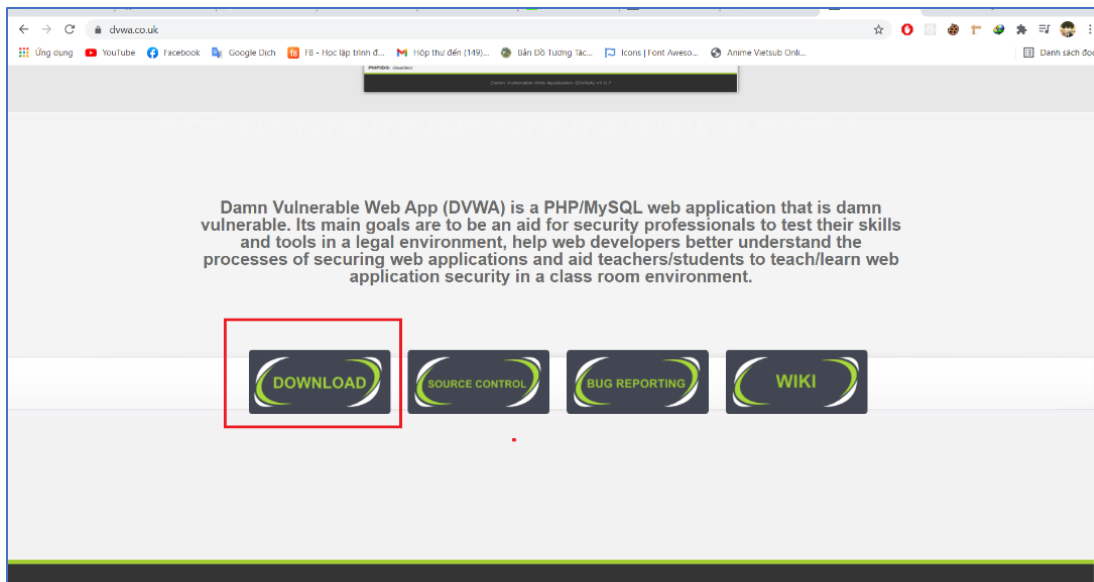
Hình 1.2.1 Cách thức hoạt động của ModSecurity

CHƯƠNG 2. YÊU CẦU CƠ BẢN

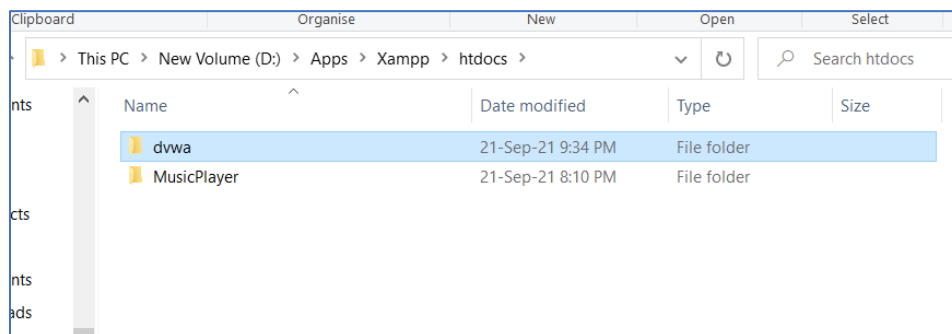
2.1. Cài đặt và cấu hình DVWA



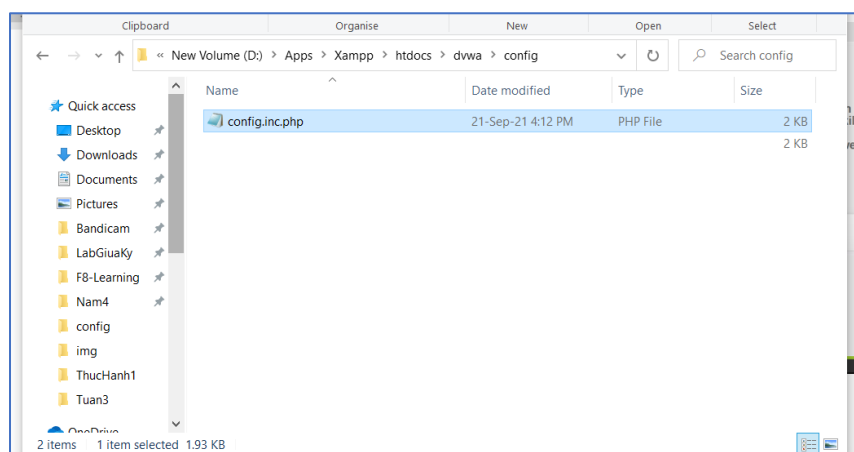
Hình 2.1.1 Đầu tiên cần chỉnh port cho Xampp để khởi động được các dịch vụ khác



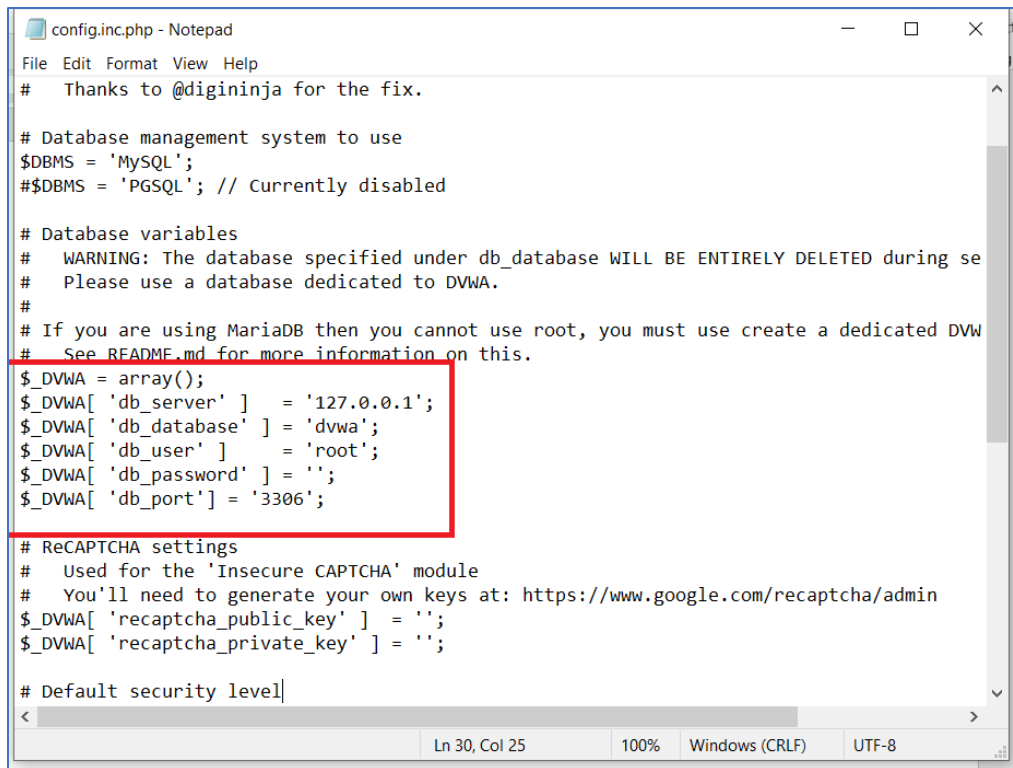
Hình 2.1.2 Tải DVWA từ trang chủ



Hình 2.1.3 Giải nén và copy thư mục dvwa vừa tải vào thư mục htdocs của Xampp



Hình 2.1.4 Vào thư mục dvwa -> config -> mở file config.inc.php (nếu có đuôi .dist thì xóa đi)



```
config.inc.php - Notepad
File Edit Format View Help
# Thanks to @digininja for the fix.

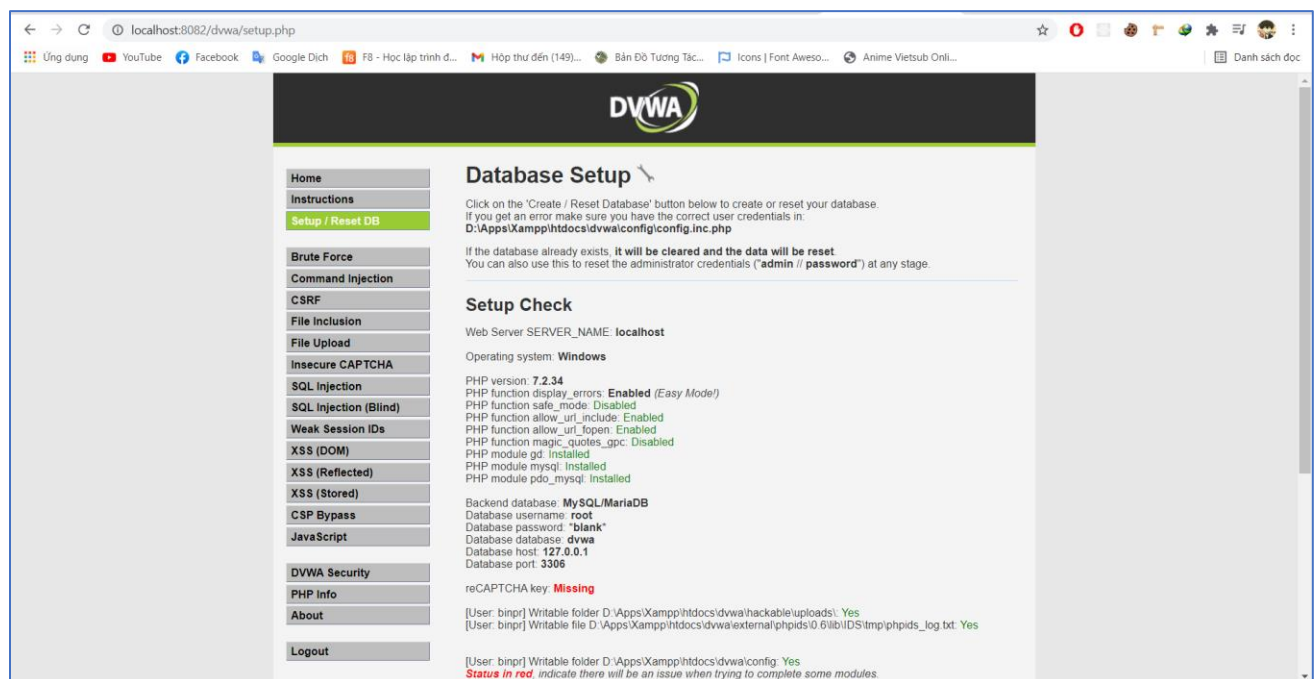
# Database management system to use
$DBMS = 'MySQL';
# $DBMS = 'PGSQL'; // Currently disabled

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during se
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVW
# See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'root';
$_DVWA[ 'db_password' ] = '';
$_DVWA[ 'db_port' ] = '3306';

# ReCAPTCHA settings
# Used for the 'Insecure CAPTCHA' module
# You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ] = '';
$_DVWA[ 'recaptcha_private_key' ] = '';

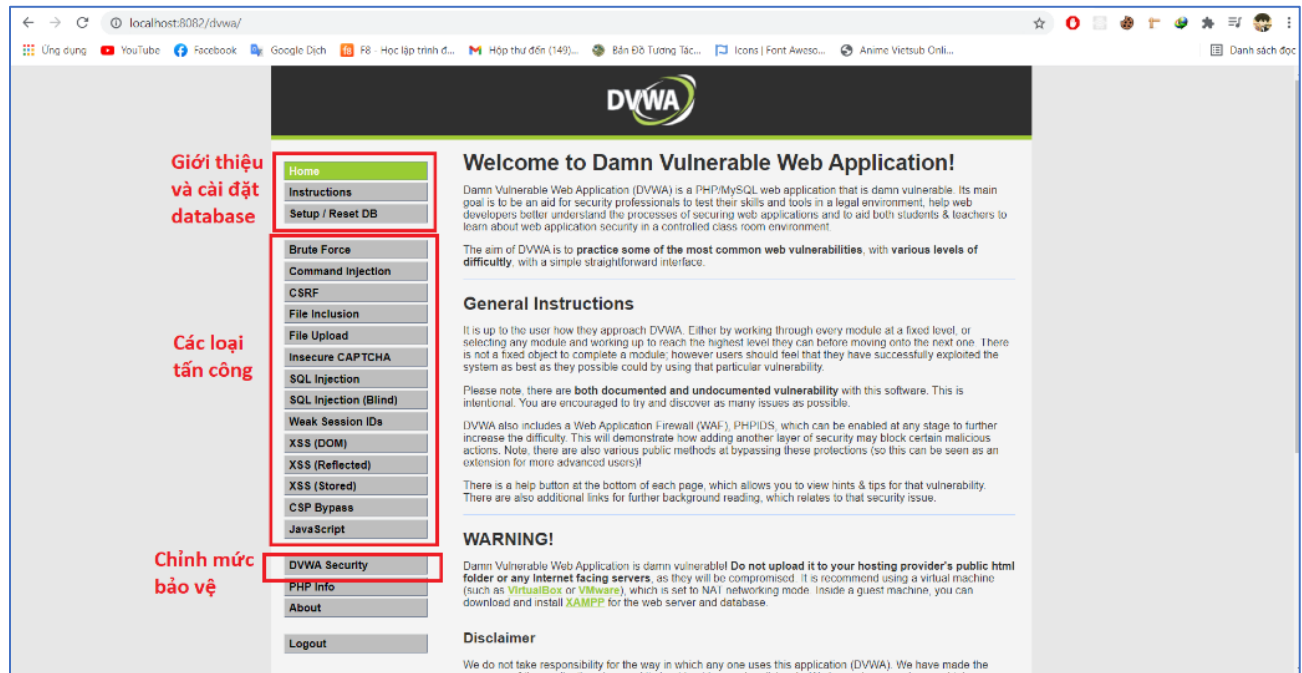
# Default security level|
<
Ln 30, Col 25    100%    Windows (CRLF)    UTF-8
```

Hình 2.1.5 Đổi tên user thành 'root' và để password trống rồi lưu lại

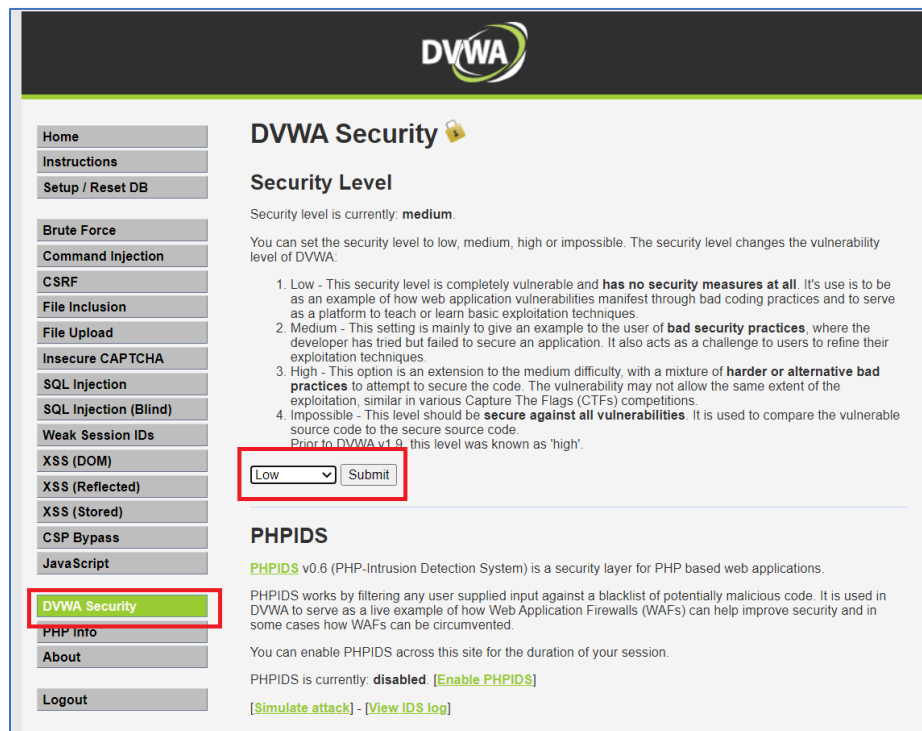


Hình 2.1.6 Kiểm tra bằng cách mở trình duyệt và truy cập đường dẫn localhost:8082/dvwa/setup.php

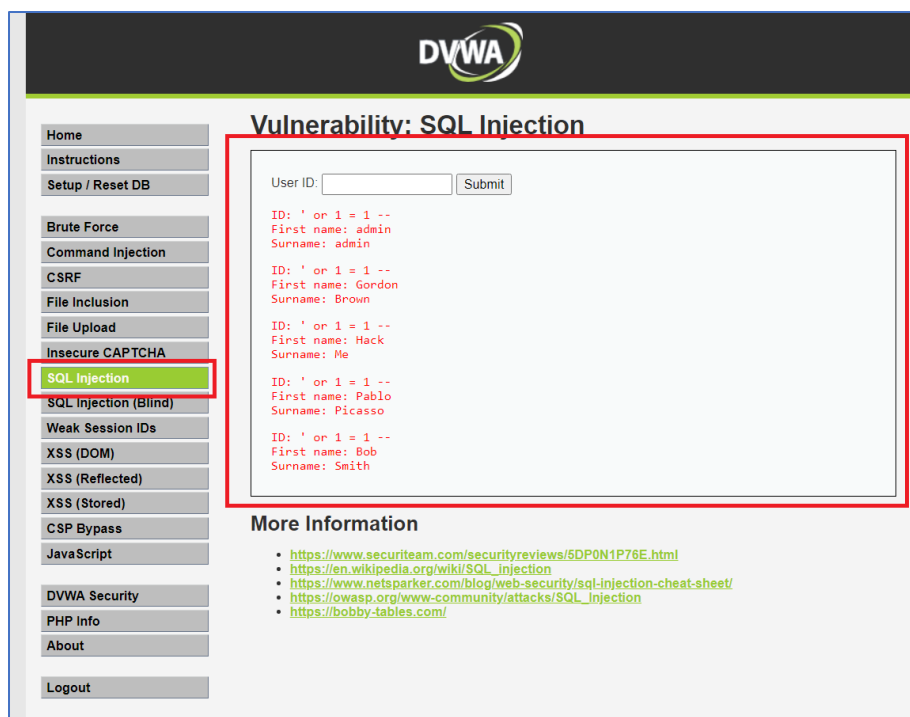
2.2 Cách sử dụng DVWA



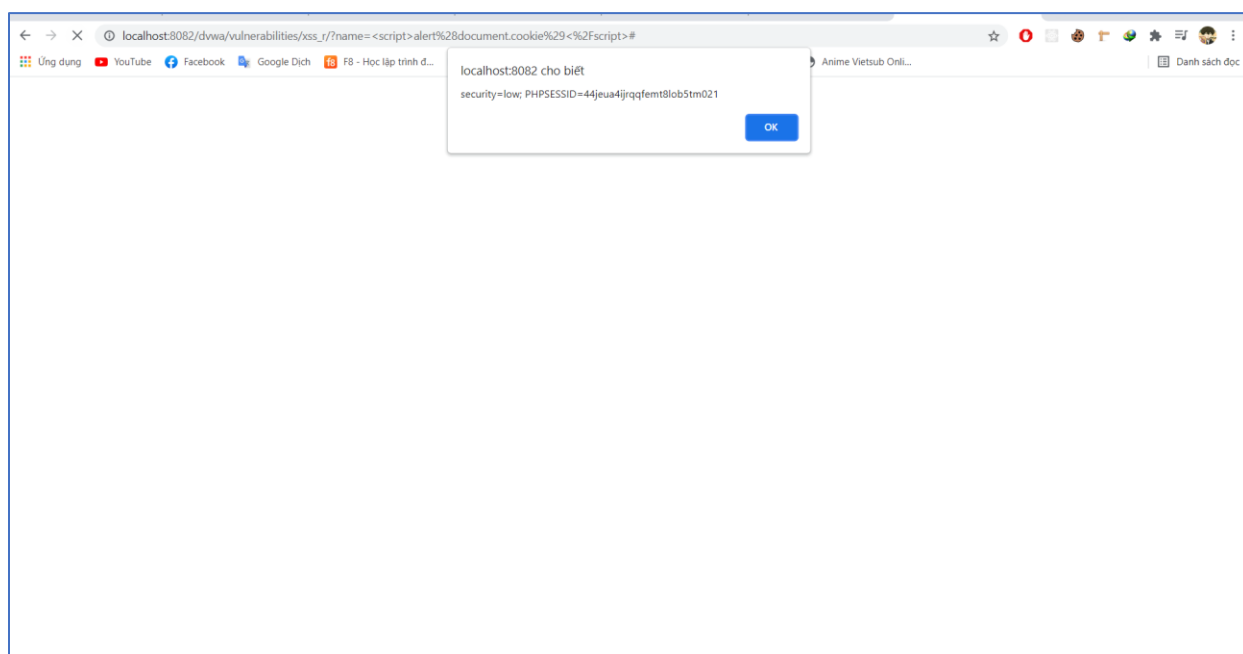
Hình 2.2.1 Giao diện chính của DVWA



Hình 2.2.2 Mặc định DVWA sẽ có mức bảo mật là Impossible nên tại em sẽ đưa về mức low bằng cách vào DVWA Security -> Low -> Submit



Hình 2.2.3 Thử tấn công SQL Injection, nhập '**or 1 = 1**' – (kết quả là select toàn bộ user)



Hình 2.2.4 Thử tấn công XSS (Reflected), nhập **<script>alert(document.cookie)</script>** (kết quả là xuất ra cookie của người dùng)

Vulnerability: Stored Cross Site Scripting (XSS)

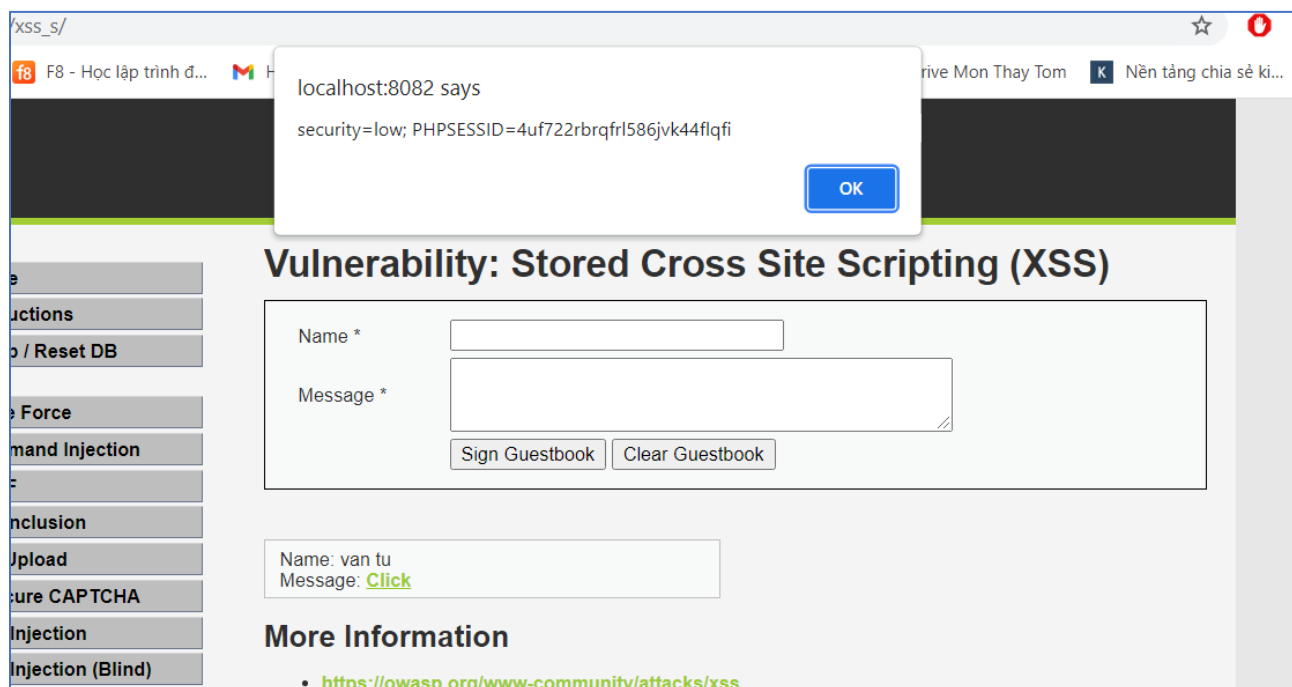
Name *

Message *

More Information

- <https://owasp.org/www-community/attacks/xss>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- https://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

Hình 2.2.5 Thử tấn công Stored XSS bằng cách chèn đoạn script vào Message nhằm tiêm 1 đường link độc hại vào database



Hình 2.2.6 Kết quả là một đường link đã được xuất hiện, nếu người dùng khác bấm vào thì đoạn lệnh hiển thị cookie sẽ được thực thi

2.3 Phân tích cách phòng thủ các tấn công XSS trong DVWA

2.3.1 Reflected XSS

Mức low

```
<?php
header ("X-XSS-Protection: 0");
// Kiểm tra input có rỗng hay không, nếu khác rỗng thì xuất ra màn hình input vừa nhập
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
    echo "<pre>Hello " . $_GET[ 'name' ] . "</pre>";
}
?>
```

Mức medium

```
<?php
header ("X-XSS-Protection: 0");
// Kiểm tra input có rỗng hay không, nếu khác rỗng thì xóa những chuỗi '<script>' trong input và xuất ra màn hình
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
    $name = str_replace( '<script>', '', $_GET[ 'name' ] );
    echo "<pre>Hello ${name}</pre>";
}
?>
```

Mức high

```
<?php
header ("X-XSS-Protection: 0");
// Kiểm tra input có rỗng hay không, nếu khác rỗng thì xóa các ký tự sao cho khi mà bỏ các ký tự ở giữa nó ra thì được chuỗi '<script>', ví dụ <sabcr23ipt>, sau đó xuất ra màn hình
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
    // Get input
    $name = preg_replace( '/<(.*?)s(.*?)c(.*?)r(.*?)i(.*?)p(.*?)t/i', '', $_GET[ 'name' ] );
    // Feedback for end user
    echo "<pre>Hello ${name}</pre>";
}
?>
```

Mức impossible

```
<?php
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
    // Hàm chống CSRF attack
    checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session token' ], 'index.php' );
    // Hàm htmlspecialchars sẽ chuyển đổi các ký tự đặc biệt thành mã html
    $name = htmlspecialchars( $_GET[ 'name' ] );
    echo "<pre>Hello ${name}</pre>";
}
generateSessionToken();
?>
```

2.3.2 Stored XSS

Mức low

```
<?php
if( isset( $_POST[ 'btnSign' ] ) ) {
    // Nhận input và bỏ khoảng trống ở hai đầu chuỗi
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name = trim( $_POST[ 'txtName' ] );
    // Loại bỏ các dấu xet ở trong biến message và lọc các ký tự không hợp lệ để mysql có thể thực thi query được bằng hàm mysqli_real_escape_string
    $message = stripslashes( $message );
    $message = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $message ) : ((trigger_error("[MySQLConverterToo] Fix the mysqli_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    // Chỉ lọc các ký tự không hợp lệ trong biến name để mysql có thể thực thi query được bằng hàm mysqli_real_escape_string
}
```

```

$name = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $name) : ((trigger_error("[MySQLConverterToo] Fix the mysqli_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));

// Thêm vào database
$query = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' );";
$result = mysqli_query($GLOBALS["__mysqli_ston"], $query) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])) ? mysqli_error($GLOBALS["__mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );
}
?>

```

Mức medium

```

<?php
if( isset( $_POST[ 'btnSign' ] ) ) {
    // Nhận input
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name = trim( $_POST[ 'txtName' ] );

    // Đối với biến message, thêm dấu xet trước những dấu nhảy nhằm mục đích chống việc đóng
    chuỗi sớm, sau đó loại bỏ các thẻ html khỏi chuỗi rồi lọc tương tự mức low
    $message = strip_tags( addslashes( $message ) );
    $message = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $message) : ((trigger_error("[MySQLConverterToo] Fix the mysqli_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    $message = htmlspecialchars( $message );

    // Đối với biến name, chỉ loại bỏ các chuỗi '<script>' và lọc tương tự mức low
    $name = str_replace( '<script>', '', $name );
    $name = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $name) : ((trigger_error("[MySQLConverterToo] Fix the mysqli_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));

    // Thêm vào database
    $query = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' );";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])) ? mysqli_error($GLOBALS["__mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );
}
?>

```

Mức high

```

<?php
if( isset( $_POST[ 'btnSign' ] ) ) {
    // Nhận input
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name = trim( $_POST[ 'txtName' ] );

    // Lọc message không khác gì mức medium
    $message = strip_tags( addslashes( $message ) );
    $message = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $message) : ((trigger_error("[MySQLConverterToo] Fix the mysqli_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    $message = htmlspecialchars( $message );

    // Lọc biến name bằng cách bỏ các ký tự sao cho khi mã bỏ các ký tự ở giữa nó ra thì được chuỗi '<script>' rồi lọc tương tự mức medium
    $name = preg_replace( '/<(.*?)s(.*?)c(.*?)r(.*?)i(.*?)p(.*?)t/i', '', $name );
    $name = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $name) : ((trigger_error("[MySQLConverterToo] Fix the mysqli_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));

    // Thêm vào database
    $query = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' );";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])) ? mysqli_error($GLOBALS["__mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );
}
?>

```

Mức impossible

```
<?php
if( isset( $_POST[ 'btnSign' ] ) ) {
    // Hàm chống CSRF attack
    checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name = trim( $_POST[ 'txtName' ] );
    // Lọc biến message tương tự như mức high
    $message = stripslashes( $message );
    $message = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $message) : ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    $message = htmlspecialchars( $message );
    // Lọc biến name tương tự như biến message
    $name = stripslashes( $name );
    $name = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $name) : ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
    $name = htmlspecialchars( $name );
    // Thêm vào database
    $data = $db->
    >prepare( 'INSERT INTO guestbook ( comment, name ) VALUES ( :message, :name );' );
    $data->bindParam( ':message', $message, PDO::PARAM_STR );
    $data->bindParam( ':name', $name, PDO::PARAM_STR );
    $data->execute();
}
generateSessionToken();
?>
```

2.3.3 DOM XSS

Mức low không có phòng thủ

Mức medium

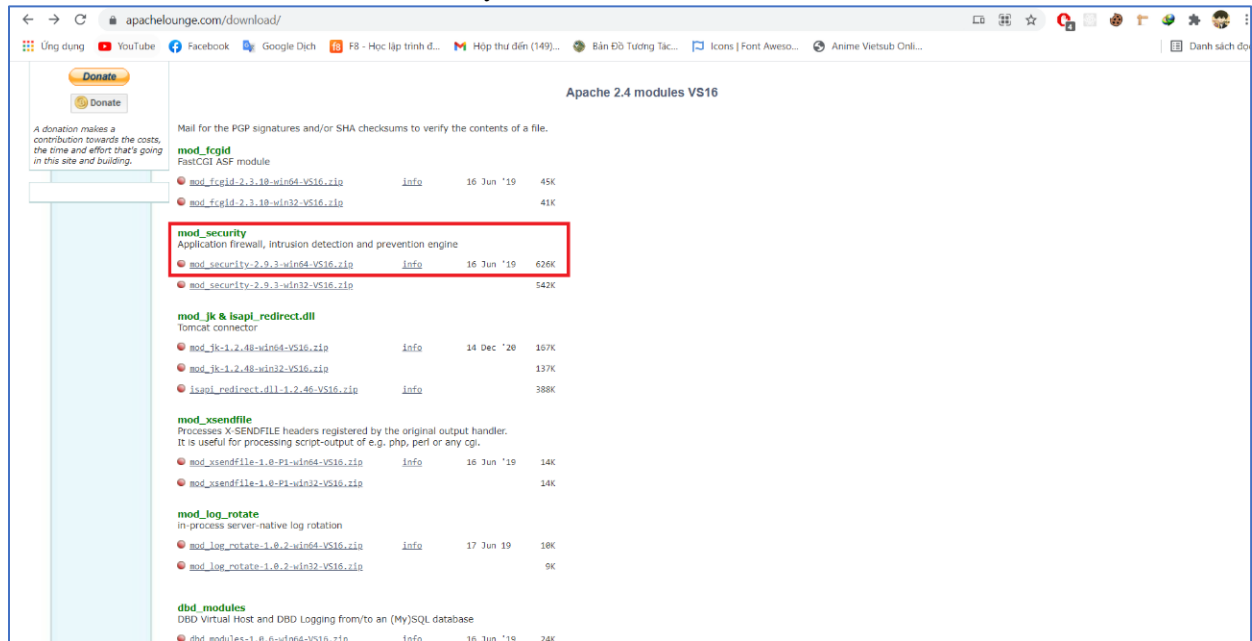
```
<?php
// Kiểm tra xem input có rỗng không, nếu không thì kiểm tra tiếp biến default có chứa chuỗi '<script' không, nếu có thì cho set giá trị của biến default bằng 'English'
if ( array_key_exists( "default", $_GET ) && !is_null ( $_GET[ 'default' ] ) ) {
    $default = $_GET[ 'default' ];
    if ( strpos ( $default, "<script" ) !== false ) {
        header ( "location: ?default=English" );
        exit;
    }
}
?>
```

Mức high

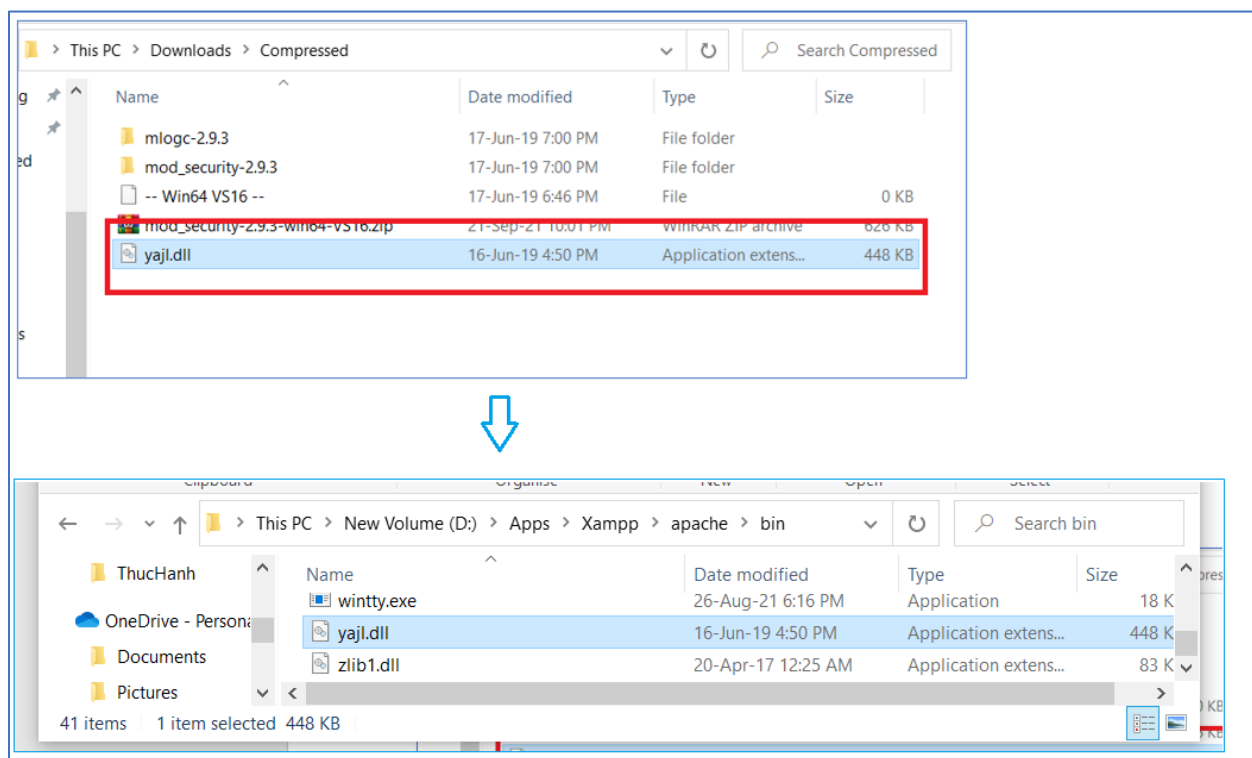
```
<?php
// Kiểm tra xem input có rỗng không, nếu không thì kiểm tra giá trị của biến default có nằm trong các từ được cho phép không (French, German,...), nếu không thì set giá trị của nó là 'English'
if ( array_key_exists( "default", $_GET ) && !is_null ( $_GET[ 'default' ] ) ) {
    # White list the allowable languages
    switch ( $_GET[ 'default' ] ) {
        case "French":
        case "English":
        case "German":
        case "Spanish":
            # ok
            break;
        default:
            header ( "location: ?default=English" );
            exit;
    }
}
?>
```

Mức impossible chỉ có thể chặn được ở phía client

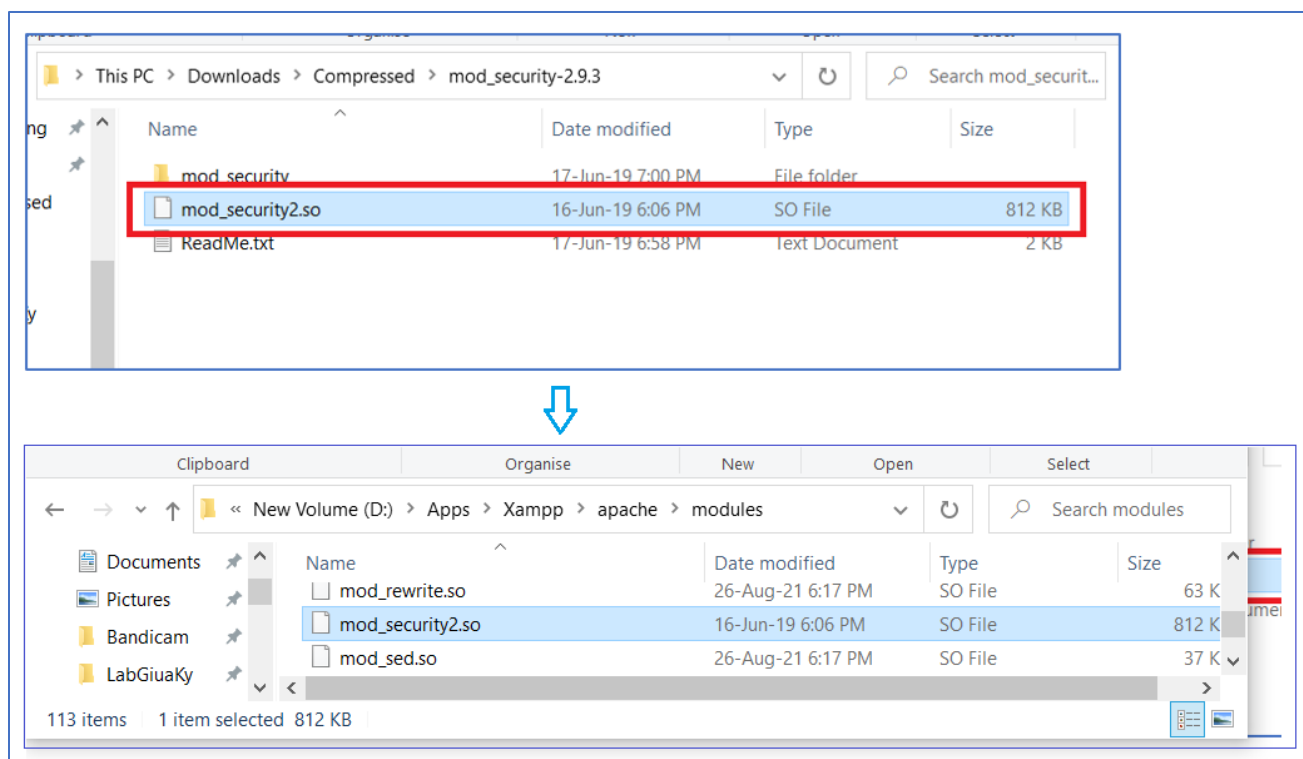
2.4 Cách cài đặt và cấu hình ModSecurity



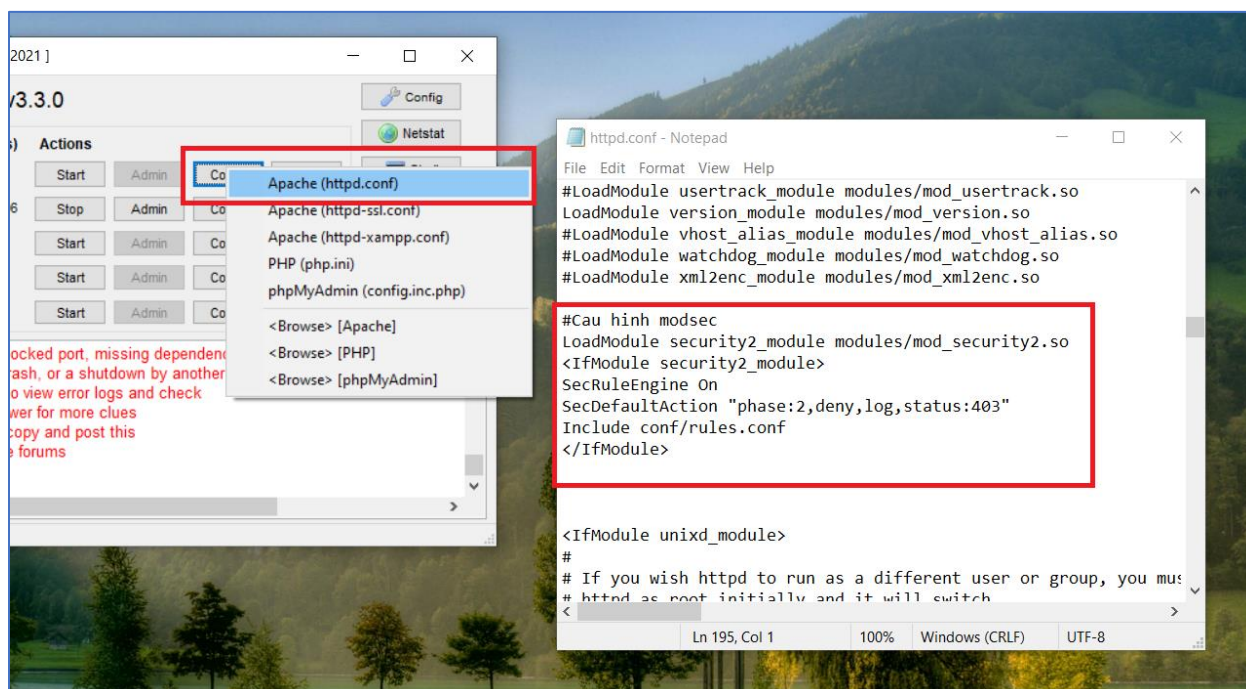
Hình 2.3.1 Đầu tiên nhóm em sẽ vào trang download ModSecurity và tải bản 2.9.3 cho win 64bit



Hình 2.3.2 Giải nén file vừa tải , copy file yajl.dll vào thư mục Xampp -> \apache\bin



Hình 2.3.3 Trong thư mục ModSecurity vừa giải nén ở bước trên, vào thư mục `mod_security-2.9.3` và copy file `mod_security2.so` dán vào thư mục chứa Xampp -> `\apache\modules`



Hình 2.3.4 Vào file `httpd.conf` của Apache trên Xampp và thêm các đoạn mã sau

Giải thích đoạn code trên:

`LoadModule security2_module modules/mod_security2.so` (load file `mod_security2.so` vừa copy ở hình 2.3.3)

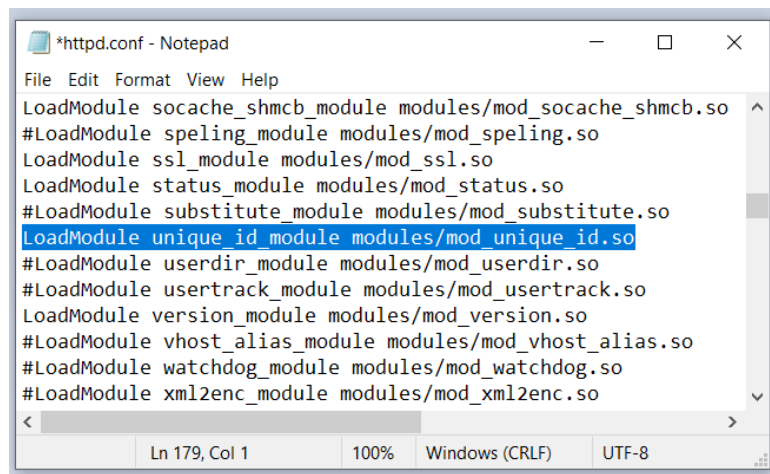
`<IfModule security2_module>`

`SecRuleEngine ON` (bật ModSecurity)

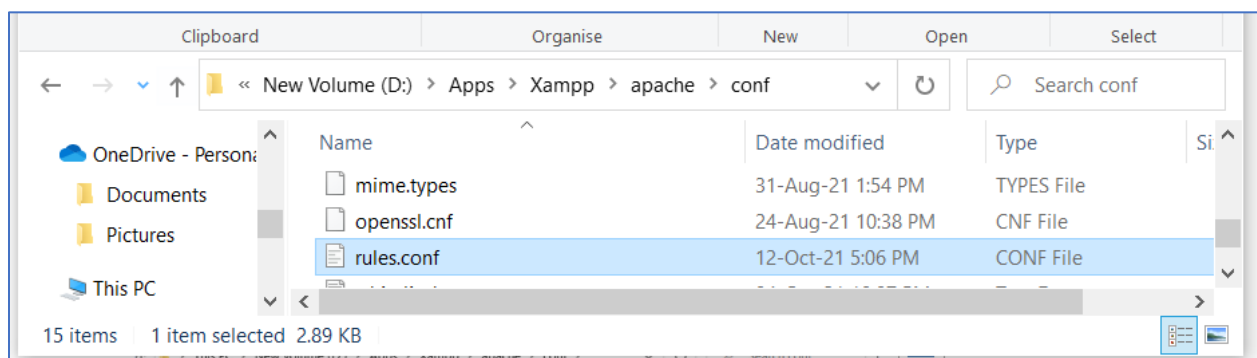
`SecDefaultAction "phase:2,deny,log,status:403"` (nếu có chặn thì sẽ ghi log và trả về mã lỗi 403, phase 2 là gì nhóm em sẽ đề cập ở phần 2.4.2)

`Include conf/rules.conf` (load file cấu hình các rule để chặn tấn công xss.conf)

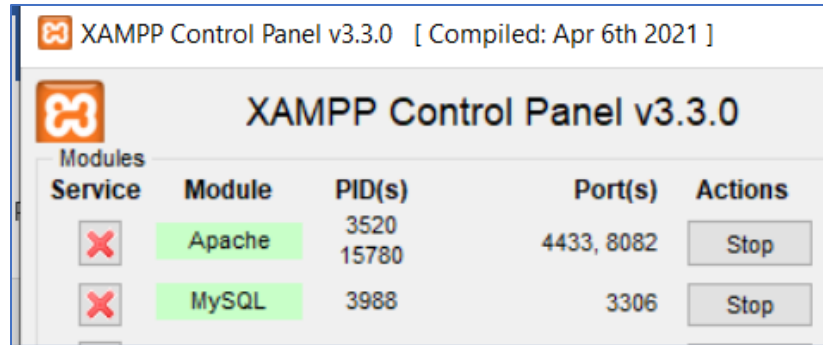
`</IfModule>`



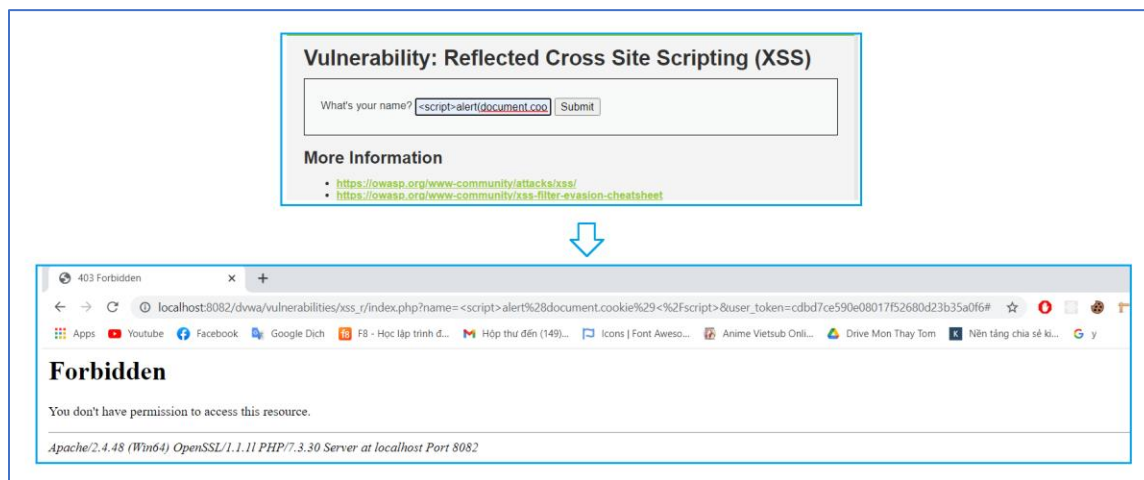
Hình 2.3.5 Và cũng phải mở comment dòng này thì nó mới chạy được



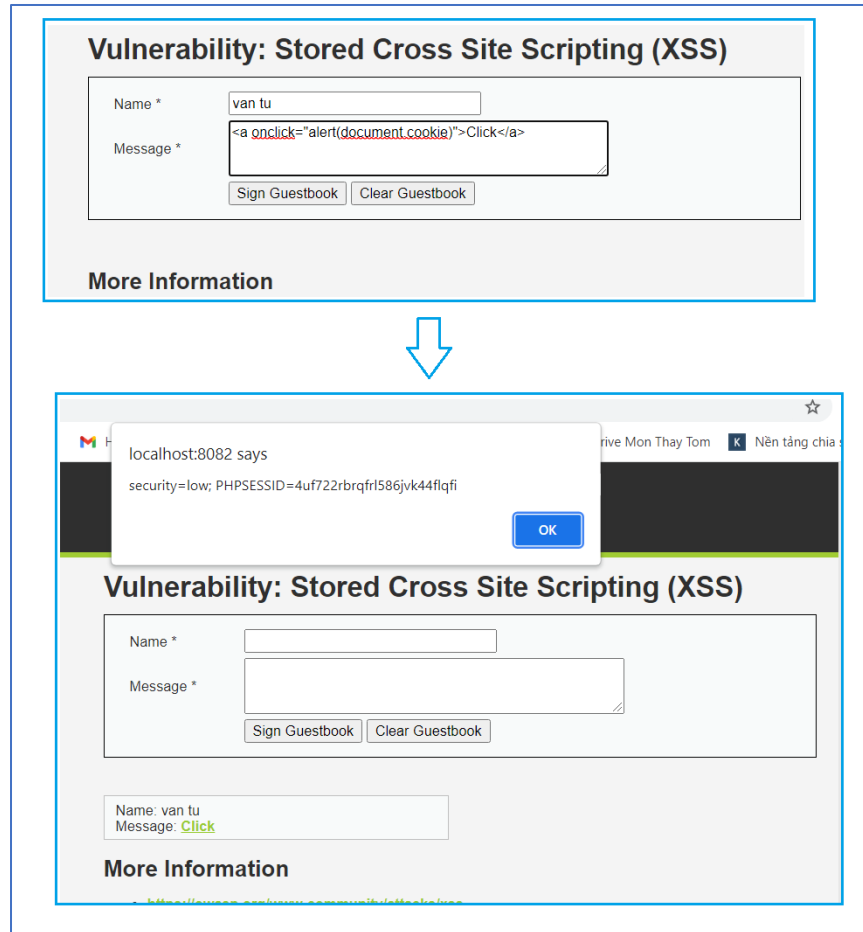
Hình 2.3.6 Tiếp theo, vào thư mục chứa Xampp -> \apache\conf -> tạo file rules.conf để chứa các rule



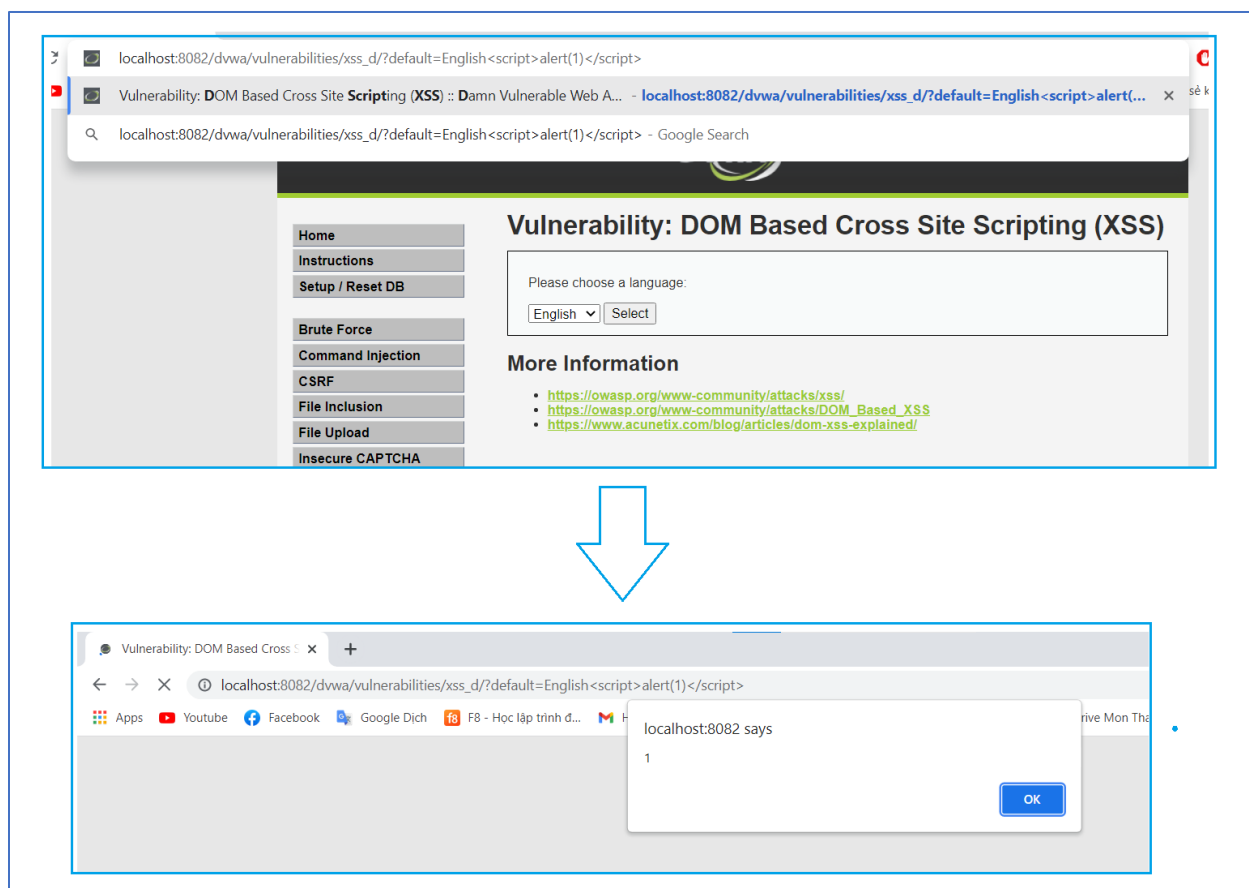
Hình 2.3.8 Cuối cùng, khởi động lại apache và MySQL trên Xampp và vào trang DVWA để kiểm tra



Hình 2.3.9 Kết quả là chặn được Reflect XSS ở mức low



Hình 2.3.10 Tuy nhiên không thể chặn được Stored XSS



Hình 2.3.11 Và cũng không thể chặn được DOM XSS

2.5 Tư viết một bộ rule cho ModSecurity

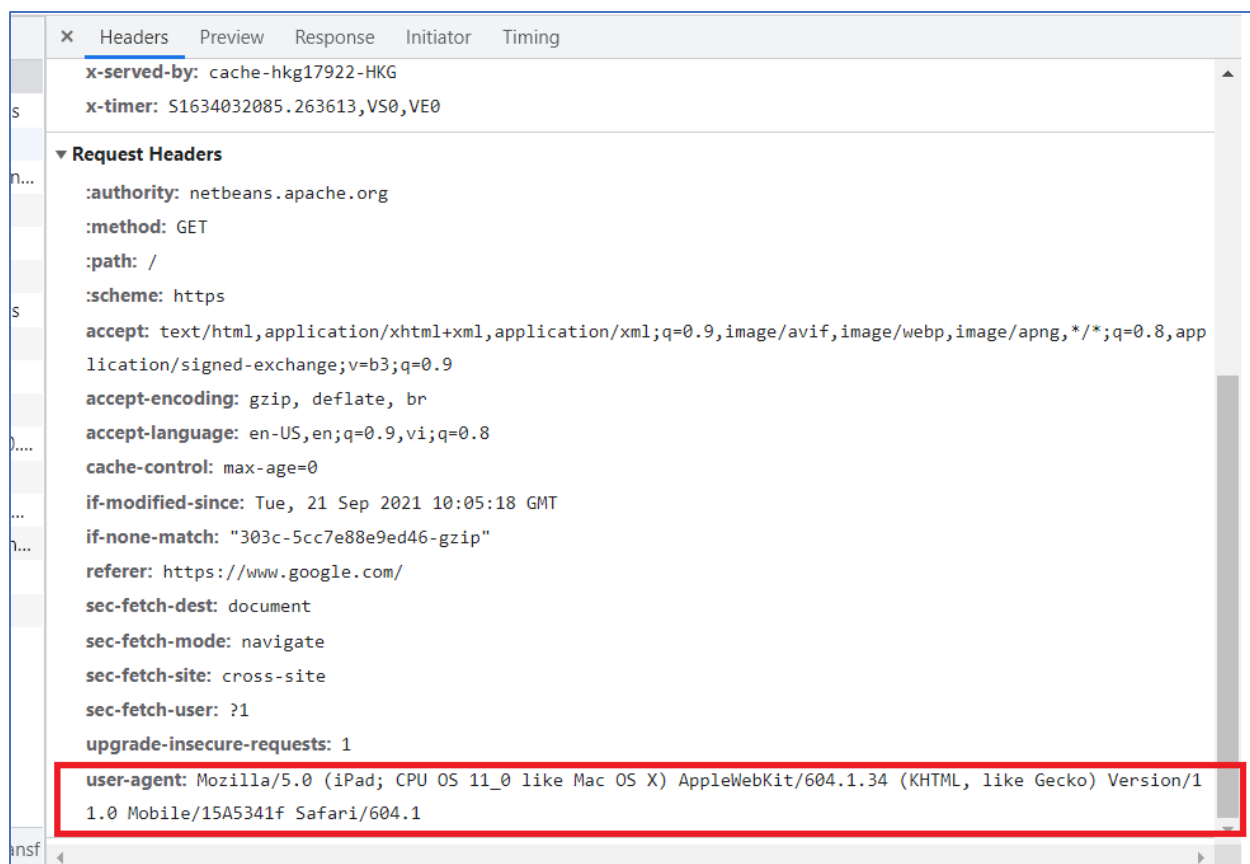
2.5.1 Cấu trúc của một rule

Cú pháp: **SecRule** **VARIABLES** **OPERATOR** [**ACTIONS**]

Trong đó:

SecRule là từ khóa

VARIABLES chính là cái mục tiêu mà rule sẽ áp dụng lên, **VARIABLES** có thể là một tập hợp hoặc một đối tượng cụ thể vd **REQUEST_HEADER**: toàn bộ dữ liệu trong Request Header, **REQUEST_HEADERS>User-Agent** là User-Agent ở trong Request Header



Hình 2.4.1.1 Trường user-agent trong Request Header

OPERATOR là nơi ghi các regex (biểu thức chính quy) hoặc keyword để kiểm tra giá trị các biến. Nó dùng để so sánh nếu khớp dữ liệu thì kích hoạt **[ACTIONS]**

[ACTIONS] là danh sách hành động nào sẽ được thực hiện khi mà rule được kích hoạt, nó có thể allow, deny các request và trả về các mã trạng thái về client, vv..., **[ACTIONS]** có thể có hoặc không, nếu không chỉ rõ action nào thì các default action (**SecDefaultAction**) sẽ được thực hiện.

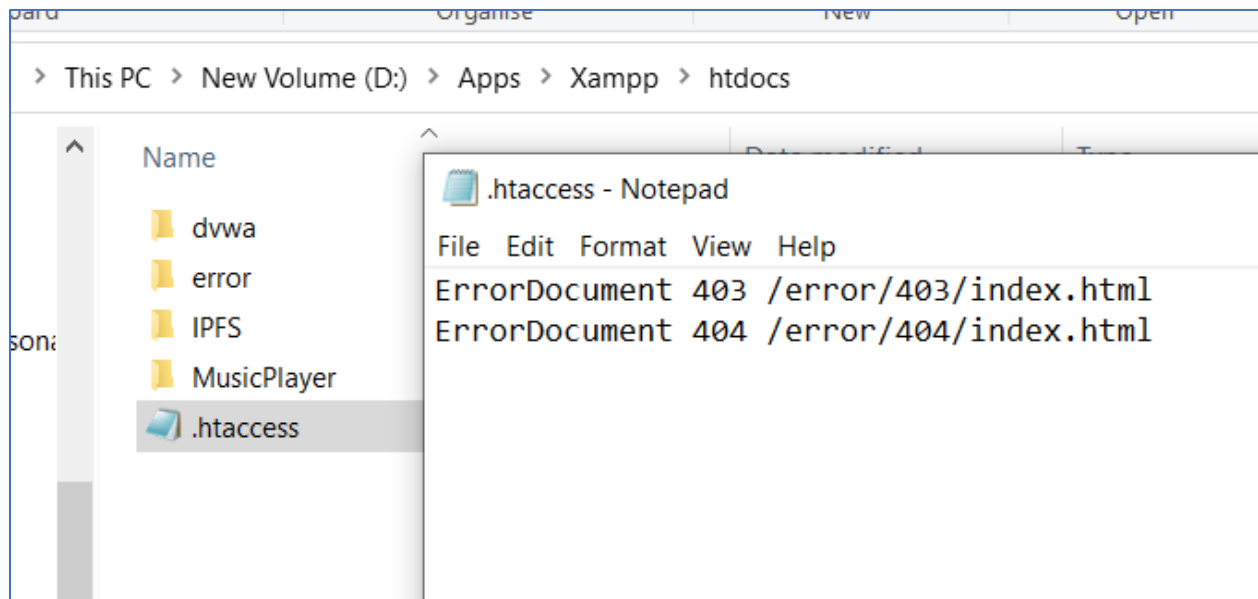
Default Action sẽ thực hiện khi có 1 rule không có action hoặc action của rule đó trùng với action của default. Cú pháp: **SecDefaultAction** **[ACTIONS]**

Nguồn:

<https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual-%28v2.x%29>

2.5.2 Một số ví dụ

Đây là video demo, mời cô xem qua: <https://www.youtube.com/watch?v=0js-6AuS1eI> (cô chỉnh 1080p60 để có chất lượng xem tốt nhất ạ)



Hình 2.4.2.1 Vào thư mục htdocs của Xampp, tạo file .htaccess và thêm các dòng mã chỉ dẫn file giao diện báo lỗi

Chặn theo từ khóa trên URI

```
SecRule REQUEST_URI "hacker" "id:1,deny,status:403,msg:'Duong dan co chu hacker'"
```

Chặn theo User Agent

```
SecRule REQUEST_HEADERS:User-Agent "@pm Mozilla Chrome" "deny,nolog,id:3"
```

Chặn theo dải địa chỉ IP

```
SecRule REMOTE_ADDR "^192\.168\.1\.[3-5]$" "id:1,deny,status:403" (từ .3 – .5)
```

Dùng tool online sau để tạo ra regex <https://www.analyticsmarket.com/freetools/ipregex/>

Chặn theo giờ

```
SecRule TIME_HOUR "@lt 18" "id:4,deny,status:406"
```

Chặn cùng lúc nhiều điều kiện trên URI (chain)

```
SecRule REQUEST_URI passwd "id:1,status:403,deny,chain"  
SecRule REMOTE_ADDR "^192.168.1.4" "chain"  
SecRule TIME_HOUR "@gt 18"
```

Chặn bằng cách đọc nội dung biến gửi theo method POST

`SecRequestBodyAccess ON`

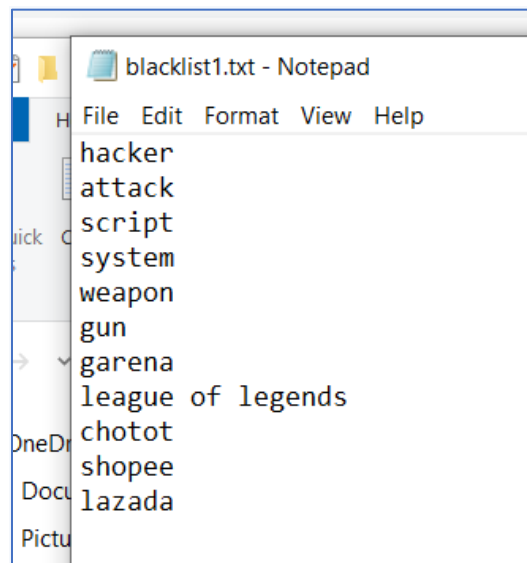
```
SecRule REQUEST_BODY "!^[A-Za-z0-9]+$" "id:2,deny,status:403,msg:'Reflect XSS Attack  
Detection!!!',phase:2"
```

Phase:

- Phase Request Header (phase: 1): Rule được đặt tại đây sẽ được thực hiện ngay sau khi Apache đọc request header, lúc này phần request body vẫn chưa được đọc.
- Phase Request Body (phase: 2): Lúc này các request argument và phần request body đã được đọc
- Phase Response Header (phase: 3): Đây là thời điểm ngay sau khi phần response header được gửi trả về cho client.
- Phase Response Body (phase: 4): Đây là lúc những dữ liệu HTML gửi trả về
- Logging: đây là thời điểm các hoạt động log được thực hiện, các rules đặt ở đây sẽ định rõ việc log sẽ như thế nào, nó sẽ kiểm tra các error message log của Apache.

Chặn theo file blacklist

```
SecRule ARGS|REQUEST_URI "@pmFromFile blacklist1.txt blacklist2.txt"  
"id:3,deny,msg:'Reflect XSS Attack in black list detection!!!',status:403,phase:2"
```



Hình 2.4.1.2 File blacklist1.txt chứa các từ khóa mà người dùng muốn chặn

2.5.3 Chặn các tấn công XSS ở mức low

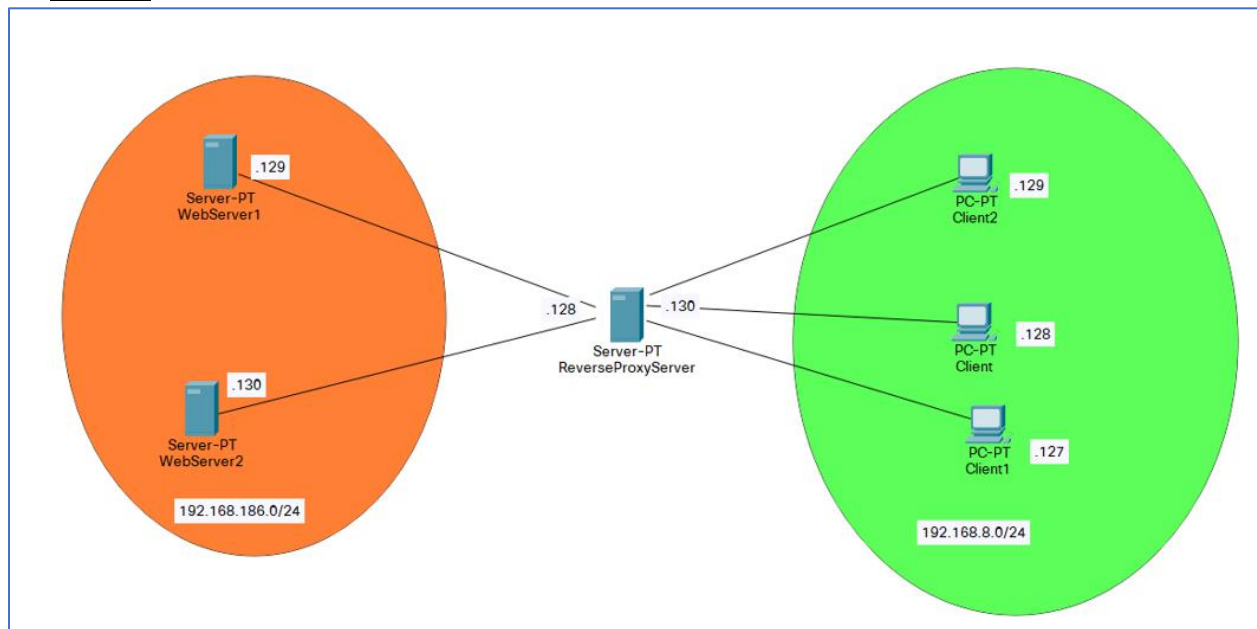
#Chặn các tấn công XSS ở mức low (%28 %29 là mã hex của dấu < >)

SecRequestBodyAccess ON

SecRule ARGS|REQUEST_BODY "@pm < > %28 %29" "id:10,deny,status:403"

CHƯƠNG 3. YÊU CẦU NÂNG CAO

3.1 Mô hình



Hình 3.1.1 Mô hình làm việc trong yêu cầu nâng cao (dùng Ubuntu 18)

3.2 Cài đặt và cấu hình DVWA

3.2.1 Cài đặt các packages yêu cầu (apache2, mysql và php) trước khi cài đặt DVWA

```
sudo apt install apache2 mysql-server php php-mysqli php-gd libapache2-mod-php git
```

3.2.2 Tải DVWA từ github và lưu vào địa chỉ /var/www/html/

```
cd /var/www/html/
```

```
git clone --recursive https://github.com/ethicalhack3r/DVWA.git
```

3.2.3 Kích hoạt file cấu hình DVWA

Sau khi cài đặt DVWA thì mặc định dvwa sẽ cung cấp cho mình một file cấu hình mặc định được giấu dưới dạng file .dist (sẽ phải xóa .dist để nó hoạt động như bình thường) và nó sẽ được sử dụng lại đối với dvwa (chỉ sửa đổi một số cấu hình để phù hợp).

```
sudo cp config/config.inc.php.dist config/config.inc.php
```

3.2.4 Kích hoạt quyền cho thư mục uploads và config

Khi cài đặt DVWA thì một số file cần quyền root để có thể đọc và lưu file nên mình cần phải cấp quyền để nó hoạt động bình thường.

```
sudo chmod 757 /var/www/html/dvwa/hackable/uploads/
```

```
sudo chmod 757 /var/www/html/dvwa/config
```

3.2.5 Kích hoạt quyền cho file phpids_log.txt

Tương tự như đối với một số thư mục phía trên thì file `phpids_log.txt` cũng cần quyền root để có thể viết hay thay đổi giá trị.

```
sudo chmod 646 /var/www/html/dvwa/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt
```

3.2.6 Bật tính năng function safe_mode

```
sudo vi /etc/php/7.2/apache2/php.ini
```

Chuyển giá trị của `allow_url_include` từ Off thành On

3.2.7 Cấu hình mysql

Truy cập mysql để cấu hình tài khoản truy cập

```
sudo mysql -uroot
```

Bên trong mysql, nhập lần lượt các lệnh sau để thay đổi mật khẩu của `root` thành `p@ssw0rd`

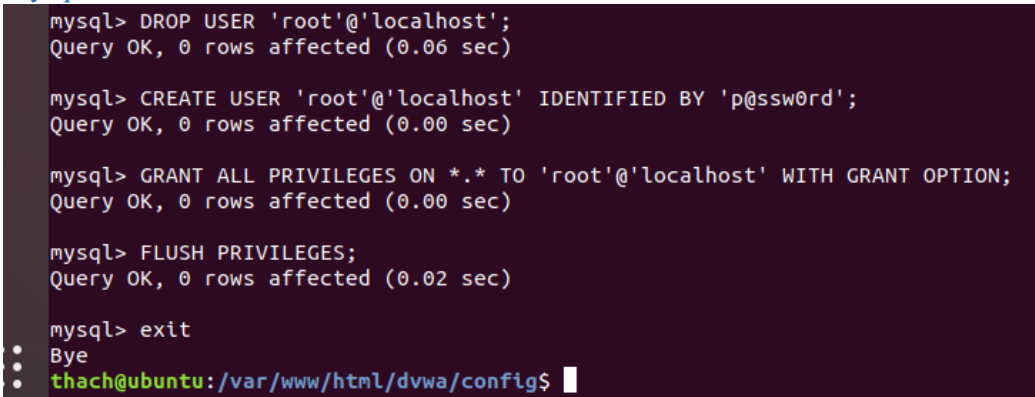
```
mysql> DROP USER 'root'@'localhost';
```

```
mysql> CREATE USER 'root'@'localhost' IDENTIFIED BY 'p@ssw0rd';
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> exit
```



```
mysql> DROP USER 'root'@'localhost';
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE USER 'root'@'localhost' IDENTIFIED BY 'p@ssw0rd';
Query OK, 0 rows affected (0.00 sec)

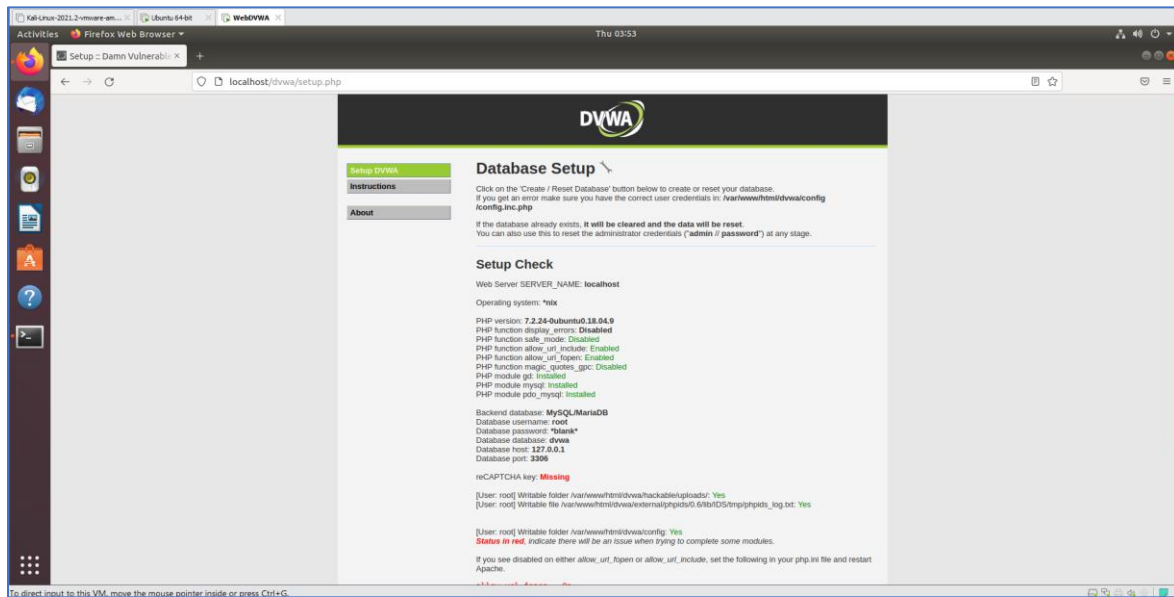
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> exit
• Bye
• thach@ubuntu: /var/www/html/dvwa/config$
```

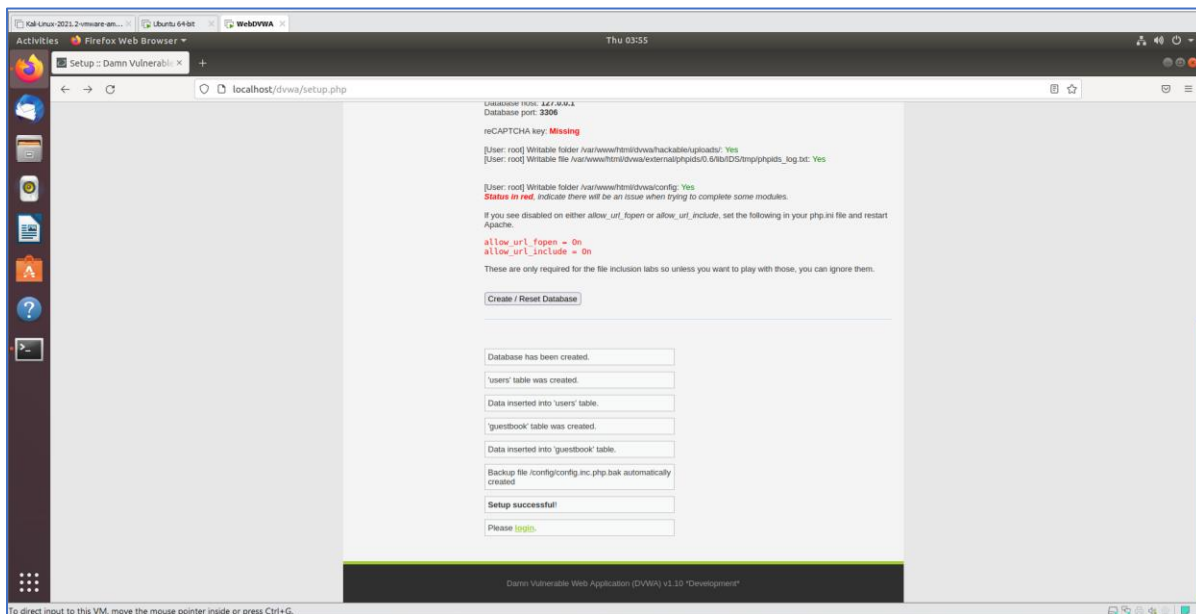
Hình 3.2.7.1 Cấu hình tài khoản mysql

3.2.8 Kiểm thử

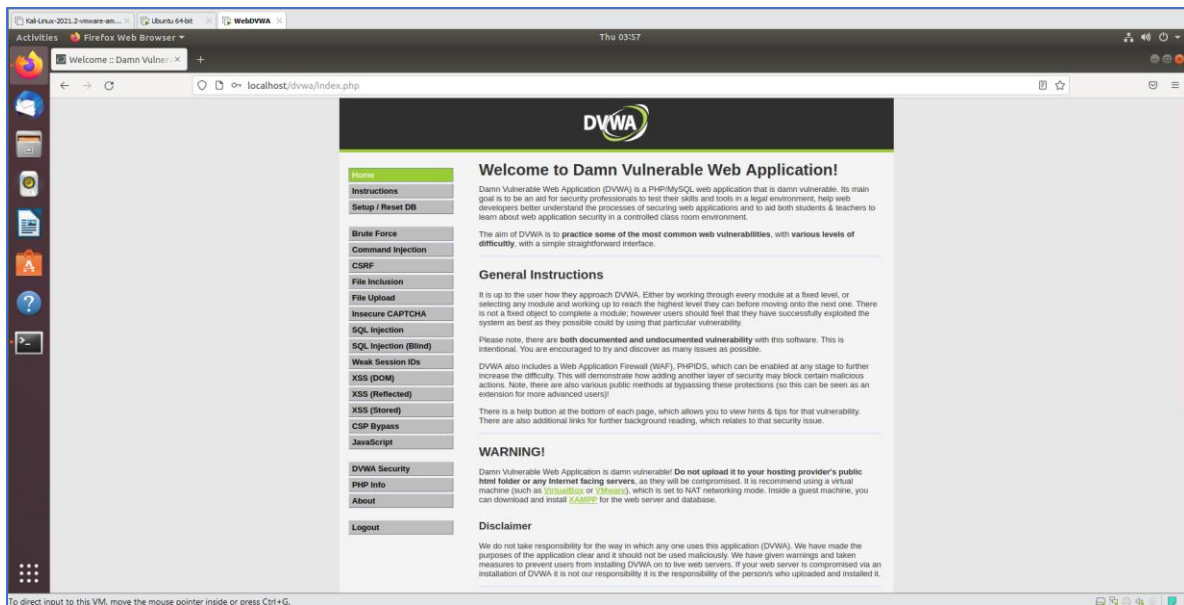


Hình 3.2.8.1 Truy cập vào địa chỉ DVWA để kiểm tra

3.2.9 Tạo database trong DVWA



Hình 3.2.9.1 Tạo database trong DVWA



Hình 3.2.9.2 Đăng nhập vào bằng tài khoản `admin` và mật khẩu là `password` sẽ xuất hiện giao diện chính của DVWA

3.3 Cài đặt Apache2 và cấu hình ReverseProxy

3.3.1 Cài đặt Apache2

`sudo apt install apache2`

3.3.2 Cấu hình VirtualHost cho web1

```

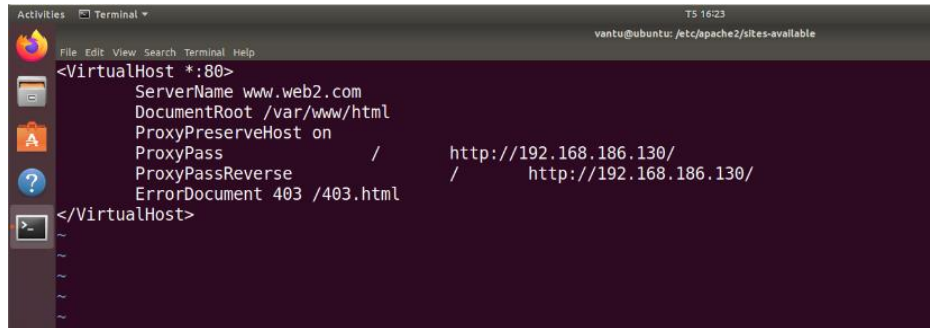
Activities  Terminal
File Edit View Search Terminal Help
ventu@ubuntu: /etc/apache2/sites-available

<VirtualHost *:80>
    ServerName www.web1.com
    DocumentRoot /var/www/html
    ProxyPreserveHost on
    ProxyPass / http://192.168.186.129/
    ProxyPassReverse / http://192.168.129/
    ErrorDocument 403 /403.html
</VirtualHost>

```

Hình 3.3.2.1 Cấu hình VirtualHost cho web1

3.3.3 Cấu hình VirtualHost cho web2



Hình 3.3.3.1 Cấu hình VirtualHost cho web2

3.3.4 Kích hoạt VirtualHost cho 2 web và proxy server

Lần lượt kích hoạt 2 virtualhost vừa tạo, disable trang mặc định của apache2 và bật tính năng proxy_http cho apache2.

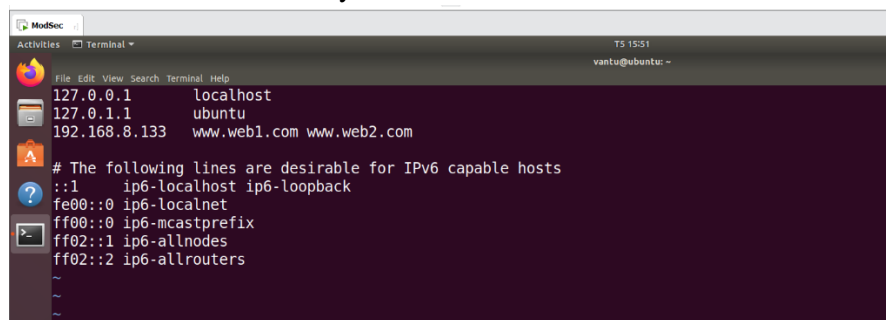
a2ensite www.web1.com

a2ensite www.web2.com

a2dissite 000-default.conf

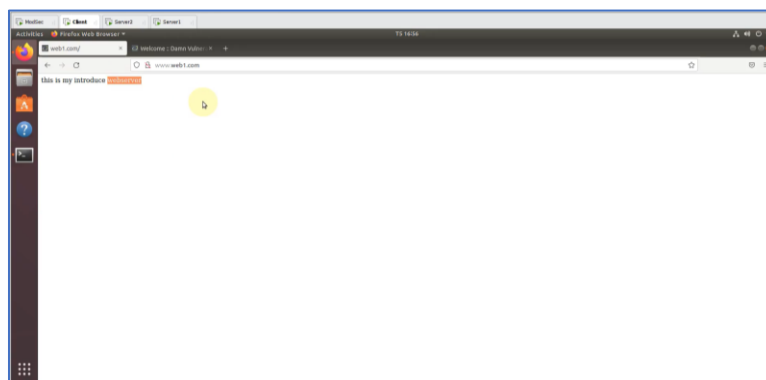
a2enmod proxy_http

3.3.5 Cấu hình DNS local cho ReverseProxy



Hình 3.3.5.1 Cấu hình để DNS local giúp phân giải tên miền www.web1.com và www.web2.com

3.3.6 Kiểm thử DVWA



Hình 3.3.6.1 Truy cập thành công vào web 1 với địa chỉ web1.com



Hình 3.3.6.2 Truy cập thành công vào web 2 với địa chỉ web2.com

3.4 Cài đặt và cấu hình ModSecurity

3.4.1 Cài đặt ModSecurity

`sudo apt install libapache2-mod-security2`

3.4.2 Cấu hình ModSecurity

Sử dụng lại file cấu hình mặc định của modsec (chỉ thay đổi một số giá trị bên trong) được lưu ẩn dưới dạng file có extension `.recommended` nên cần phải xóa extension này để file cấu hình hoạt động như bình thường.

`sudo cp /etc/modsecurity/modsecurity.conf.recommended /etc/modsecurity/modsecurity.conf`

`sudo vi /etc/modsecurity/modsecurity.conf`

Chuyển giá trị của **SecuRuleEngine** từ **DetectionOnly** thành **On**

Restart lại apache: `systemctl restart apache2`

3.4.3 Tạo rules

Tại đây, tất cả rules trên modsec sẽ được lưu ở folder và được lưu dưới dạng file có định dạng *.conf

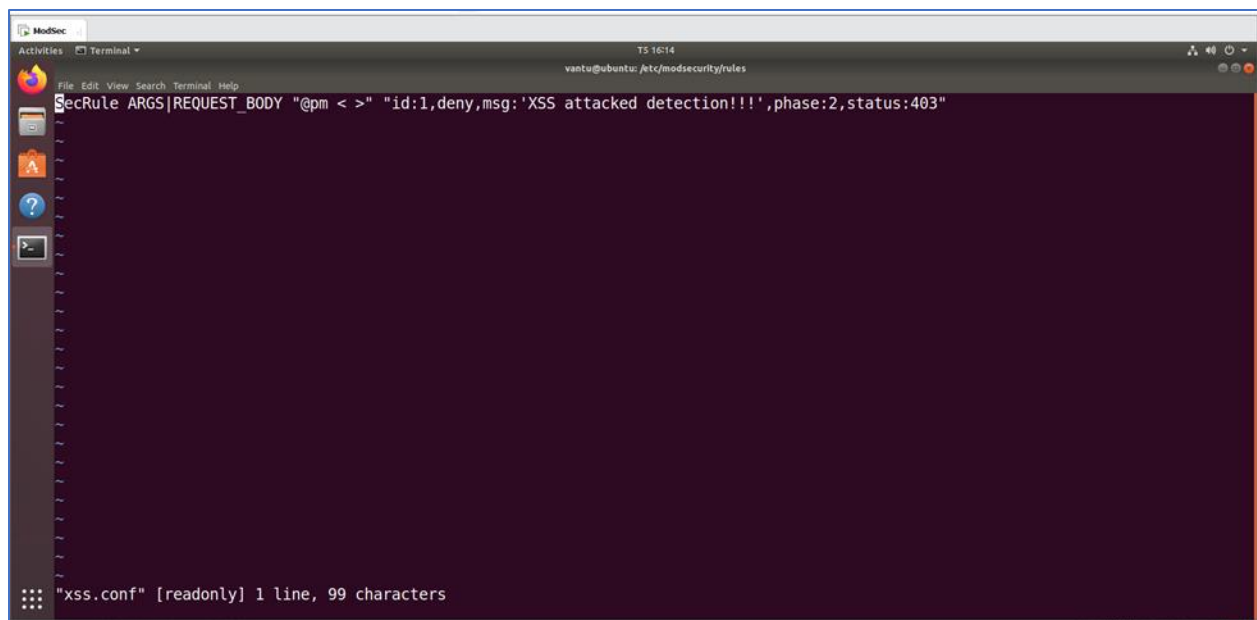
`cd /etc/modsecurity`

`mkdir rules`

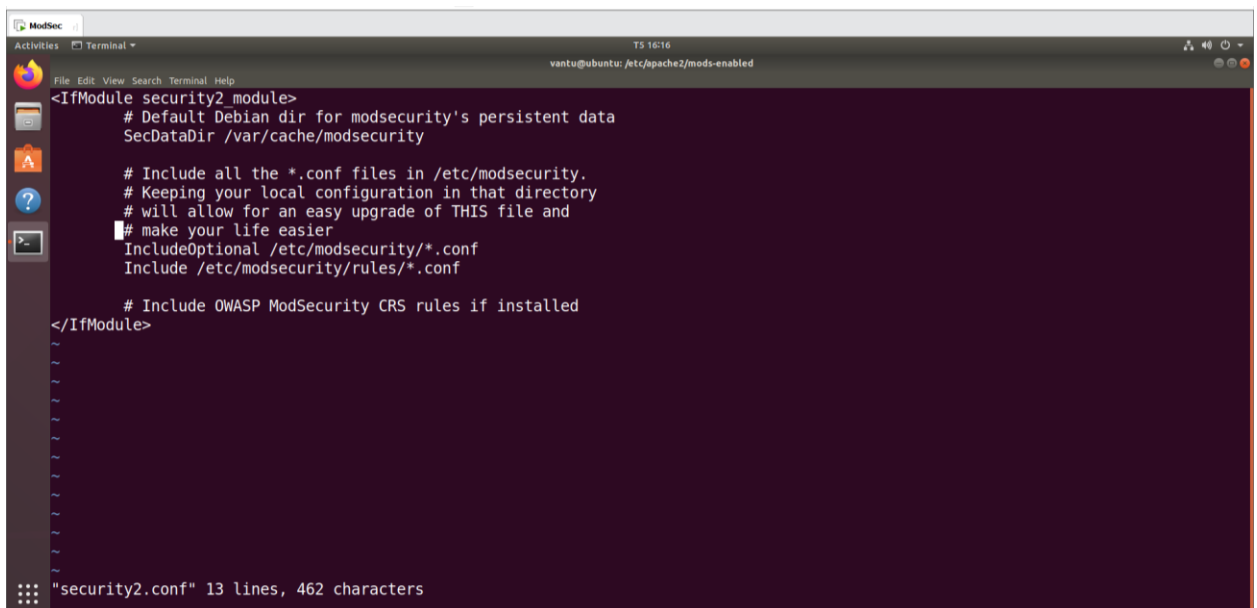
`cd rules`

`sudo vi xss.conf`

Viết rule vào đây (xss.conf)

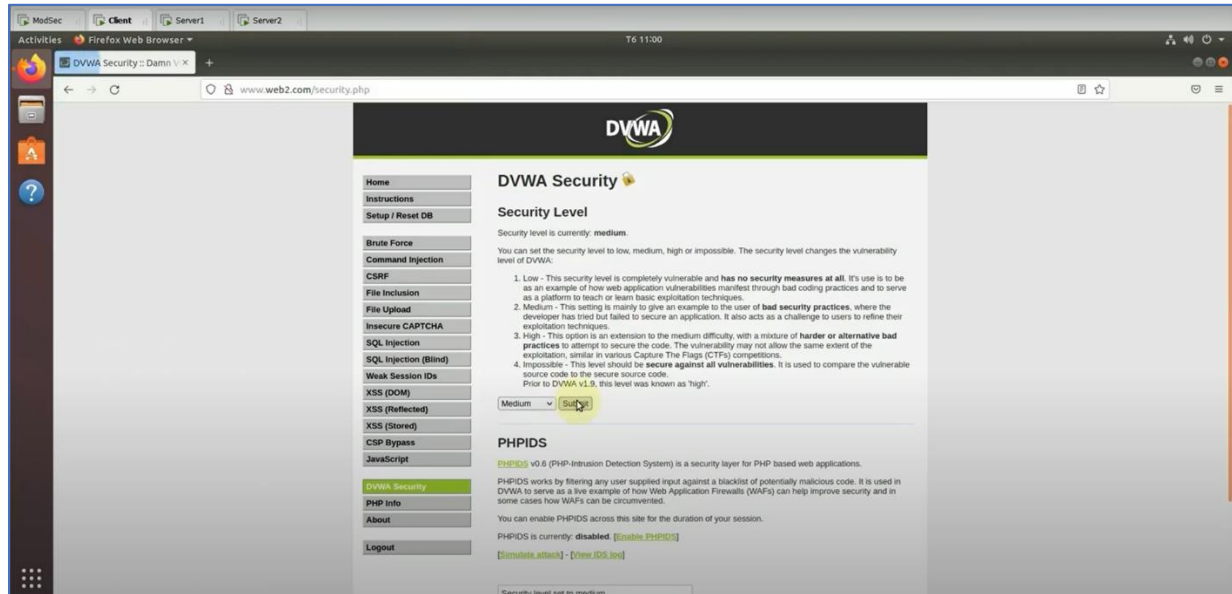


Hình 3.4.3.1 Rule chặn XSS

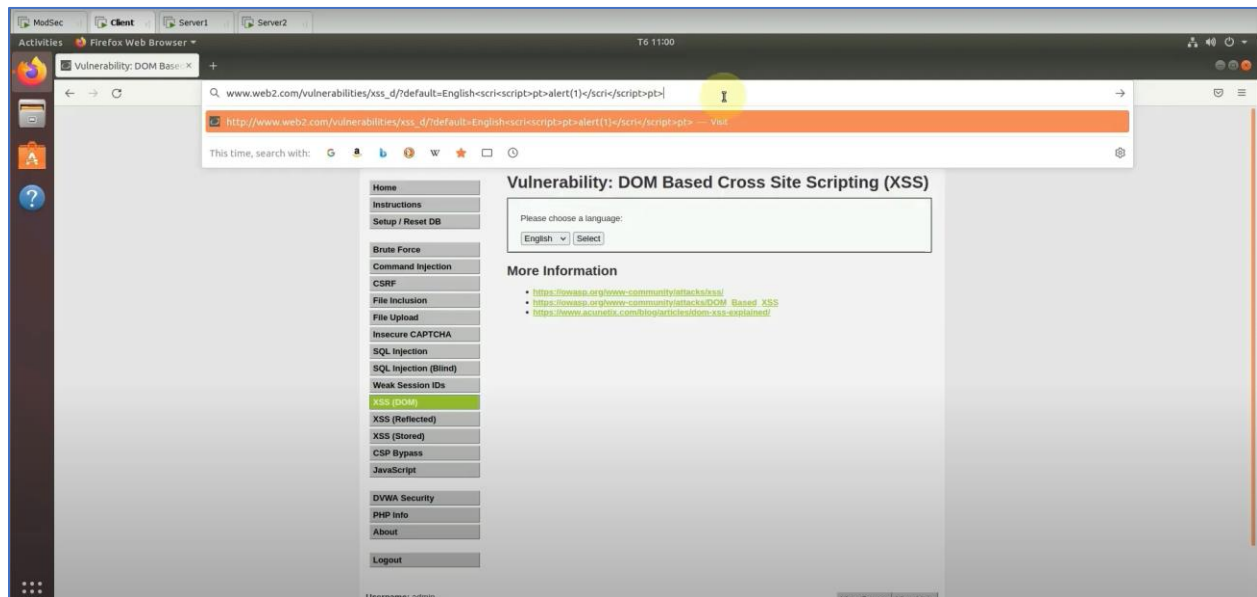


Hình 3.4.3.2 Trong file /etc/apache2/mods-enable, khai báo cho Apache nhận được file rule vừa tạo

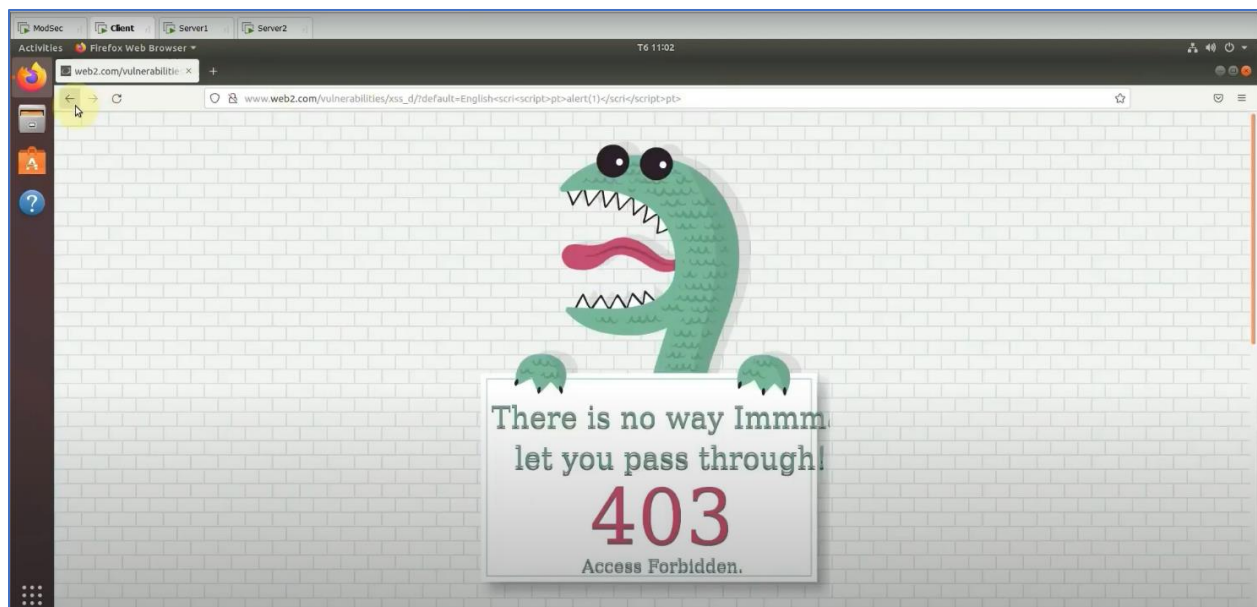
3.5 Kiểm thử ModSecurity (nhóm em sẽ test trên web server 2)



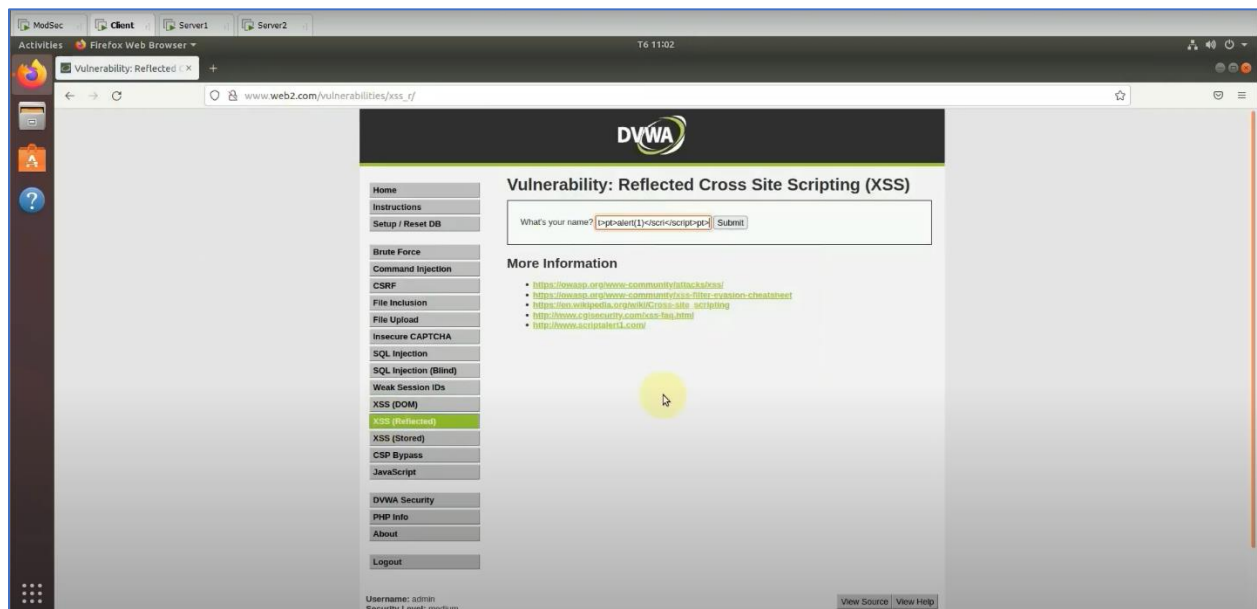
Hình 3.5.1 Truy cập vào DVWA và chỉnh về mức Medium



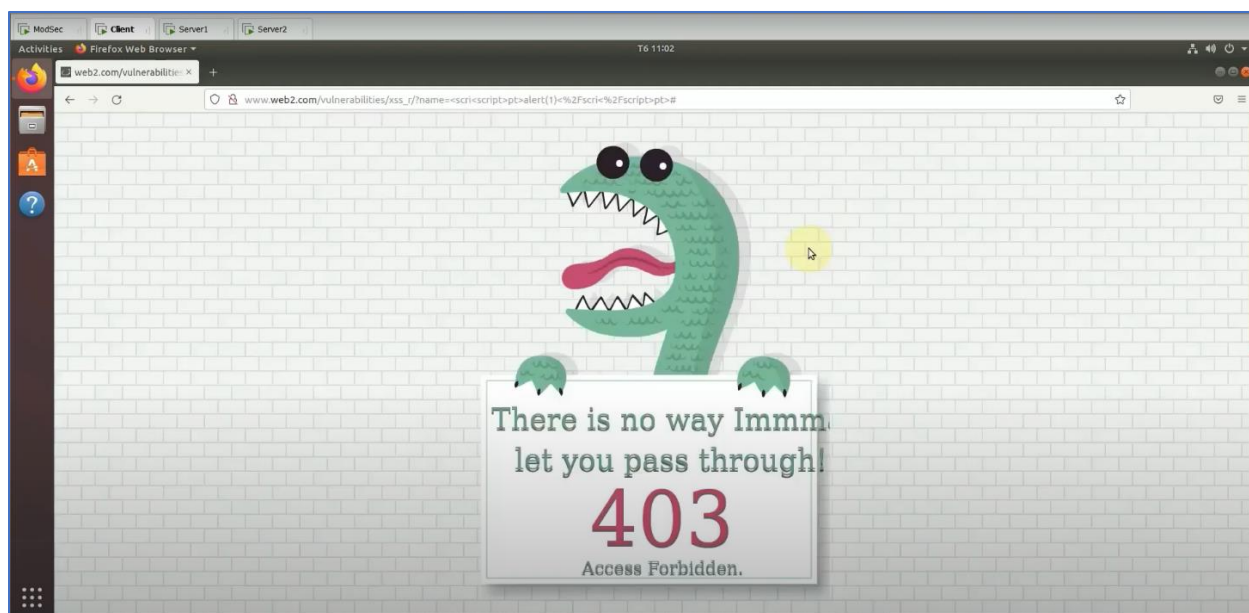
Hình 3.5.2 Tấn công DOM XSS bằng cách chèn đoạn lệnh `<scr<script>ipt>alert(1)<scr<script>ipt>` vào URL của trang web



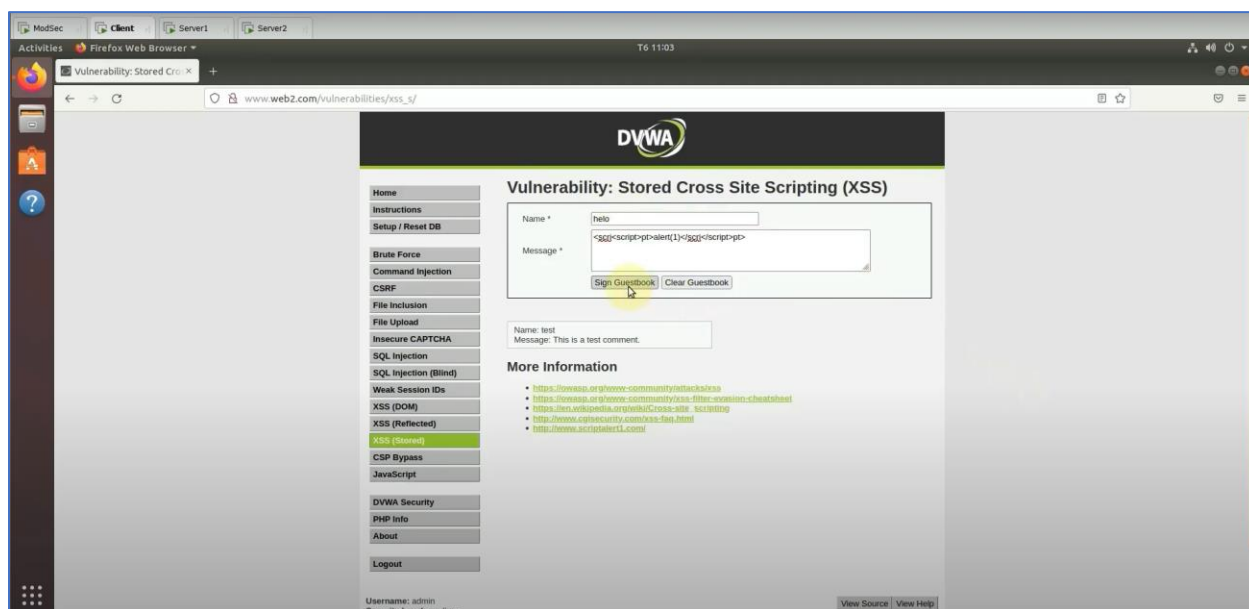
Hình 3.5.3 Kết quả chặn thành công



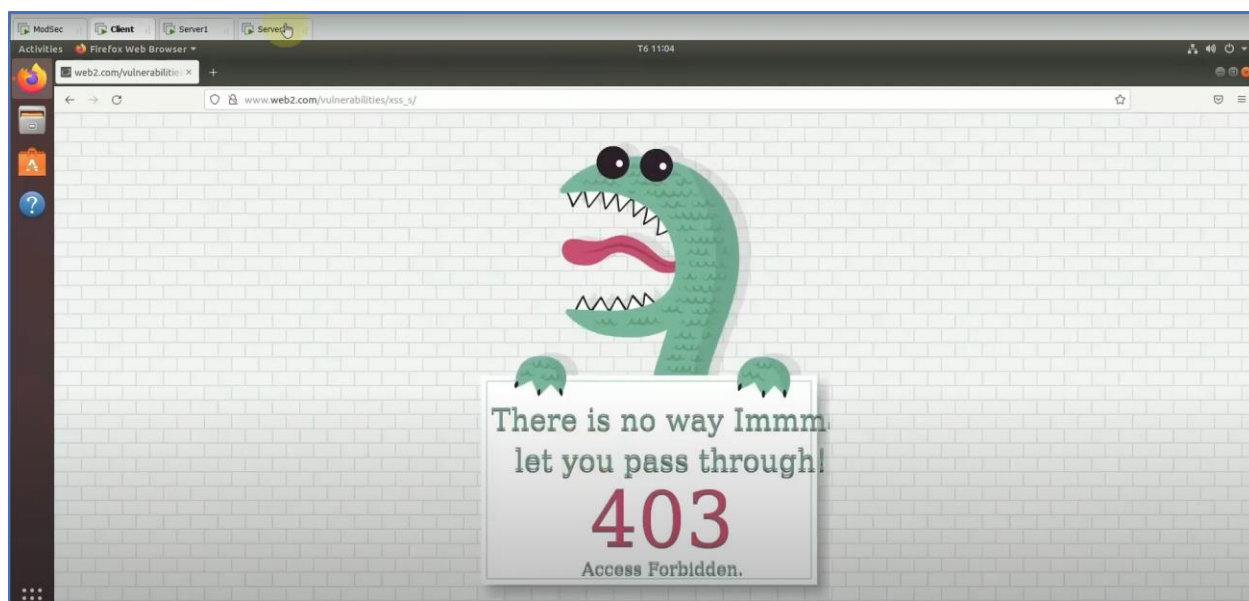
Hình 3.5.4 Tấn công Reflected XSS bằng cách chèn đoạn lệnh `<script>alert(1)</script>` vào phần input



Hình 3.5.5 Chặn thành công

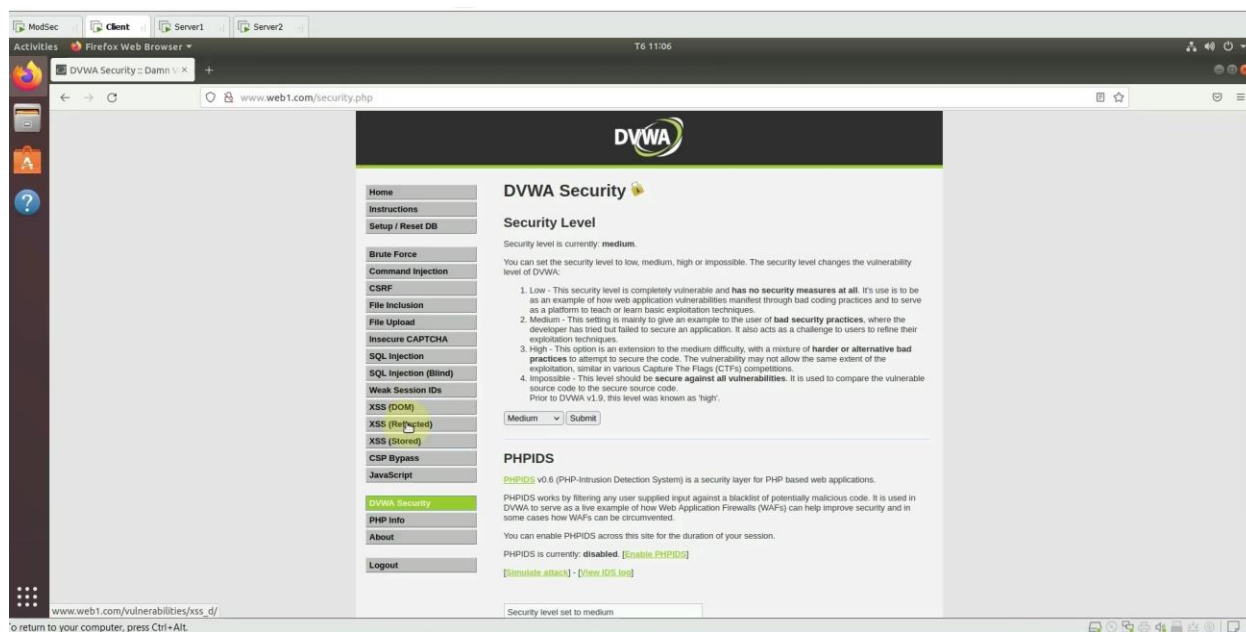


Hình 3.5.6 Tấn công Stored XSS bằng cách chèn đoạn lệnh `<scr<script>ipt>alert(1)<sc<script>ript>` vào ô message

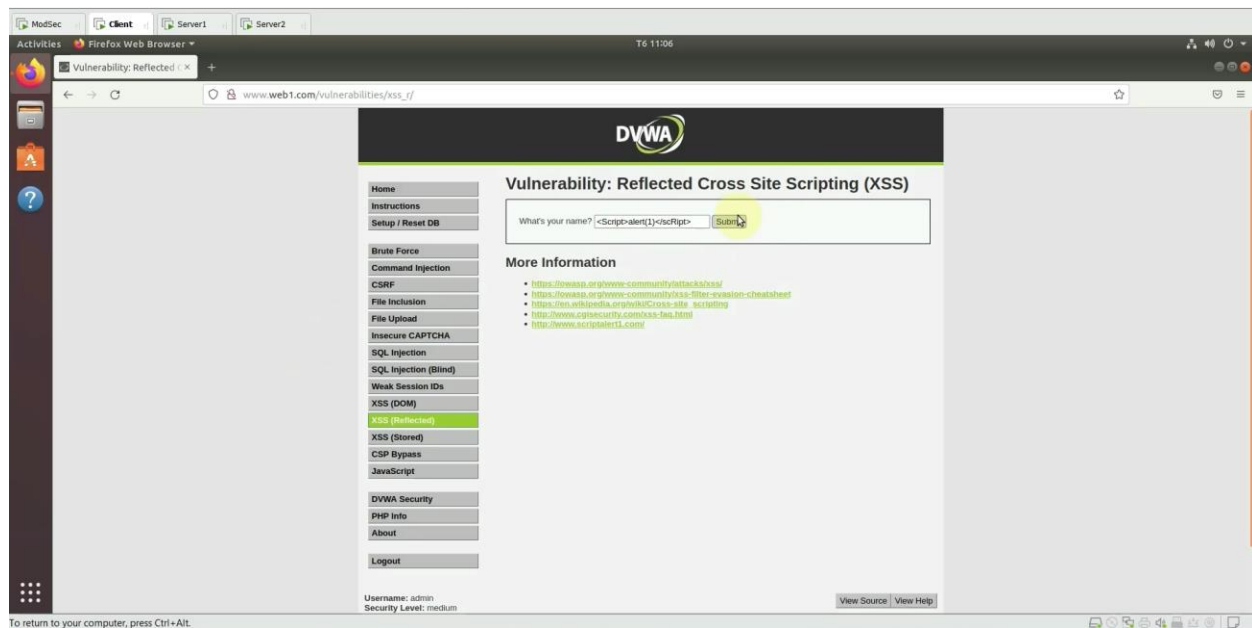


Hình 3.5.7 Kết quả chặn thành công

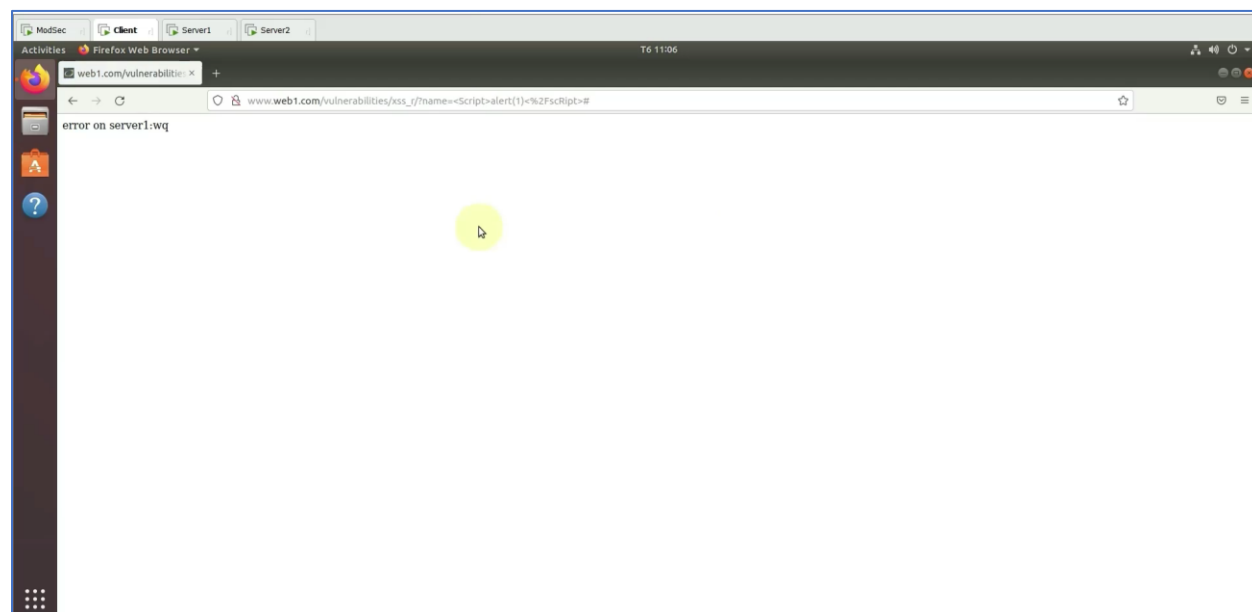
Chuyển sang web server 1 để test



Hình 3.5.8 Truy cập vào web của server 1 để kiểm thử, đưa DVWA về mức Medium



Hình 3.5.9 Vào phần Reflected XSS và thử tấn công bằng cách nhập đoạn lệnh `<Script>alert(1)</scRipt>`



Hình 3.5.10 Đã chặn được thành công tương tự như bên web server 2 (Đây là trang báo lỗi tui em viết nó hơi xấu)

⇒ Như vậy ModSecurity đã được tích hợp và hoạt động hiệu quả trên cả 2 con server cùng một lúc

CHƯƠNG 4. KẾT LUẬN

Bằng những kiến thức được học từ trên lớp kết hợp với việc tự nghiên cứu từ các nguồn bên ngoài, nhóm 3 đã đạt được những kết quả như sau; đối với yêu cầu cơ bản, nhóm em đã có thể tích hợp ModSecurity vào Xampp, hiểu cú pháp của một WAF (Web Application Firewall) rule, và tự viết được các rule chặn được các tấn công XSS ở mức độ Low, cũng như có thể tạo ra được các rule cho nhiều mục đích phổ thông như chặn theo địa chỉ IP, chặn theo thời điểm, chặn theo danh sách đen,... và tùy chỉnh giao diện báo lỗi bắt mắt hơn so với mặc định của trình duyệt.

Về yêu cầu nâng cao, nhóm em đã xây dựng được mô hình web Server – Client và Firewall đứng ở giữa bằng Reverse Proxy. Tích hợp được ModSecurity vào máy Firewall để khi Client gửi request đến Server hay cũng như khi Server response về Client cũng sẽ đều phải thông qua Firewall này.

CHƯƠNG 5. BẢNG PHÂN CHIA CÔNG VIỆC

Thành viên	Mã số sinh viên	Công việc đảm nhiệm
Lương Minh Tiến	N18DCAT069	Phân tích cách phòng thủ các tấn công XSS trong DVWA + Viết các rule + Viết word (phần 2.3 + 2.5)
Trần Văn Tư (Nhóm trưởng)	N18DCAT081	Cơ sở lý thuyết + Cách cài đặt và cấu hình DVWA, ModSecurity (chương 1 + chương 2)
Phạm Thạch	N18DCAT082	Làm yêu cầu nâng cao (chương 3)

Nhóm 3 xin chân thành cảm ơn cô Trần Thị Dung đã nhiệt tình giảng dạy trên lớp học cũng như luôn sẵn sàng giải đáp các thắc mắc ngoài giờ học của chúng em. Những kiến thức mà cô đem đến là nguồn lực vô cùng quan trọng và ý nghĩa để chúng em có thể hoàn thành được bài báo cáo này. Một lần nữa, chúng em xin cảm ơn cô Dung và chúc cô sẽ luôn gặp nhiều hạnh phúc trong cuộc sống.

----- Kết thúc -----