XcheCK

静态应用安全测试报告

Tencent腾讯

项目信息

项目名称：	DVWA-master.zip
扫描地址：	zip://DVWA-master.zip
提交人　：	indiv

扫描信息

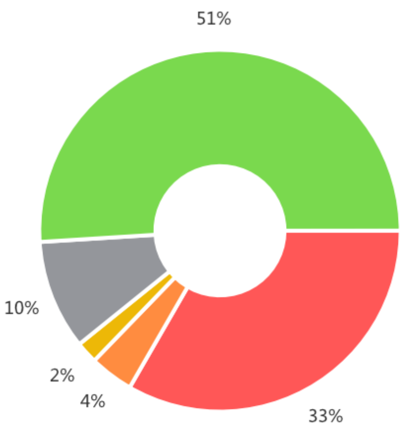
扫描耗时：	37.34 s
扫描语言：	py,php,java,go,js,cpp
创建时间：	2022-01-20 11:30:37

漏洞概览

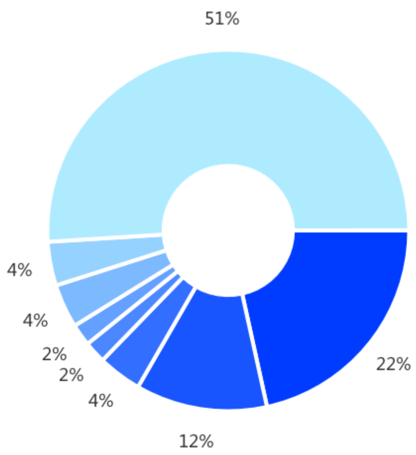
漏洞类型	漏洞等级	漏洞数量
SQL注入漏洞	● 紧急	11
任意命令执行漏洞	● 紧急	6
文件包含漏洞	● 高危	2
服务端请求伪造漏洞	● 中危	1
信息泄露	● 低危	1
跨站脚本攻击漏洞	● 低危	2
目录穿越漏洞	● 低危	2
提示性风险（可忽略）	● 提示	26

漏洞分布

等级分布



类型分布



● 紧急 ● 高危 ● 中危 ● 低危 ● 提示

● SQL注入漏洞 ● 任意命令执行漏洞
● 文件包含漏洞 ● 服务端请求伪造漏洞
● 信息泄露 ● 跨站脚本攻击漏洞 ● 目录穿越漏洞
● 提示性风险（可忽略）

SQL注入漏洞（11个）

漏洞描述

SQL注入是网站存在最多也是最简单的漏洞，原因是应用程序在处理用户输入没有过滤, 转义, 限制或处理不严谨, 导致用户可以通过输入精心构造的字符串去非法获取到数据库中的数据

1. 获取敏感数据, 修改数据库数据(插入, 更新, 删除), 执行数据库管理操作(如关闭数据库管理系统)等
2. 在某些情况下能执行系统命令, 进而直接获取数据库服务器的系统权限

修复方案

1. SQL查询语句使用标准化的数据库API接口, 设定语句的参数进行过滤一些非法的字符, 防止用户输入恶意的字符传入到数据库中执行SQL语句
2. 对用户提交的的参数安全过滤, 像一些特殊的字符[,()*&%#] 进行字符转义操作, 以及编码的安全转换
3. 网站的代码层编码尽量统一, 建议使用utf8编码, 如果代码里的编码都不一样, 会导致一些过滤被直接绕过

参考链接

http://159.75.6.40:8083/docs#SQL注入漏洞

SQL注入漏洞1

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/brute/source/low.php:13

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/brute/source/low.php:5

`$user`是污点来源

```
$user = $_GET[ 'username' ];
```

漏洞链路2: DVWA-master/vulnerabilities/brute/source/low.php:12

污点从`$user`传递至`$query`

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

漏洞链路3: DVWA-master/vulnerabilities/brute/source/low.php:13

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($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>' );
```

SQL注入漏洞2

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli/source/low.php:11

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/sqli/source/low.php:5

`$id`是污点来源

```
$id = $_REQUEST[ 'id' ];
```

漏洞链路2: DVWA-master/vulnerabilities/sqli/source/low.php:10

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
```

漏洞链路3: DVWA-master/vulnerabilities/sqli/source/low.php:11

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($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>' );
```

SQL注入漏洞3

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli/source/low.php:34

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/sqli/source/low.php:5

`$id`是污点来源

```
$id = $_REQUEST[ 'id' ];
```

漏洞链路2: DVWA-master/vulnerabilities/sqli/source/low.php:31

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
```

漏洞链路3: DVWA-master/vulnerabilities/sqli/source/low.php:34

`SQLI`类型风险触发，由入参`$query`导致

```
$results = $sqlite_db_connection->query($query);
```

SQL注入漏洞4

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli/source/medium.php:12

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/sqli/source/medium.php:5

`$id`是污点来源

```
$id = $_POST[ 'id' ];
```

漏洞链路2: DVWA-master/vulnerabilities/sqli/source/medium.php:7

污点从`$id`传递至`$id`

```
$id = mysqli_real_escape_string($GLOBALS["___mysqli_ston"], $id);
```

漏洞链路3: DVWA-master/vulnerabilities/sqli/source/medium.php:11

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = $id;";
```

漏洞链路4: DVWA-master/vulnerabilities/sqli/source/medium.php:12

`SQLI`类型风险触发，由入参`$query`导致

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

SQL注入漏洞5

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli/source/medium.php:30

在线结果： [点击查看]

污染传递链路

漏洞链路1：DVWA-master/vulnerabilities/sqli/source/medium.php:5

`$id`是污点来源

```
$id = $_POST[ 'id' ];
```

漏洞链路2：DVWA-master/vulnerabilities/sqli/source/medium.php:7

污点从`$id`传递至`$id`

```
$id = mysqli_real_escape_string($GLOBALS["___mysqli_ston"], $id);
```

漏洞链路3：DVWA-master/vulnerabilities/sqli/source/medium.php:27

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = $id;";
```

漏洞链路4：DVWA-master/vulnerabilities/sqli/source/medium.php:30

`SQLI`类型风险触发，由入参`$query`导致

```
$results = $sqlite_db_connection->query($query);
```

SQL注入漏洞6

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli_blind/source/high.php:12

在线结果： [点击查看]

污染传递链路

漏洞链路1：DVWA-master/vulnerabilities/sqli_blind/source/high.php:5

`$id`是污点来源

```
$id = $_COOKIE[ 'id' ];
```

漏洞链路2：DVWA-master/vulnerabilities/sqli_blind/source/high.php:11

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id' LIMIT 1;";
```

漏洞链路3：DVWA-master/vulnerabilities/sqli_blind/source/high.php:12

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ); // Removed 'or die' to suppress mysql
 errors
```

SQL注入漏洞7

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli_blind/source/high.php:28

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/vulnerabilities/sqli_blind/source/high.php:5

`$id`是污点来源

```
$id = $_COOKIE[ 'id' ];
```

漏洞链路2：DVWA-master/vulnerabilities/sqli_blind/source/high.php:26

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id' LIMIT 1;";
```

漏洞链路3：DVWA-master/vulnerabilities/sqli_blind/source/high.php:28

`SQLI`类型风险触发，由入参`$query`导致

```
$results = $sqlite_db_connection->query($query);
```

SQL注入漏洞8

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli_blind/source/low.php:12

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/vulnerabilities/sqli_blind/source/low.php:5

`$id`是污点来源

```
$id = $_GET[ 'id' ];
```

漏洞链路2: DVWA-master/vulnerabilities/sqli_blind/source/low.php:11

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
```

漏洞链路3: DVWA-master/vulnerabilities/sqli_blind/source/low.php:12

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ); // Removed 'or die' to suppress mysql errors
```

SQL注入漏洞9

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli_blind/source/low.php:26

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/sqli_blind/source/low.php:5

`$id`是污点来源

```
$id = $_GET[ 'id' ];
```

漏洞链路2: DVWA-master/vulnerabilities/sqli_blind/source/low.php:24

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = '$id';";
```

漏洞链路3: DVWA-master/vulnerabilities/sqli_blind/source/low.php:26

`SQLI`类型风险触发，由入参`$query`导致

```
$results = $sqlite_db_connection->query($query);
```

SQL注入漏洞10

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli_blind/source/medium.php:14

在线结果： [点击查看]

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/sqli_blind/source/medium.php:5

`$id`是污点来源

```
$id = $_POST[ 'id' ];
```

漏洞链路2: DVWA-master/vulnerabilities/sqli_blind/source/medium.php:10

污点从`$id`传递至`$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)) ? "" : ""));
```

漏洞链路3: DVWA-master/vulnerabilities/sqli_blind/source/medium.php:13

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = $id;";
```

漏洞链路4: DVWA-master/vulnerabilities/sqli_blind/source/medium.php:14

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($GLOBALS["___mysqli_ston"],  $query ); // Removed 'or die' to suppress mysql errors
```

SQL注入漏洞11

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/sqli_blind/source/medium.php:28

在线结果： [点击查看]

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/sqli_blind/source/medium.php:5

`$id`是污点来源

```
$id = $_POST[ 'id' ];
```

漏洞链路2: DVWA-master/vulnerabilities/sqli_blind/source/medium.php:10

污点从`$id`传递至`$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)) ? "" : ""));
```

漏洞链路3: DVWA-master/vulnerabilities/sqli_blind/source/medium.php:26

污点从`$id`传递至`$query`

```
$query  = "SELECT first_name, last_name FROM users WHERE user_id = $id;";
```

漏洞链路4: DVWA-master/vulnerabilities/sqli_blind/source/medium.php:28

`SQLI`类型风险触发，由入参`$query`导致

```
$results = $sqlite_db_connection->query($query);
```

任意命令执行漏洞（6个）

漏洞描述

在Web应用中, 有时候会用到一些命令执行的函数, 如python中的os.system, subprocess.Popen等.

当应用程序没有对用户输入的命令进行限制或者过滤不严导致用户可以执行任意命令时, 就会造成命令执行漏洞.

1. 继承WEB服务程序的权限, 执行系统命令
2. 继承WEB服务程序的权限，读写文件
3. 反弹shell
4. 控制整个网站, 甚至控制整个服务器
5. 内网渗透

修复方案

1. 尽量少用执行命令的函数.
2. 净化用户可控数据, 过滤或者转义.

参考链接

http://159.75.6.40:8083/docs#任意命令执行漏洞

任意命令执行漏洞1

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/exec/source/high.php:26

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/exec/source/high.php:5

`$target`是污点来源

```
$target = trim($_REQUEST[ 'ip' ]);
```

漏洞链路2: DVWA-master/vulnerabilities/exec/source/high.php:21

污点从`$target`传递至`$target`

```
$target = str_replace( array_keys( $substitutions ), $substitutions, $target );
```

漏洞链路3: DVWA-master/vulnerabilities/exec/source/high.php:26

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  ' . $target );
```

任意命令执行漏洞2

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/exec/source/high.php:30

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/exec/source/high.php:5

`$target`是污点来源

```
$target = trim($_REQUEST[ 'ip' ]);
```

漏洞链路2: DVWA-master/vulnerabilities/exec/source/high.php:21

污点从`$target`传递至`$target`

```
$target = str_replace( array_keys( $substitutions ), $substitutions, $target );
```

漏洞链路3: DVWA-master/vulnerabilities/exec/source/high.php:30

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  -c 4 ' . $target );
```

任意命令执行漏洞3

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/exec/source/low.php:10

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/exec/source/low.php:5

`$target`是污点来源

```
$target = $_REQUEST[ 'ip' ];
```

漏洞链路2: DVWA-master/vulnerabilities/exec/source/low.php:10

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  ' . $target );
```

任意命令执行漏洞4

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/exec/source/low.php:14

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/exec/source/low.php:5

`$target`是污点来源

```
$target = $_REQUEST[ 'ip' ];
```

漏洞链路2: DVWA-master/vulnerabilities/exec/source/low.php:14

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  -c 4 ' . $target );
```

任意命令执行漏洞5

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/exec/source/medium.php:19

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/exec/source/medium.php:5

`$target`是污点来源

```
$target = $_REQUEST[ 'ip' ];
```

漏洞链路2: DVWA-master/vulnerabilities/exec/source/medium.php:14

污点从`$target`传递至`$target`

```
$target = str_replace( array_keys( $substitutions ), $substitutions, $target );
```

漏洞链路3: DVWA-master/vulnerabilities/exec/source/medium.php:19

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  ' . $target );
```

任意命令执行漏洞6

风险等级： ● 紧急

漏洞位置： DVWA-master/vulnerabilities/exec/source/medium.php:23

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/exec/source/medium.php:5

`$target`是污点来源

```
$target = $_REQUEST[ 'ip' ];
```

漏洞链路2: DVWA-master/vulnerabilities/exec/source/medium.php:14

污点从`$target`传递至`$target`

```
$target = str_replace( array_keys( $substitutions ), $substitutions, $target );
```

漏洞链路3: DVWA-master/vulnerabilities/exec/source/medium.php:23

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  -c 4 ' . $target );
```

文件包含漏洞（2个）

漏洞描述

在通过PHP相应函数引入文件时，由于传入的文件名没有经过合理的校验，从而操作了预想之外的文件，就可能导致意外的文件泄露甚至恶意的代码注入。

要能触发该漏洞，首先有相关的引入函数(比如include，require等)，其次，用户可以控制引入函数的输入。
包括但不限于：
敏感信息泄露
获取webshell
任意命令执行

修复方案

设置白名单，代码在进行文件包含时，如果文件名可以确定，可以设置白名单对传入的参数进行比较。

过滤危险字符，由于Include/Require可以对PHP Wrapper形式的地址进行包含执行（需要配置php.ini），在Linux环境中可以通过"../../"的形式进行目录绕过，所以需要判断文件名称是否为合法的PHP文件。

设置文件目录，PHP配置文件中有open_basedir选项可以设置用户需要执行的文件目录，如果设置目录的话，PHP仅仅在该目录内搜索文件。

关闭危险配置，PHP配置中的allow_url_include选项如果打开，PHP会通过Include/Require进行远程文件包含，由于远程文件的不可信任性及不确定性，在开发中禁止打开此选项，PHP默认是关闭的。

参考链接

http://159.75.6.40:8083/docs#文件包含漏洞

文件包含漏洞1

风险等级： ● 高危
漏洞位置： DVWA-master/vulnerabilities/fi/index.php:36
在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/vulnerabilities/fi/source/impossible.php:4

污点从`GET`传递至`$file`

```
$file = $_GET[ 'page' ];
```

漏洞链路3: DVWA-master/vulnerabilities/fi/index.php:36

`LFI`类型风险触发，由入参`$file`导致

```
include( $file );
```

文件包含漏洞2

风险等级： ● 高危

漏洞位置： DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php:368

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/external/phpids/0.6/docs/examples/cakephp/ids.php:125

`$init`是污点来源

```
    $ids        = new IDS_Monitor($this->init, $_REQUEST);
```

漏洞链路2: DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php:177

污点从`$init`传递至`General`

```
    $version = isset($init->config['General']['min_php_version'])
```

漏洞链路3: DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php:206

污点从`General`传递至`pathToHTMLPurifier`

```
        $this->pathToHTMLPurifier =
```

漏洞链路4: DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php:368

`LFI`类型风险触发，由入参`pathToHTMLPurifier`导致

```
    include_once $this->pathToHTMLPurifier;
```

服务端请求伪造漏洞（1个）

漏洞描述

SSRF（Server-Side Request Forgery）服务端请求伪造是一种由攻击者构造形成由服务端发起请求的一个安全漏洞。

一般情况下，SSRF攻击的目标是从外网无法访问的内部系统。正是因为它是由服务端发起的，所以它能够请求到与它相连而与外网隔离的内部系统。

ssrf的攻击利用主要有以下几种：

恶意用户可以利用此漏洞:

1. 内网, 本地端口扫描, 获取开放端口信息
2. 主机信息收集, web应用指纹识别, 获取服务banner信息
3. 根据识别出的应用针对性的发送payload攻击, 例如struts2
4. 攻击内网和本地的应用程序及服务
5. 穿越防火墙
6. 利用file协议读取本地文件, 比如file:///etc/passwd

修复方案

1. 限制请求的端口为http常用的端口, 比如: 80,443,8080,8090
2. 黑名单内网ip, 避免应用被用来获取获取内网数据, 攻击内网
3. 禁用不需要的协议, 仅仅允许http和https请求. 可以防止类似于file:///, gopher://, ftp://协议等引起的问题

参考链接

http://159.75.6.40:8083/docs#服务端请求伪造漏洞

服务端请求伪造漏洞1

风险等级： ● 中危

漏洞位置： DVWA-master/external/phpids/0.6/lib/IDS/Filter/Storage.php:305

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/external/phpids/0.6/docs/examples/cakephp/ids.php:125

`$init`是污点来源

```
        $ids        = new IDS_Monitor($this->init, $_REQUEST);
```

漏洞链路2: DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php:177

污点从`$init`传递至`General`

```
        $version = isset($init->config['General']['min_php_version'])
```

漏洞链路3: DVWA-master/external/phpids/0.6/lib/IDS/Filter/Storage.php:100

污点从`General`传递至`source`

```
            $this->source = $init->getBasePath()
```

漏洞链路4: DVWA-master/external/phpids/0.6/lib/IDS/Filter/Storage.php:305

`SSRF`类型风险触发，由入参`source`导致

```
                $filters = json_decode(file_get_contents($this->source));
```

信息泄露（1个）

漏洞描述

phpinfo()信息泄露

phpinfo可能会泄露项目php版本和服务器变量等信息，安全起见我们需要禁用phpinfo函数

修复方案

打开php.ini，找到“disable_functions”，没有则新增，修改成以下：

disable_functions =phpinfo

disable_functions是禁用php函数，多个函数英文逗号分隔禁用：

disable_functions =函数1,函数2,函数3,phpinfo

参考链接

http://159.75.6.40:8083/docs#信息泄露

信息泄露1

风险等级： ● 低危

漏洞位置： DVWA-master/phpinfo.php:8

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/phpinfo.php:8

`INFO`类型风险触发，由入参`INFO`导致

```
phpinfo();
```

跨站脚本攻击漏洞（2个）

漏洞描述

由于web应用程序对用户的输入过滤不严产生的. 攻击者利用网站漏把恶意的脚本代码注入到网页中, 当用户浏览这些网页时, 就会执行其中的恶意代码.

1. 网络钓鱼，包括盗取各类的用户账号
2. 窃取用户cookie
3. 窃取用户浏览请回话
4. 强制弹出广告页面、刷流量

修复方案

在输出所有用户可控的数据时, 对数据做转义或者编码

参考链接

http://159.75.6.40:8083/docs#跨站脚本攻击漏洞

跨站脚本攻击漏洞1

风险等级： ● 低危

漏洞位置： DVWA-master/dvwa/includes/dvwaPage.inc.php:415

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`POST`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/vulnerabilities/sqli/session-input.php:12

污点从`POST`传递至`id`

```
$_SESSION[ 'id' ] =  $_POST[ 'id' ];
```

漏洞链路3: DVWA-master/vulnerabilities/sqli/session-input.php:14

污点从`id`传递至`body`

```
$page[ 'body' ] .= "Session ID: {$_SESSION[ 'id' ]}<br /><br /><br />";
```

漏洞链路4：DVWA-master/vulnerabilities/sqli/session-input.php:15

污点从`body`传递至`body`

```
$page[ 'body' ] .= "<script>window.opener.location.reload(true);</script>";
```

漏洞链路5：DVWA-master/vulnerabilities/sqli/session-input.php:18

污点从`body`传递至`body`

```
$page[ 'body' ] .= "
```

漏洞链路6：DVWA-master/dvwa/includes/dvwaPage.inc.php:415

`XSS`类型风险触发，由入参`body`导致

```
echo "<!DOCTYPE html>
```

跨站脚本攻击漏洞2

风险等级： ● 低危

漏洞位置： DVWA-master/dvwa/includes/dvwaPage.inc.php:415

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2：DVWA-master/vulnerabilities/view_source.php:12

污点从`GET`传递至`$security`

```
$security = $_GET[ 'security' ];
```

漏洞链路3：DVWA-master/vulnerabilities/view_source.php:71

污点从`$security`传递至`body`

```
$page[ 'body' ] .= "
```

漏洞链路4：DVWA-master/dvwa/includes/dvwaPage.inc.php:415

`XSS`类型风险触发，由入参`body`导致

```
echo "<!DOCTYPE html>
```

目录穿越漏洞（2个）

漏洞描述

修复方案

数据过滤，对网站用户提交过来的文件名进行硬编码或者统一编码，对文件后缀进行白名单控制，对包含了恶意的符号或者空字节进行拒绝

参考链接

http://159.75.6.40:8083/docs#目录穿越漏洞

目录穿越漏洞1

风险等级： ● 低危

漏洞位置： DVWA-master/vulnerabilities/upload/source/impossible.php:40

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/vulnerabilities/upload/source/impossible.php:9

`$uploaded_name`是污点来源

```
$uploaded_name = $_FILES[ 'uploaded' ][ 'name' ];
```

漏洞链路2：DVWA-master/vulnerabilities/upload/source/impossible.php:10

污点从`$uploaded_name`传递至`$uploaded_ext`

```
$uploaded_ext  = substr( $uploaded_name, strrpos( $uploaded_name, '.' ) + 1);
```

漏洞链路3：DVWA-master/vulnerabilities/upload/source/impossible.php:20

污点从`$uploaded_ext`传递至`$temp_file`

```
$temp_file    .= DIRECTORY_SEPARATOR . md5( uniqid() . $uploaded_name ) . '.' . $uploaded_ext;
```

漏洞链路4：DVWA-master/vulnerabilities/upload/source/impossible.php:40

`PATH`类型风险触发，由入参`$temp_file`导致

```
if( rename( $temp_file, ( getcwd() . DIRECTORY_SEPARATOR . $target_path . $target_file ) ) ) {
```

目录穿越漏洞2

风险等级： ● 低危

漏洞位置： DVWA-master/vulnerabilities/upload/source/impossible.php:40

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/upload/source/impossible.php:9

`$uploaded_name`是污点来源

```
$uploaded_name = $_FILES[ 'uploaded' ][ 'name' ];
```

漏洞链路2: DVWA-master/vulnerabilities/upload/source/impossible.php:10

污点从`$uploaded_name`传递至`$uploaded_ext`

```
$uploaded_ext  = substr( $uploaded_name, strrpos( $uploaded_name, '.' ) + 1);
```

漏洞链路3: DVWA-master/vulnerabilities/upload/source/impossible.php:18

污点从`$uploaded_ext`传递至`$target_file`

```
$target_file   =  md5( uniqid() . $uploaded_name ) . '.' . $uploaded_ext;
```

漏洞链路4: DVWA-master/vulnerabilities/upload/source/impossible.php:40

`PATH`类型风险触发，由入参`$target_file`导致

```
if( rename( $temp_file, ( getcwd() . DIRECTORY_SEPARATOR . $target_path . $target_file ) ) ) {
```

提示性风险（可忽略）（26个）

漏洞描述

提示性漏洞仅供参考，可以忽略该类漏洞。

SQL注入漏洞1

风险等级： ● 提示

漏洞位置： DVWA-master/login.php:40

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`POST`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/login.php:20

污点从`POST`传递至`$user`

```
$user = $_POST[ 'username' ];
```

漏洞链路3: DVWA-master/login.php:21

污点从`$user`传递至`$user`

```
$user = stripslashes( $user );
```

漏洞链路4: DVWA-master/login.php:22

污点从`$user`传递至`$user`

```
$user = ((isset($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)) ? "" : ""));
```

漏洞链路5: DVWA-master/login.php:39

污点从`$user`传递至`$query`

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

漏洞链路6: DVWA-master/login.php:40

`SQLI`类型风险触发，由入参`$query`导致

```
$result = @mysqli_query($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)) . '.<br />Try <a href="setup.php">installing again</a>.</pre>' );
```

SQL注入漏洞2

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/brute/source/high.php:20

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/brute/source/high.php:8

`$user`是污点来源

```
$user = $_GET[ 'username' ];
```

漏洞链路2: DVWA-master/vulnerabilities/brute/source/high.php:9

污点从`$user`传递至`$user`

```
$user = stripslashes( $user );
```

漏洞链路3: DVWA-master/vulnerabilities/brute/source/high.php:10

污点从`$user`传递至`$user`

```
$user = ((isset($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)) ? "" : ""));
```

漏洞链路4: DVWA-master/vulnerabilities/brute/source/high.php:19

污点从`$user`传递至`$query`

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

漏洞链路5: DVWA-master/vulnerabilities/brute/source/high.php:20

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($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>' );
```

SQL注入漏洞3

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/brute/source/medium.php:15

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/vulnerabilities/brute/source/medium.php:5

`$user`是污点来源

```
$user = $_GET[ 'username' ];
```

漏洞链路2：DVWA-master/vulnerabilities/brute/source/medium.php:6

污点从`$user`传递至`$user`

```
$user = ((isset($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)) ? "" : ""));
```

漏洞链路3：DVWA-master/vulnerabilities/brute/source/medium.php:14

污点从`$user`传递至`$query`

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

漏洞链路4：DVWA-master/vulnerabilities/brute/source/medium.php:15

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($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>' );
```

SQL注入漏洞4

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/csrf/test_credentials.php:22

在线结果： [点击查看]

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`POST`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/vulnerabilities/csrf/test_credentials.php:12

污点从`POST`传递至`$user`

```
$user = $_POST[ 'username' ];
```

漏洞链路3: DVWA-master/vulnerabilities/csrf/test_credentials.php:13

污点从`$user`传递至`$user`

```
$user = stripslashes( $user );
```

漏洞链路4: DVWA-master/vulnerabilities/csrf/test_credentials.php:14

污点从`$user`传递至`$user`

```
$user = mysqli_real_escape_string($GLOBALS["___mysqli_ston"], $user);
```

漏洞链路5: DVWA-master/vulnerabilities/csrf/test_credentials.php:21

污点从`$user`传递至`$query`

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

漏洞链路6: DVWA-master/vulnerabilities/csrf/test_credentials.php:22

`SQLI`类型风险触发，由入参`$query`导致

```
$result = @mysqli_query($GLOBALS["___mysqli_ston"], $query) or die( '<pre>'.  mysqli_connect_error(
) . '.<br />Try <a href="setup.php">installing again</a>.</pre>' );
```

跨站脚本攻击漏洞5

风险等级： ● 提示

漏洞位置： DVWA-master/dvwa/includes/dvwaPage.inc.php:415

在线结果： [点击查看]

污染传递链路

漏洞链路1：DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`POST`是污点来源

```
$request = array(
```

漏洞链路2：DVWA-master/vulnerabilities/csrf/test_credentials.php:12

污点从`POST`传递至`$user`

```
$user = $_POST[ 'username' ];
```

漏洞链路3：DVWA-master/vulnerabilities/csrf/test_credentials.php:13

污点从`$user`传递至`$user`

```
$user = stripslashes( $user );
```

漏洞链路4：DVWA-master/vulnerabilities/csrf/test_credentials.php:14

污点从`$user`传递至`$user`

```
$user = mysqli_real_escape_string($GLOBALS["___mysqli_ston"], $user);
```

漏洞链路5：DVWA-master/vulnerabilities/csrf/test_credentials.php:24

污点从`$user`传递至`$login_state`

```
$login_state = "<h3 class=\"loginSuccess\">Valid password for '{$user}'</h3>";
```

漏洞链路6：DVWA-master/vulnerabilities/csrf/test_credentials.php:35

污点从`$login_state`传递至`body`

```
$page[ 'body' ] .= "
```

漏洞链路7：DVWA-master/dvwa/includes/dvwaPage.inc.php:415

`XSS`类型风险触发，由入参`body`导致

```
echo "<!DOCTYPE html>
```

任意命令执行漏洞6

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/exec/source/impossible.php:22

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`REQUEST`是污点来源

```
$request = array(
```

漏洞链路2：DVWA-master/vulnerabilities/exec/source/impossible.php:8

污点从`REQUEST`传递至`$target`

```
$target = $_REQUEST[ 'ip' ];
```

漏洞链路3：DVWA-master/vulnerabilities/exec/source/impossible.php:9

污点从`$target`传递至`$target`

```
$target = stripslashes( $target );
```

漏洞链路4：DVWA-master/vulnerabilities/exec/source/impossible.php:12

污点从`$target`传递至`$octet`

```
$octet = explode( ".", $target );
```

漏洞链路5：DVWA-master/vulnerabilities/exec/source/impossible.php:15

污点从`$octet`传递至`3`

```
if( ( is_numeric( $octet[0] ) ) && ( is_numeric( $octet[1] ) ) && ( is_numeric( $octet[2] ) ) && ( is_numeric( $octet[3] ) ) && ( sizeof( $octet ) == 4 ) ) {
```

漏洞链路6：DVWA-master/vulnerabilities/exec/source/impossible.php:17

污点从`3`传递至`$target`

```
$target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3];
```

漏洞链路7：DVWA-master/vulnerabilities/exec/source/impossible.php:22

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  ' . $target );
```

任意命令执行漏洞7

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/exec/source/impossible.php:26

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`REQUEST`是污点来源

```
$request = array(
```

漏洞链路2：DVWA-master/vulnerabilities/exec/source/impossible.php:8

污点从`REQUEST`传递至`$target`

```
$target = $_REQUEST[ 'ip' ];
```

漏洞链路3：DVWA-master/vulnerabilities/exec/source/impossible.php:9

污点从`$target`传递至`$target`

```
$target = stripslashes( $target );
```

漏洞链路4：DVWA-master/vulnerabilities/exec/source/impossible.php:12

污点从`$target`传递至`$octet`

```
$octet = explode( ".", $target );
```

漏洞链路5：DVWA-master/vulnerabilities/exec/source/impossible.php:15

污点从`$octet`传递至`3`

```
if( ( is_numeric( $octet[0] ) ) && ( is_numeric( $octet[1] ) ) && ( is_numeric( $octet[2] ) ) && ( is_numeric( $octet[3] ) ) && ( sizeof( $octet ) == 4 ) ) {
```

漏洞链路6：DVWA-master/vulnerabilities/exec/source/impossible.php:17

污点从`3`传递至`$target`

```
$target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3];
```

漏洞链路7：DVWA-master/vulnerabilities/exec/source/impossible.php:26

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  -c 4 ' . $target );
```

任意命令执行漏洞8

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/exec/source/impossible.php:22

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/vulnerabilities/exec/source/impossible.php:8

`$target`是污点来源

```
$target = $_REQUEST[ 'ip' ];
```

漏洞链路2：DVWA-master/vulnerabilities/exec/source/impossible.php:9

污点从`$target`传递至`$target`

```
$target = stripslashes( $target );
```

漏洞链路3：DVWA-master/vulnerabilities/exec/source/impossible.php:12

污点从`$target`传递至`$octet`

```
$octet = explode( ".", $target );
```

漏洞链路4：DVWA-master/vulnerabilities/exec/source/impossible.php:15

污点从`$octet`传递至`3`

```
if( ( is_numeric( $octet[0] ) ) && ( is_numeric( $octet[1] ) ) && ( is_numeric( $octet[2] ) ) && ( is_numeric( $octet[3] ) ) && ( sizeof( $octet ) == 4 ) ) {
```

漏洞链路5：DVWA-master/vulnerabilities/exec/source/impossible.php:17

污点从`3`传递至`$target`

```
$target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3];
```

漏洞链路6：DVWA-master/vulnerabilities/exec/source/impossible.php:22

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  ' . $target );
```

任意命令执行漏洞9

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/exec/source/impossible.php:26

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/vulnerabilities/exec/source/impossible.php:8

`$target`是污点来源

```
$target = $_REQUEST[ 'ip' ];
```

漏洞链路2: DVWA-master/vulnerabilities/exec/source/impossible.php:9

污点从`$target`传递至`$target`

```
$target = stripslashes( $target );
```

漏洞链路3: DVWA-master/vulnerabilities/exec/source/impossible.php:12

污点从`$target`传递至`$octet`

```
$octet = explode( ".", $target );
```

漏洞链路4: DVWA-master/vulnerabilities/exec/source/impossible.php:15

污点从`$octet`传递至`3`

```
if( ( is_numeric( $octet[0] ) ) && ( is_numeric( $octet[1] ) ) && ( is_numeric( $octet[2] ) ) && ( is_numeric( $octet[3] ) ) && ( sizeof( $octet ) == 4 ) ) {
```

漏洞链路5: DVWA-master/vulnerabilities/exec/source/impossible.php:17

污点从`3`传递至`$target`

```
$target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3];
```

漏洞链路6: DVWA-master/vulnerabilities/exec/source/impossible.php:26

`RCE`类型风险触发，由入参`$target`导致

```
$cmd = shell_exec( 'ping  -c 4 ' . $target );
```

回车换行注入漏洞10

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/sqli_blind/cookie-input.php:12

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`POST`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/vulnerabilities/sqli_blind/cookie-input.php:12

`CRLF`类型风险触发，由入参`POST`导致

```
setcookie( 'id', $_POST[ 'id' ]);
```

任意文件删除漏洞11

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/upload/source/impossible.php:51

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/upload/source/impossible.php:9

`$uploaded_name`是污点来源

```
$uploaded_name = $_FILES[ 'uploaded' ][ 'name' ];
```

漏洞链路2: DVWA-master/vulnerabilities/upload/source/impossible.php:10

污点从`$uploaded_name`传递至`$uploaded_ext`

```
$uploaded_ext  = substr( $uploaded_name, strrpos( $uploaded_name, '.' ) + 1);
```

漏洞链路3: DVWA-master/vulnerabilities/upload/source/impossible.php:20

污点从`$uploaded_ext`传递至`$temp_file`

```
$temp_file    .= DIRECTORY_SEPARATOR . md5( uniqid() . $uploaded_name ) . '.' . $uploaded_ext;
```

漏洞链路4: DVWA-master/vulnerabilities/upload/source/impossible.php:51

`DEL`类型风险触发，由入参`$temp_file`导致

```
unlink( $temp_file );
```

任意文件上传漏洞12

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/upload/source/high.php:20

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/upload/source/high.php:6

`$target_path`是污点来源

```
$target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );
```

漏洞链路2: DVWA-master/vulnerabilities/upload/source/high.php:20

`WRITE`类型风险触发，由入参`$target_path`导致

```
if( !move_uploaded_file( $uploaded_tmp, $target_path ) ) {
```

任意文件上传漏洞13

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/upload/source/low.php:9

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/upload/source/low.php:6

`$target_path`是污点来源

```
$target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );
```

漏洞链路2: DVWA-master/vulnerabilities/upload/source/low.php:9

`WRITE`类型风险触发，由入参`$target_path`导致

```
if( !move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ], $target_path ) ) {
```

任意文件上传漏洞14

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/upload/source/medium.php:18

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/upload/source/medium.php:6

`$target_path`是污点来源

```
$target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );
```

漏洞链路2: DVWA-master/vulnerabilities/upload/source/medium.php:18

`WRITE`类型风险触发，由入参`$target_path`导致

```
if( !move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ], $target_path ) ) {
```

任意文件读取/下载漏洞15

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/view_help.php:17

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/vulnerabilities/view_help.php:11

污点从`GET`传递至`$id`

```
$id        = $_GET[ 'id' ];
```

漏洞链路3: DVWA-master/vulnerabilities/view_help.php:17

`READ`类型风险触发，由入参`$id`导致

```
eval( '?>' . file_get_contents( DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/{$id}/help/help.php" ) . '<?php ' );
```

任意文件读取/下载漏洞16

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/view_help.php:19

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2：DVWA-master/vulnerabilities/view_help.php:13

污点从`GET`传递至`$locale`

```
$locale = $_GET[ 'locale' ];
```

漏洞链路3：DVWA-master/vulnerabilities/view_help.php:19

`READ`类型风险触发，由入参`$locale`导致

```
eval( '?>' . file_get_contents( DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/{$id}/help/help.{$locale}.php" ) . '<?php ' );
```

任意文件读取/下载漏洞17

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/view_source.php:53

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2：DVWA-master/vulnerabilities/view_source.php:12

污点从`GET`传递至`$security`

```
$security = $_GET[ 'security' ];
```

漏洞链路3：DVWA-master/vulnerabilities/view_source.php:53

`READ`类型风险触发，由入参`$security`导致

```
$source = @file_get_contents( DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/{$id}/source/{$security}.php" );
```

任意文件读取/下载漏洞18

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/view_source.php:58

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2：DVWA-master/vulnerabilities/view_source.php:12

污点从`GET`传递至`$security`

```
$security = $_GET[ 'security' ];
```

漏洞链路3：DVWA-master/vulnerabilities/view_source.php:58

`READ`类型风险触发，由入参`$security`导致

```
$js_source = @file_get_contents( DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/{$id}/source/{$security}.js" );
```

任意文件读取/下载漏洞19

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/view_source_all.php:13

在线结果： 点击查看

污染传递链路

漏洞链路1：DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2：DVWA-master/vulnerabilities/view_source_all.php:11

污点从`GET`传递至`$id`

```
$id = $_GET[ 'id' ];
```

漏洞链路3：DVWA-master/vulnerabilities/view_source_all.php:13

`READ`类型风险触发，由入参`$id`导致

```
$lowsrc = @file_get_contents("./{$id}/source/low.php");
```

任意文件读取/下载漏洞20

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/view_source_all.php:17

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/vulnerabilities/view_source_all.php:11

污点从`GET`传递至`$id`

```
$id = $_GET[ 'id' ];
```

漏洞链路3: DVWA-master/vulnerabilities/view_source_all.php:17

`READ`类型风险触发，由入参`$id`导致

```
$medsrc = @file_get_contents("./{$id}/source/medium.php");
```

任意文件读取/下载漏洞21

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/view_source_all.php:21

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/vulnerabilities/view_source_all.php:11

污点从`GET`传递至`$id`

```
$id = $_GET[ 'id' ];
```

漏洞链路3: DVWA-master/vulnerabilities/view_source_all.php:21

`READ`类型风险触发，由入参`$id`导致

```
$highsrc = @file_get_contents("./{$id}/source/high.php");
```

任意文件读取/下载漏洞22

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/view_source_all.php:25

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/dvwa/includes/dvwaPhpIds.inc.php:61

`GET`是污点来源

```
$request = array(
```

漏洞链路2: DVWA-master/vulnerabilities/view_source_all.php:11

污点从`GET`传递至`$id`

```
$id = $_GET[ 'id' ];
```

漏洞链路3: DVWA-master/vulnerabilities/view_source_all.php:25

`READ`类型风险触发，由入参`$id`导致

```
$impsrc = @file_get_contents("./{$id}/source/impossible.php");
```

SQL注入漏洞23

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/xss_s/source/high.php:19

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/xss_s/source/high.php:6

`$name`是污点来源

```
$name    = trim( $_POST[ 'txtName' ] );
```

漏洞链路2: DVWA-master/vulnerabilities/xss_s/source/high.php:14

污点从`$name`传递至`$name`

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

漏洞链路3: DVWA-master/vulnerabilities/xss_s/source/high.php:15

污点从`$name`传递至`$name`

```
$name = ((isset($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)) ? "" : ""));
```

漏洞链路4: DVWA-master/vulnerabilities/xss_s/source/high.php:18

污点从`$name`传递至`$query`

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

漏洞链路5: DVWA-master/vulnerabilities/xss_s/source/high.php:19

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($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>' );
```

SQL注入漏洞24

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/xss_s/source/low.php:17

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/xss_s/source/low.php:6

`$name`是污点来源

```
$name    = trim( $_POST[ 'txtName' ] );
```

漏洞链路2: DVWA-master/vulnerabilities/xss_s/source/low.php:13

污点从`$name`传递至`$name`

```
$name = ((isset($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)) ? "" : ""));
```

漏洞链路3: DVWA-master/vulnerabilities/xss_s/source/low.php:16

污点从`$name`传递至`$query`

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

漏洞链路4: DVWA-master/vulnerabilities/xss_s/source/low.php:17

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($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>' );
```

SQL注入漏洞25

风险等级： ● 提示

漏洞位置： DVWA-master/vulnerabilities/xss_s/source/medium.php:19

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/vulnerabilities/xss_s/source/medium.php:6

`$name`是污点来源

```
$name    = trim( $_POST[ 'txtName' ] );
```

漏洞链路2: DVWA-master/vulnerabilities/xss_s/source/medium.php:14

污点从`$name`传递至`$name`

```
$name = str_replace( '<script>', '', $name );
```

漏洞链路3: DVWA-master/vulnerabilities/xss_s/source/medium.php:15

污点从`$name`传递至`$name`

```
$name = ((isset($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)) ? "" : ""));
```

漏洞链路4: DVWA-master/vulnerabilities/xss_s/source/medium.php:18

污点从`$name`传递至`$query`

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

漏洞链路5: DVWA-master/vulnerabilities/xss_s/source/medium.php:19

`SQLI`类型风险触发，由入参`$query`导致

```
$result = mysqli_query($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>' );
```

文件包含漏洞26

风险等级： ● 提示

漏洞位置： DVWA-master/external/phpids/0.6/lib/IDS/Caching/Factory.php:76

在线结果： 点击查看

污染传递链路

漏洞链路1: DVWA-master/external/phpids/0.6/docs/examples/cakephp/ids.php:125

`$init`是污点来源

```
        $ids        = new IDS_Monitor($this->init, $_REQUEST);
```

漏洞链路2: DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php:177

污点从`$init`传递至`config`

```
        $version = isset($init->config['General']['min_php_version'])
```

漏洞链路3: DVWA-master/external/phpids/0.6/lib/IDS/Filter/Storage.php:96

污点从`config`传递至`caching`

```
            $caching = isset($init->config['Caching']['caching']) ?
```

漏洞链路4: DVWA-master/external/phpids/0.6/lib/IDS/Caching/Factory.php:66

污点从`caching`传递至`$wrapper`

```
        $wrapper = preg_replace(
```

漏洞链路5: DVWA-master/external/phpids/0.6/lib/IDS/Caching/Factory.php:72

污点从`$wrapper`传递至`$path`

```
        $path     = dirname(__FILE__) . DIRECTORY_SEPARATOR .
```

漏洞链路6: DVWA-master/external/phpids/0.6/lib/IDS/Caching/Factory.php:76

`LFI`类型风险触发，由入参`$path`导致

```
        include_once $path;
```