# Insecure CAPTCHE

Insecure CAPTCHE，意思是不安全的验证码，或者不安全的验证流程。这里使用的是谷歌的验证服务，首先申请该服务的API，获得公私钥对，这里需要自己申请，由于我们只是验证是否可以绕过验证，所以在config.inc.php中我们随意输入公私钥即可。这里的验证过程，首先向谷歌发送服务器发送请求验证码（利用公钥），返回验证码之后输入验证码给服务器，服务器利用私钥、用户IP、验证码等向谷歌验证验证码，然后谷歌返回结果。

# Level:Low

漏洞分析

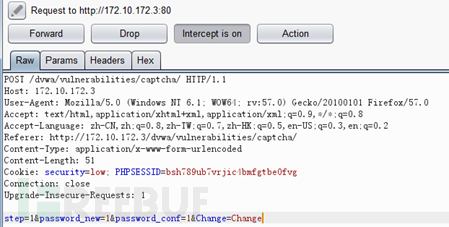
设置为Low等级，查看源码可以发现，服务器将改密操作分成两步，第一步是验证验证码是否正确，源码如下

<?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'],  
        $\_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( 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 ) : ((trigger\_error("[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($GLOBALS["\_\_\_mysqli\_ston"]) : (($\_\_\_mysqli\_res = mysqli\_connect\_error()) ? $\_\_\_mysqli\_res : false)) . '</pre>' );  
  
        // 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);  
}  
  
?>

第二步是在第一步验证成功且密码相同后返回表单，提交Post请求，完成修改密码的操作。

修改step绕过验证码认证过程

我们可以将请求拦截，拦截如下



这里将step参数改为2，则可以直接绕过，完成修改密码请求



利用CSRF绕过验证

我们已经得到只要发送step=2以及修改密码的表单就可以直接绕过验证码输入，然后修改密码，这里我们可以伪造一个提交step=2的修改密码表单的网页，具体的页面如下，这里不做具体操作，只提供思路



# Level:Medium

漏洞分析

设置为Medium等级，我们可以发现源码部分第二步POST请求增加了对第一步的验证结果参数。

<?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' ],  
        $\_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=\"hidden\" name=\"passed\_captcha\" value=\"true\" />  
                    <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;  
    }  
  
    // 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 ) : ((trigger\_error("[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($GLOBALS["\_\_\_mysqli\_ston"]) : (($\_\_\_mysqli\_res = mysqli\_connect\_error()) ? $\_\_\_mysqli\_res : false)) . '</pre>' );  
  
        // 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);  
}  
  
?>

利用添加passed\_captcha参数绕过验证

拦截修改请求，结果如下



将此处修改为step=2，并添加passed\_captcha字段，修改如下



得到结果为



# Level:High

漏洞分析

设置为High等级，源码修改如下，这里采用一次请求来进行修改密码，而且确定了验证失败的条件是返回为false，并且请求满足一定条件。

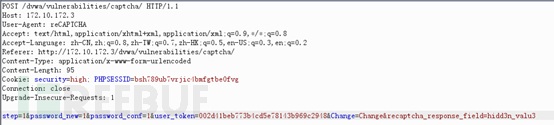
<?php  
  
if( isset( $\_POST[ 'Change' ] ) ) {  
    // Hide the CAPTCHA form  
    $hide\_form = true;  
  
    // Get input  
    $pass\_new  = $\_POST[ 'password\_new' ];  
    $pass\_conf = $\_POST[ 'password\_conf' ];  
  
    // Check CAPTCHA from 3rd party  
    $resp = recaptcha\_check\_answer(  
        $\_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 ) : ((trigger\_error("[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($GLOBALS["\_\_\_mysqli\_ston"]) : (($\_\_\_mysqli\_res = mysqli\_connect\_error()) ? $\_\_\_mysqli\_res : false)) . '</pre>' );  
  
            // Feedback for user  
            echo "<pre>Password Changed.</pre>";  
  
        } else {  
            // Ops. Password mismatch  
            $html     .= "<pre>Both passwords must match.</pre>";  
            $hide\_form = false;  
        }  
  
    } else {  
        // What happens when the CAPTCHA was entered incorrectly  
        $html     .= "<pre><br />The CAPTCHA was incorrect. Please try again.</pre>";  
        $hide\_form = false;  
        return;  
    }  
  
    ((is\_null($\_\_\_mysqli\_res = mysqli\_close($GLOBALS["\_\_\_mysqli\_ston"]))) ? false : $\_\_\_mysqli\_res);  
}  
  
// Generate Anti-CSRF token  
generateSessionToken();  
  
?>

利用修改参数来避免触发验证失败条件

这里我们无法决定验证码返回的结果，所以要想避开失败情况，则需要&&后面为假，则需要设置recaptcha\_response\_field为hidd3n\_valu3、HTTP\_USER\_AGENT为reCAPTCHA即可，拦截修改请求，截取如下



修改User-Agent，并添加recaptcha\_response\_field字段，修改如下



得到如下结果，说明漏洞存在



# Level:Impossible

<?php  
  
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 );  
    $pass\_new  = ((isset($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 );  
  
    $pass\_conf = $\_POST[ 'password\_conf' ];  
    $pass\_conf = stripslashes( $pass\_conf );  
    $pass\_conf = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"],  $pass\_conf ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));  
    $pass\_conf = md5( $pass\_conf );  
  
    $pass\_curr = $\_POST[ 'password\_current' ];  
    $pass\_curr = stripslashes( $pass\_curr );  
    $pass\_curr = ((isset($GLOBALS["\_\_\_mysqli\_ston"]) && is\_object($GLOBALS["\_\_\_mysqli\_ston"])) ? mysqli\_real\_escape\_string($GLOBALS["\_\_\_mysqli\_ston"],  $pass\_curr ) : ((trigger\_error("[MySQLConverterToo] Fix the mysql\_escape\_string() call! This code does not work.", E\_USER\_ERROR)) ? "" : ""));  
    $pass\_curr = md5( $pass\_curr );  
  
    // 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  
        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) AND password = (:password) LIMIT 1;' );  
        $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) WHERE user = (:user);' );  
            $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 passwords did not match.<br />Please try again.</pre>";  
            $hide\_form = false;  
        }  
    }  
}  
  
// Generate Anti-CSRF token  
generateSessionToken();  
  
?>

设置为Impossible等级，源码如下，增加了Anti-CSRF token，增加了原密码选项，并且采用PDO机制防止注入，而且验证采用一步且无法绕过，进一步加强了身份认证。