Security Analysis Report

DVWA 1.19

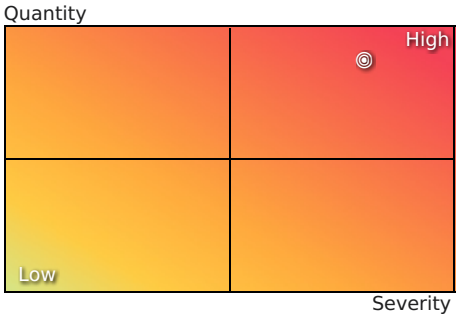
Date: 2017-08-31
User: johannes

1. Executive Summary

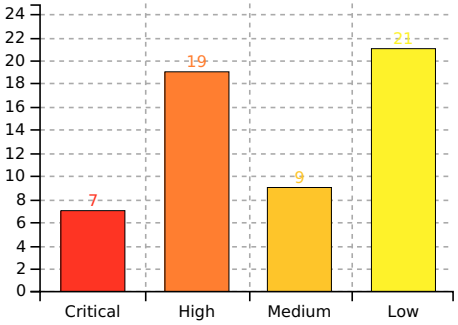
Project Name:	DVWA 1.19
Analysis Start Date:	2017-08-04, 15:01
Analysis End Date:	2017-08-04, 15:04
Analysis Time:	2m 48s

Analyzed Files:	126
Analyzed LOC:	12,555
Analyzed Issue Types:	108
Detected Issues:	56

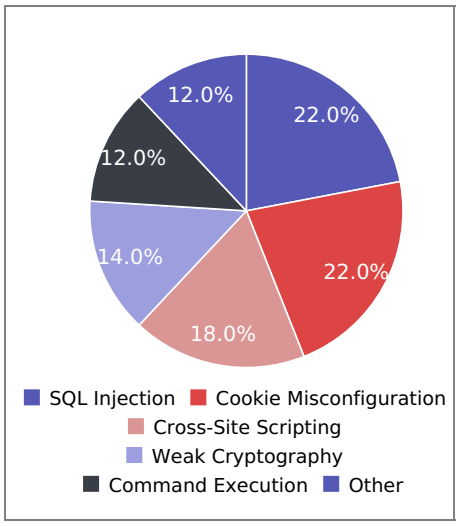
Risk Matrix



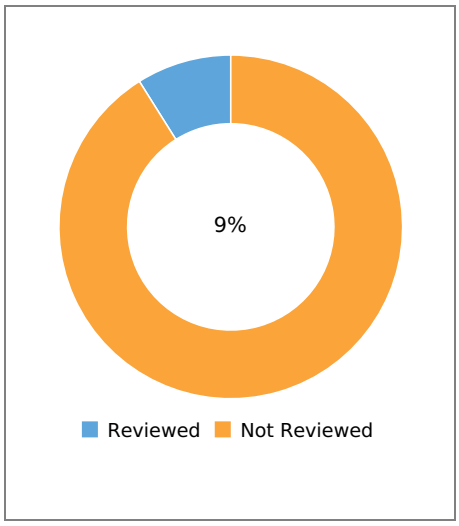
Vulnerabilities by Risk



Top Vulnerability Types



Review Status



2. Issue Breakdown

The detected security issues in this project are categorized as follows.

Severity	Vulnerability Type	CWE [?]	OWASP Top 10 [?]	SANS 25 [?]	PCI DSS [?]	Issues
Critical	Command Execution	78	A1	Rank 2	6.5.1	6
High	File Inclusion	98	A4	Rank 13	6.5.8	1
High	SQL Injection	89	A1	Rank 1	6.5.1	11
High	File Upload	434	A5	Rank 9	6.5.8	6
High	PHP Object Injection	502	A4	Rank 18	6.5.1	1
Medium	Cross-Site Scripting	79	A3	Rank 4	6.5.7	9
Medium	CVE		A9	Not Ranked	6.2	1
Low	Session Fixation	384	A2	Not Ranked	6.5.10	1
Low	Weak Cryptography	310	A6	Not Ranked	6.5.3	7
Low	Cookie Misconfiguration	494	A5	Not Ranked	6.5.10	11
Low	Information Leakage	209	A6	Not Ranked	6.5.5	2

3. Issue Details

In the following, all security issues detected in the analyzed project are presented in detail. The issues are grouped by vulnerability type and by the detected markup context. A *markup context* represents the position of user-supplied data (*source*) used in a sensitive operation (*sink*). Depending on the markup context, an attacker can alter the operation and different security mechanisms must be applied in order to patch the security issue thoroughly.

3.1. Command Execution

CWE: 78 **OWASP Top 10:** A1 **SANS 25 Rank:** 2 **PCI DSS:** 6.5.1 **Severity:** Critical

A command execution vulnerability occurs when user input is embedded unsanitized into a system command. It allows an attacker to terminate the intended command and to append arbitrary system commands for execution. Common character sequences to separate commands are semicolons (;), boolean operators (&|), $(subcommands), `backticks`, and newline characters (%0a). This can be prevented by using the built-in function escapeshellcmd() on the whole command or by escaping all command arguments with the built-in function escapeshellarg(). It is recommended to avoid invoking system commands from a web application whenever possible.

Issue #375377 - DVWA-master/vulnerabilities/exec/source/medium.php: 19

Path: DVWA-master/vulnerabilities/exec/source/medium.php
Line: 19
Sink: shell_exec
Source: _REQUEST
Taint: HTTP
Status: Suspicious

Code Summary

The GET parameter "ip" is received in line 5 of DVWA-master/vulnerabilities/exec/source/medium.php.
It is concatenated into command markup in line 19 of DVWA-master/vulnerabilities/exec/source/medium.php.
The user-supplied data is then used unsanitized in the sensitive operation shell_exec() in line 19 of DVWA-master/vulnerabilities/exec/source/medium.php.

DVWA-master/vulnerabilities/exec/source/medium.php

```
5  // Get input
   $target = $_REQUEST['ip'];
   ⋮
8  // Set blacklist
   $substitutions = array('&&' => '', ';' => '');
   ⋮
14 // Remove any of the charactars in the array (blacklist).
   $target = str_replace(array_keys($substitutions), $substitutions, $target);
   ⋮
19 shell_exec('ping  ' . $target);
```

Command Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
ping $_REQUEST['ip']
```

Issue #375378 - DVWA-master/vulnerabilities/exec/source/medium.php: 23

Path: DVWA-master/vulnerabilities/exec/source/medium.php
Line: 23
Sink: shell_exec
Source: _REQUEST
Taint: HTTP
Status: Suspicious

Code Summary

The GET parameter "ip" is received in line 5 of DVWA-master/vulnerabilities/exec/source/medium.php.
It is concatenated into command markup in line 23 of DVWA-master/vulnerabilities/exec/source/medium.php.
The user-supplied data is then used unsanitized in the sensitive operation shell_exec() in line 23 of DVWA-master/vulnerabilities/exec/source/medium.php.

DVWA-master/vulnerabilities/exec/source/medium.php

```
5  // Get input
   $target = $_REQUEST['ip'];
   ⋮
8  // Set blacklist
```

```
    $substitutions = array('&&' => '', ';' => '');
14  // Remove any of the charactars in the array (blacklist).
    $target = str_replace(array_keys($substitutions), $substitutions, $target);
    
23  shell_exec('ping  -c 4 ' . $target);
```

Command Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
ping -c 4 $_REQUEST['ip']
```

Issue #375379 - DVWA-master/vulnerabilities/exec/source/low.php: 10

Path: DVWA-master/vulnerabilities/exec/source/low.php
Line: 10
Sink: shell_exec
Source: _REQUEST
Taint: HTTP
Status: Suspicious

Code Summary

The GET parameter "ip" is received in line 5 of DVWA-master/vulnerabilities/exec/source/low.php.
It is concatenated into command markup in line 10 of DVWA-master/vulnerabilities/exec/source/low.php.
The user-supplied data is then used unsanitized in the sensitive operation shell_exec() in line 10 of DVWA-master/vulnerabilities/exec/source/low.php.

DVWA-master/vulnerabilities/exec/source/low.php

```
5   // Get input
    $target = $_REQUEST['ip'];
    
10  shell_exec('ping  ' . $target);
```

Command Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
ping $_REQUEST['ip']
```

Issue #375380 - DVWA-master/vulnerabilities/exec/source/low.php: 14

Path: DVWA-master/vulnerabilities/exec/source/low.php
Line: 14
Sink: shell_exec
Source: _REQUEST
Taint: HTTP
Status: Suspicious

Code Summary

The GET parameter "ip" is received in line 5 of DVWA-master/vulnerabilities/exec/source/low.php.
It is concatenated into command markup in line 14 of DVWA-master/vulnerabilities/exec/source/low.php.
The user-supplied data is then used unsanitized in the sensitive operation shell_exec() in line 14 of DVWA-master/vulnerabilities/exec/source/low.php.

DVWA-master/vulnerabilities/exec/source/low.php

```
5   // Get input
    $target = $_REQUEST['ip'];
    
14  shell_exec('ping  -c 4 ' . $target);
```

Command Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
ping -c 4 $_REQUEST['ip']
```

Issue #375381 - DVWA-master/vulnerabilities/exec/source/high.php: 26

Path: DVWA-master/vulnerabilities/exec/source/high.php
Line: 26
Sink: shell_exec
Source: _REQUEST
Taint: HTTP

Code Summary

The GET parameter "ip" is received in line 5 of DVWA-master/vulnerabilities/exec/source/high.php.
It is concatenated into command markup in line 26 of DVWA-master/vulnerabilities/exec/source/high.php.
The user-supplied data is then used unsanitized in the sensitive operation shell_exec() in line 26 of DVWA-master/vulnerabilities/exec/source/high.php.

DVWA-master/vulnerabilities/exec/source/high.php

```
5   // Get input
    $target = trim($_REQUEST['ip']);
    ⋮
    // Set blacklist
8   $substitutions = array('&' => '', ';' => '', '| ' => '', '-' => '', '$' => '', '(' => '', ')' => '', '`' => '', '||' =>
    '');
    ⋮
    // Remove any of the charactars in the array (blacklist).
21  $target = str_replace(array_keys($substitutions), $substitutions, $target);
    ⋮
26  shell_exec('ping  ' . $target);
```

Command Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
ping $_REQUEST['ip']
```

Issue #375382 - DVWA-master/vulnerabilities/exec/source/high.php: 30

Path: DVWA-master/vulnerabilities/exec/source/high.php
Line: 30
Sink: shell_exec
Source: _REQUEST
Taint: HTTP

Code Summary

The GET parameter "ip" is received in line 5 of DVWA-master/vulnerabilities/exec/source/high.php.
It is concatenated into command markup in line 30 of DVWA-master/vulnerabilities/exec/source/high.php.
The user-supplied data is then used unsanitized in the sensitive operation shell_exec() in line 30 of DVWA-master/vulnerabilities/exec/source/high.php.

DVWA-master/vulnerabilities/exec/source/high.php

```
5   // Get input
    $target = trim($_REQUEST['ip']);
    ⋮
    // Set blacklist
8   $substitutions = array('&' => '', ';' => '', '| ' => '', '-' => '', '$' => '', '(' => '', ')' => '', '`' => '', '||' =>
    '');
    ⋮
    // Remove any of the charactars in the array (blacklist).
21  $target = str_replace(array_keys($substitutions), $substitutions, $target);
    ⋮
30  shell_exec('ping  -c 4 ' . $target);
```

Command Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
ping -c 4 $_REQUEST['ip']
```

3.2. File Inclusion

CWE: 98 **OWASP Top 10:** A4 **SANS 25 Rank:** 13 **PCI DSS:** 6.5.8 **Severity:** High

A file inclusion vulnerability occurs when user input is embedded unsanitized into a file path used for inclusion. It allows an attacker to tamper the file path that is used for the inclusion.

3.2.1. Local File Inclusion

CWE: 97 **OWASP Top 10:** A4 **SANS 25 Rank:** 13 **PCI DSS:** 6.5.8 **Severity:** High

This vulnerability is categorized as a local file inclusion (LFI) vulnerability because a path name is prefixed to the injection point. An attacker can use path traversal character sequences (../) to access and include arbitrary files from the file system as PHP code. This can lead to the disclosure of sensitive files or to the execution of PHP code that was placed by the attacker on the file system, for example by injecting a payload into a log file. In order to prevent path traversal, the built-in function basename() can be used

to limit the user input to a file name and to ignore injected path names. It is recommended to create a whitelist of all allowed file names.

Issue #375376 - DVWA-master/vulnerabilities/fi/index.php: 36

Path: DVWA-master/vulnerabilities/fi/index.php
Line: 36
Sink: include
Source: _GET
Taint: HTTP

Code Summary

The GET parameter "page" is received in line 4 of DVWA-master/vulnerabilities/fi/source/medium.php.
The user-supplied data is then used unsanitized in the sensitive operation include() in line 36 of DVWA-master/vulnerabilities/fi/index.php.

DVWA-master/vulnerabilities/fi/index.php

```
32 require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/fi/source/{$vulnerabilityFile}";
⋮  ⋮
36 include $file;
```

DVWA-master/vulnerabilities/fi/source/medium.php

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

Path Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
$_GET['page']
```

3.3. SQL Injection

CWE: 89 **OWASP Top 10:** A1 **SANS 25 Rank:** 1 **PCI DSS:** 6.5.1 **Severity:** High

A sql injection vulnerability occurs when user input is embedded unsanitized into a SQL query. An attacker can modify the SQL syntax and alter the query's target or result. This can lead to the retrieval of sensitive information from the database or to an attack against the underlying web server by using SQL file operations. An attacker can also elevate privileges if the SQL query is used for authentication. It is recommended to use prepared statements and to run the database user with the least privileges necessary.

3.3.1. SQL Injection (single-quoted)

CWE: 89 **OWASP Top 10:** A1 **SANS 25 Rank:** 1 **PCI DSS:** 6.5.1 **Severity:** High

The detected injection point in the SQL query occurs within single quotes. Thus, the user input can be sanitized by using the built-in function addslashes() that escapes the data and prevents breaking out of the quotes.

Issue #375367 - DVWA-master/vulnerabilities/sqli/source/low.php: 9

Path: DVWA-master/vulnerabilities/sqli/source/low.php
Line: 9
Sink: mysqli_query
Source: _REQUEST
Taint: HTTP

Code Summary

The GET parameter "id" is received in line 5 of DVWA-master/vulnerabilities/sqli/source/low.php.
It is concatenated into sql markup in line 8 of DVWA-master/vulnerabilities/sqli/source/low.php.
The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 9 of DVWA-master/vulnerabilities/sqli/source/low.php.

DVWA-master/vulnerabilities/sqli/source/low.php

```
5 // Get input
  $id = $_REQUEST['id'];
⋮  ⋮
8 // Check database
  $query = "SELECT first_name, last_name FROM users WHERE user_id = '{$id}';";
9 mysqli_query($GLOBALS["___mysqli_ston"], $query);
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT first_name, last_name FROM users WHERE user_id = '$_REQUEST['id']';
```

Issue #375369 - DVWA-master/vulnerabilities/sqli/source/medium.php: 10

Path: DVWA-master/vulnerabilities/sqli/source/medium.php
Line: 10
Sink: mysqli_query
Source: _REQUEST
Taint: HTTP

Code Summary

The GET parameter "id" is received in line 5 of DVWA-master/vulnerabilities/sqli/source/low.php.

The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 10 of DVWA-master/vulnerabilities/sqli/source/medium.php.

DVWA-master/vulnerabilities/sqli/index.php

```
34 require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/sqli/source/{$vulnerabilityFile}";
```

DVWA-master/vulnerabilities/sqli/source/medium.php

```
10 mysqli_query($GLOBALS["___mysqli_ston"], $query);
```

DVWA-master/vulnerabilities/sqli/source/low.php

```
5  // Get input
   $id = $_REQUEST['id'];
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT first_name, last_name FROM users WHERE user_id = '$_REQUEST['id']';
```

Issue #375371 - DVWA-master/vulnerabilities/sqli_blind/source/low.php: 9

Path: DVWA-master/vulnerabilities/sqli_blind/source/low.php
Line: 9
Sink: mysqli_query
Source: _GET
Taint: HTTP

Code Summary

The GET parameter "id" is received in line 5 of DVWA-master/vulnerabilities/sqli_blind/source/low.php.

It is concatenated into sql markup in line 8 of DVWA-master/vulnerabilities/sqli_blind/source/low.php.

The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 9 of DVWA-master/vulnerabilities/sqli_blind/source/low.php.

DVWA-master/vulnerabilities/sqli_blind/source/low.php

```
5  // Get input
   $id = $_GET['id'];
⋮  ⋮
8  // Check database
   $getid = "SELECT first_name, last_name FROM users WHERE user_id = '{$id}';";
9  mysqli_query($GLOBALS["___mysqli_ston"], $getid);
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT first_name, last_name FROM users WHERE user_id = '$_GET['id']';
```

Issue #375372 - DVWA-master/vulnerabilities/sqli_blind/source/high.php: 9

Path: DVWA-master/vulnerabilities/sqli_blind/source/high.php
Line: 9
Sink: mysqli_query
Source: _COOKIE
Taint: HTTP

Code Summary

The cookie "id" is received in line 5 of DVWA-master/vulnerabilities/sqli_blind/source/high.php.

It is concatenated into sql markup in line 8 of DVWA-master/vulnerabilities/sqli_blind/source/high.php.

The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 9 of DVWA-master/vulnerabilities/sqli_blind/source/high.php.

DVWA-master/vulnerabilities/sqli_blind/source/high.php

```
5  // Get input
   $id = $_COOKIE['id'];
   ⋮
8  // Check database
   $getid = "SELECT first_name, last_name FROM users WHERE user_id = '{$id}' LIMIT 1;";
9  mysqli_query($GLOBALS["___mysqli_ston"], $getid);
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT first_name, last_name FROM users WHERE user_id = '$_COOKIE['id']' LIMIT 1;
```

Issue #375385 - DVWA-master/vulnerabilities/brute/source/low.php: 13

Path: DVWA-master/vulnerabilities/brute/source/low.php
Line: 13
Sink: mysqli_query
Source: _GET
Taint: HTTP

Code Summary

The GET parameter "username" is received in line 5 of DVWA-master/vulnerabilities/brute/source/low.php.

It is concatenated into sql markup in line 12 of DVWA-master/vulnerabilities/brute/source/low.php.

The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 13 of DVWA-master/vulnerabilities/brute/source/low.php.

DVWA-master/vulnerabilities/brute/source/low.php

```
5  // Get username
   $user = $_GET['username'];
   ⋮
9  $pass = md5($pass);
   ⋮
12 // Check the database
   $query = "SELECT * FROM `users` WHERE user = '{$user}' AND password = '{$pass}';";
13 mysqli_query($GLOBALS["___mysqli_ston"], $query);
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT * FROM `users` WHERE user = '$_GET['username']' AND password = 'abcdef0123456789';
```

Issue #375386 - DVWA-master/vulnerabilities/brute/source/medium.php: 15

Path: DVWA-master/vulnerabilities/brute/source/medium.php
Line: 15
Sink: mysqli_query
Source: _GET
Taint: HTTP

Code Summary

The GET parameter "username" is received in line 5 of DVWA-master/vulnerabilities/brute/source/low.php.

The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 15 of DVWA-master/vulnerabilities/brute/source/medium.php.

DVWA-master/vulnerabilities/brute/index.php

```
33 require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/brute/source/{$vulnerabilityFile}";
```

DVWA-master/vulnerabilities/brute/source/medium.php

```
15 mysqli_query($GLOBALS["___mysqli_ston"], $query);
```

DVWA-master/vulnerabilities/brute/source/low.php

```
5 // Get username
  $user = $_GET['username'];
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT * FROM `users` WHERE user = '$_GET['username']' AND password = 'abcdef0123456789';
```

Issue #375387 - DVWA-master/vulnerabilities/brute/source/high.php: 20

Path: DVWA-master/vulnerabilities/brute/source/high.php
Line: 20
Sink: mysqli_query
Source: _GET
Taint: HTTP

Code Summary

The GET parameter "username" is received in line 5 of DVWA-master/vulnerabilities/brute/source/low.php.

The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 20 of DVWA-master/vulnerabilities/brute/source/high.php.

DVWA-master/vulnerabilities/brute/index.php

```
33 require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/brute/source/{$vulnerabilityFile}";
```

DVWA-master/vulnerabilities/brute/source/high.php

```
20 mysqli_query($GLOBALS["___mysqli_ston"], $query);
```

DVWA-master/vulnerabilities/brute/source/low.php

```
5 // Get username
  $user = $_GET['username'];
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT * FROM `users` WHERE user = '$_GET['username']' AND password = 'abcdef0123456789';
```

Issue #375389 - DVWA-master/dvwa/includes/DBMS/MySQL.php: 56

Path: DVWA-master/dvwa/includes/DBMS/MySQL.php
Line: 56
Sink: mysqli_query
Source: _SERVER
Taint: HTTP

Code Summary

The HTTP header "PHP-SELF" is received in line 46 of DVWA-master/dvwa/includes/DBMS/MySQL.php.

It is concatenated into sql markup in line 50 of DVWA-master/dvwa/includes/DBMS/MySQL.php.

The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 56 of DVWA-master/dvwa/includes/DBMS/MySQL.php.

DVWA-master/dvwa/includes/DBMS/MySQL.php

```
   // Insert some data into users
46 // Get the base directory for the avatar media...
   $baseUrl = 'http://' . $_SERVER['SERVER_NAME'] . $_SERVER['PHP_SELF'];
47 $stripPos = strpos($baseUrl, 'setup.php');
48 $baseUrl = substr($baseUrl, 0, $stripPos) . 'hackable/users/';
⋮  ⋮
   $insert = "INSERT INTO users VALUES\r\n\t('1','admin','admin','admin',MD5('password'),'{$baseUrl}admin.jpg', NOW(), '0')
50 ,\r\n\t('2','Gordon','Brown','gordonb',MD5('abc123'),'{$baseUrl}gordonb.jpg', NOW(), '0'),\r\n\t('3','Hack','Me','1337',
   MD5('charley'),'{$baseUrl}1337.jpg', NOW(), '0'),\r\n\t('4','Pablo','Picasso','pablo',MD5('letmein'),'{$baseUrl}pablo.jp
   g', NOW(), '0'),\r\n\t('5','Bob','Smith','smithy',MD5('password'),'{$baseUrl}smithy.jpg', NOW(), '0');";
⋮  ⋮
56 mysqli_query($GLOBALS["___mysqli_ston"], $insert);
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
INSERT INTO users VALUES
('1','admin','admin','admin',MD5('password'),'http://.*$_SERVER['PHP_SELF']hackable/users/admin.jpg', NOW(), '0'),
```

```
('2','Gordon','Brown','gordonb',MD5('abc123'),'http://.*$_SERVER['PHP_SELF']hackable/users/gordonb.jpg', NOW(), '0'),
('3','Hack','Me','1337',MD5('charley'),'http://.*$_SERVER['PHP_SELF']hackable/users/1337.jpg', NOW(), '0'),
('4','Pablo','Picasso','pablo',MD5('letmein'),'http://.*$_SERVER['PHP_SELF']hackable/users/pablo.jpg', NOW(), '0'),
('5','Bob','Smith','smithy',MD5('password'),'http://.*$_SERVER['PHP_SELF']hackable/users/smithy.jpg', NOW(), '0');
```

Issue #375391 - DVWA-master/login.php: 34

Path: DVWA-master/login.php
Line: 34
Sink: mysqli_query
Source: _POST
Taint: HTTP

Code Summary

The POST parameter "username" is received in line 14 of DVWA-master/login.php.
It is concatenated into sql markup in line 33 of DVWA-master/login.php.
The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 34 of DVWA-master/login.php.

DVWA-master/login.php

```
14 $user = $_POST['username'];
15 $user = stripslashes($user);
⋮  ⋮
33 $query = "SELECT * FROM `users` WHERE user='{$user}' AND password='{$pass}';";
34 mysqli_query($GLOBALS["___mysqli_ston"], $query);
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT * FROM `users` WHERE user='$_POST['username']' AND password='.+';
```

Issue #375392 - DVWA-master/dvwa/includes/DBMS/PGSQL.php: 66

Path: DVWA-master/dvwa/includes/DBMS/PGSQL.php
Line: 66
Sink: pg_query
Source: _SERVER
Taint: HTTP
Status: Not an issue

Code Summary

The HTTP header "PHP-SELF" is received in line 56 of DVWA-master/dvwa/includes/DBMS/PGSQL.php.
It is concatenated into sql markup in line 60 of DVWA-master/dvwa/includes/DBMS/PGSQL.php.
The user-supplied data is then used unsanitized in the sensitive operation pg_query() in line 66 of DVWA-master/dvwa/includes/DBMS/PGSQL.php.

DVWA-master/dvwa/includes/DBMS/PGSQL.php

```
56 // Get the base directory for the avatar media...
   $baseUrl = 'http://' . $_SERVER['SERVER_NAME'] . $_SERVER['PHP_SELF'];
57 $stripPos = strpos($baseUrl, 'dvwa/setup.php');
58 $baseUrl = substr($baseUrl, 0, $stripPos) . 'dvwa/hackable/users/';
⋮  ⋮
60 $insert = "INSERT INTO users VALUES\r\n\t('1','admin','admin','admin',MD5('password'),'{$baseUrl}admin.jpg'),\r\n\t('2',
   'Gordon','Brown','gordonb',MD5('abc123'),'{$baseUrl}gordonb.jpg'),\r\n\t('3','Hack','Me','1337',MD5('charley'),'{$baseUr
   l}1337.jpg'),\r\n\t('4','Pablo','Picasso','pablo',MD5('letmein'),'{$baseUrl}pablo.jpg'),\r\n\t('5','bob','smith','smithy
   ',MD5('password'),'{$baseUrl}smithy.jpg');";
⋮  ⋮
66 pg_query($insert);
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
INSERT INTO users VALUES
('1','admin','admin','admin',MD5('password'),'http://.*$_SERVER['PHP_SELF']dvwa/hackable/users/admin.jpg'),
('2','Gordon','Brown','gordonb',MD5('abc123'),'http://.*$_SERVER['PHP_SELF']dvwa/hackable/users/gordonb.jpg'),
('3','Hack','Me','1337',MD5('charley'),'http://.*$_SERVER['PHP_SELF']dvwa/hackable/users/1337.jpg'),
('4','Pablo','Picasso','pablo',MD5('letmein'),'http://.*$_SERVER['PHP_SELF']dvwa/hackable/users/pablo.jpg'),
('5','bob','smith','smithy',MD5('password'),'http://.*$_SERVER['PHP_SELF']dvwa/hackable/users/smithy.jpg');
```

Issue #375396 - DVWA-master/vulnerabilities/sqli/source/high.php: 9

DVWA-

Path: master/vulnerabilities/sqli/source/high.php
Line: 9
Sink: mysqli_query
Source: _POST
Taint: Second-Order

Code Summary

User input stored in the session variable $_SESSION['id'] is received in line 5 of DVWA-master/vulnerabilities/sqli/source/high.php.

It is concatenated into sql markup in line 8 of DVWA-master/vulnerabilities/sqli/source/high.php.

The user-supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 9 of DVWA-master/vulnerabilities/sqli/source/high.php.

DVWA-master/vulnerabilities/sqli/source/high.php

```
5  // Get input
   $id = $_SESSION['id'];
   ⋮
8  // Check database
   $query = "SELECT first_name, last_name FROM users WHERE user_id = '{$id}' LIMIT 1;";
9  mysqli_query($GLOBALS["___mysqli_ston"], $query);
```

DVWA-master/vulnerabilities/sqli/session-input.php

```
12 $_SESSION['id'] = $_POST['id'];
```

SQL Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT first_name, last_name FROM users WHERE user_id = '.+' LIMIT 1;
```

3.4. File Upload

CWE: 434 **OWASP Top 10:** A5 **SANS 25 Rank:** 9 **PCI DSS:** 6.5.8 **Severity:** High

A file upload is an often misused feature that allows users to upload malicious files to the web server. For example, a remote code execution can be achieved by uploading a PHP file into the document root. Often, the file extension plays an important role and depending on the web server's configuration, different file extensions such as .php, .php4, or .php.zzz are passed to the PHP interpreter. Furthermore, uploading a .htaccess file can lead to remote code execution and uploading .htm, .html, .swf, or .svg files can lead to persistent cross-site scripting. Note, that certain built-in functions used to ensure the presence of an image, such as getimagesize(), can be bypassed. It is recommended, that uploaded files are not stored in the document root and the file's name is not in control of the user.

Issue #375358 - DVWA-master/vulnerabilities/upload/source/medium.php: 18

Path: DVWA-master/vulnerabilities/upload/source/medium.php
Line: 18
Sink: move_uploaded_file
Source: _FILES
Taint: HTTP

Code Summary

The name of the file "uploaded[name]" is received in line 6 of DVWA-master/vulnerabilities/upload/source/medium.php.

It is concatenated into path markup in line 6 of DVWA-master/vulnerabilities/upload/source/medium.php.

The user-supplied data is then used unsanitized in the sensitive operation move_uploaded_file() in line 18 of DVWA-master/vulnerabilities/upload/source/medium.php.

DVWA-master/vulnerabilities/upload/source/medium.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../');
    ⋮
5   // Where are we going to be writing to?
    $target_path = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
6   $target_path .= basename($_FILES['uploaded']['name']);
    ⋮
18  move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path);
    ⋮
427 $security = dvwaSecurityLevelGet();
    ⋮
433 $security = dvwaSecurityLevelGet();
```

Path Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
../../hackable/uploads/$_FILES['uploaded']['name']
```

Issue #375359 - DVWA-master/vulnerabilities/upload/source/low.php: 9

Path: DVWA-master/vulnerabilities/upload/source/low.php
Line: 9
Sink: move_uploaded_file
Source: _FILES
Taint: HTTP

Code Summary

The name of the file "uploaded[name]" is received in line 6 of DVWA-master/vulnerabilities/upload/source/low.php.
It is concatenated into path markup in line 6 of DVWA-master/vulnerabilities/upload/source/low.php.
The user-supplied data is then used unsanitized in the sensitive operation move_uploaded_file() in line 9 of DVWA-master/vulnerabilities/upload/source/low.php.

DVWA-master/vulnerabilities/upload/source/low.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../');
    :
5   // Where are we going to be writing to?
    $target_path = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
6   $target_path .= basename($_FILES['uploaded']['name']);
    :
9   move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path);
    :
427 $security = dvwaSecurityLevelGet();
    :
433 $security = dvwaSecurityLevelGet();
```

Path Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
../../hackable/uploads/$_FILES['uploaded']['name']
```

Issue #375362 - DVWA-master/vulnerabilities/upload/source/impossible.php: 31

Path: DVWA-master/vulnerabilities/upload/source/impossible.php
Line: 31
Sink: imagejpeg
Source: _FILES
Taint: HTTP

Code Summary

The name of the file "uploaded[name]" is received in line 9 of DVWA-master/vulnerabilities/upload/source/impossible.php.
It is concatenated into path markup in line 20 of DVWA-master/vulnerabilities/upload/source/impossible.php.
The user-supplied data is then used unsanitized in the sensitive operation imagejpeg() in line 31 of DVWA-master/vulnerabilities/upload/source/impossible.php.

DVWA-master/vulnerabilities/upload/source/impossible.php

```
9   // File information
    $uploaded_name = $_FILES['uploaded']['name'];
10  $uploaded_ext = substr($uploaded_name, strrpos($uploaded_name, '.') + 1);
    :
19  $temp_file = ini_get('upload_tmp_dir') == '' ? sys_get_temp_dir() : ini_get('upload_tmp_dir');
20  $temp_file .= DIRECTORY_SEPARATOR . md5(uniqid() . $uploaded_name) . '.' . $uploaded_ext;
    :
31  imagejpeg($img, $temp_file, 100);
```

Path Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
info/abcdef0123456789.$_FILES['uploaded']['name']
```

Issue #375363 - DVWA-master/vulnerabilities/upload/source/impossible.php: 35

Path: DVWA-master/vulnerabilities/upload/source/impossible.php
Line: 35
Sink: imagepng
Source: _FILES
Taint: HTTP

Code Summary

The name of the file "uploaded[name]" is received in line 9 of DVWA-master/vulnerabilities/upload/source/impossible.php.

The user-supplied data is then used unsanitized in the sensitive operation imagepng() in line 35 of DVWA-master/vulnerabilities/upload/source/impossible.php.

DVWA-master/vulnerabilities/upload/source/impossible.php

```
9  // File information
10 $uploaded_name = $_FILES['uploaded']['name'];
   $uploaded_ext = substr($uploaded_name, strrpos($uploaded_name, '.') + 1);
   ⋮
19 $temp_file = ini_get('upload_tmp_dir') == '' ? sys_get_temp_dir() : ini_get('upload_tmp_dir');
20 $temp_file .= DIRECTORY_SEPARATOR . md5(uniqid() . $uploaded_name) . '.' . $uploaded_ext;
   ⋮
35 imagepng($img, $temp_file, 9);
```

Path Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
info/abcdef0123456789.$_FILES['uploaded']['name']
```

Issue #375364 - DVWA-master/vulnerabilities/upload/source/impossible.php: 40

Path: DVWA-master/vulnerabilities/upload/source/impossible.php
Line: 40
Sink: rename
Source: _FILES
Taint: HTTP

Code Summary

The name of the file "uploaded[name]" is received in line 9 of DVWA-master/vulnerabilities/upload/source/impossible.php.

It is concatenated into path markup in line 40 of DVWA-master/vulnerabilities/upload/source/impossible.php.

The user-supplied data is then used unsanitized in the sensitive operation rename() in line 40 of DVWA-master/vulnerabilities/upload/source/impossible.php.

DVWA-master/vulnerabilities/upload/source/impossible.php

```
9  // File information
10 $uploaded_name = $_FILES['uploaded']['name'];
   $uploaded_ext = substr($uploaded_name, strrpos($uploaded_name, '.') + 1);
   ⋮
18 //$target_file   = basename( $uploaded_name, '.' . $uploaded_ext ) . '-';
   $target_file = md5(uniqid() . $uploaded_name) . '.' . $uploaded_ext;
19 $temp_file = ini_get('upload_tmp_dir') == '' ? sys_get_temp_dir() : ini_get('upload_tmp_dir');
20 $temp_file .= DIRECTORY_SEPARATOR . md5(uniqid() . $uploaded_name) . '.' . $uploaded_ext;
   ⋮
40 rename($temp_file, getcwd() . DIRECTORY_SEPARATOR . $target_path . $target_file);
```

Path Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
info/../../hackable/uploads/abcdef0123456789.$_FILES['uploaded']['name']
```

Issue #375365 - DVWA-master/vulnerabilities/upload/source/high.php: 20

Path: DVWA-master/vulnerabilities/upload/source/high.php
Line: 20
Sink: move_uploaded_file
Source: _FILES
Taint: HTTP

Code Summary

The name of the file "uploaded[name]" is received in line 6 of DVWA-master/vulnerabilities/upload/source/high.php.

It is concatenated into path markup in line 6 of DVWA-master/vulnerabilities/upload/source/high.php.

The user-supplied data is then used unsanitized in the sensitive operation move_uploaded_file() in line 20 of DVWA-master/vulnerabilities/upload/source/high.php.

DVWA-master/vulnerabilities/upload/source/high.php

```
3  define('DVWA_WEB_PAGE_TO_ROOT', '../');
   ⋮
5  // Where are we going to be writing to?
   $target_path = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
6  $target_path .= basename($_FILES['uploaded']['name']);
```

```
⋮
20  move_uploaded_file($uploaded_tmp, $target_path);
⋮
427 $security = dvwaSecurityLevelGet();
⋮
433 $security = dvwaSecurityLevelGet();
```

Path Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
../../hackable/uploads/$_FILES['uploaded']['name']
```

3.5. PHP Object Injection

CWE: 502 **OWASP Top 10:** A4 **SANS 25 Rank:** 18 **PCI DSS:** 6.5.1 **Severity:** High

A PHP object injection vulnerability occurs when user input is deserialized. An attacker can inject objects of any class type with arbitrary properties into the application's scope by providing these in a serialized string format. When the object defined by the attacker is instantiated, certain magic methods of the injected class type, such as __wakeup() and __destruct(), are automatically invoked. Depending on the magic methods defined within the application's code and the involved properties that can be controlled by the attacker, further vulnerabilities or code can be triggered. It is highly recommended to avoid the deserialization of user input and to use the built-in functions json_encode() instead.

Issue #375345 - DVWA-master/external/phpids/0.6/lib/IDS/Converter.php: 632

Path: DVWA-master/external/phpids/0.6/lib/IDS/Converter.php
Line: 632
Sink: unserialize
Source: _REQUEST
Taint: HTTP

Code Summary

The GET parameter is received in method IDS_Monitor::run() in line 236 of DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php.

The user-supplied data is then used unsanitized in the sensitive operation unserialize() in method IDS_Converter::runCentrifuge() in line 632 of DVWA-master/external/phpids/0.6/lib/IDS/Converter.php.

DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php

```
    class IDS_Monitor
    {
233 public function run()
    {
⋮
236 foreach ($this->request as $key => $value) {
237 $this->_iterate($key, $value);
⋮
    class IDS_Monitor
    {
253 private function _iterate($key, $value)
    {
⋮
259 $this->_detect($key, $value);
⋮
    class IDS_Monitor
    {
285 private function _detect($key, $value)
    {
⋮
308 list($key, $value) = $this->_purifyValues($key, $value);
⋮
313 list($key, $value) = $this->_jsonDecodeValues($key, $value);
⋮
    // scan keys if activated via config
322 $key = $this->scanKeys ? IDS_Converter::runAll($key) : $key;
⋮
324 IDS_Converter::runCentrifuge($key, $this);
```

DVWA-master/external/phpids/0.6/lib/IDS/Converter.php

```
    class IDS_Converter
    {
627 public static function runCentrifuge($value, IDS_Monitor $monitor = null)
    {
⋮
```

```
632 unserialize($value);
```

Input Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
$_REQUEST[.*]
```

3.6. Cross-Site Scripting

CWE: 79 **OWASP Top 10:** A3 **SANS 25 Rank:** 4 **PCI DSS:** 6.5.7 **Severity:** Medium

A reflected cross-site scripting (XSS) vulnerability occurs when user input is embedded unsanitized into the HTML response page of the web application. It allows an attacker to inject arbitrary HTML or JavaScript code into the response page of a tampered request. Usually, this attack is performed by crafting a malicious link that is sent to a victim. When opened, the attacker's JavaScript payload within the link is reflected by the application and executed in the victim's browser in the context of the web application's domain. This enables the attacker to perform phishing attacks or to steal cookies associated with the domain.

3.6.1. Cross-Site Scripting (normal tag)

CWE: 80 **OWASP Top 10:** A3 **SANS 25 Rank:** 4 **PCI DSS:** 6.5.7 **Severity:** Medium

The detected injection occurs between two HTML elements. An attacker can inject a new HTML element, such as the <script> element, to invoke the JavaScript interpreter and execute arbitrary JavaScript code. To prevent this, the HTML markup characters < and > must be transformed to their corresponding HTML entities < and >. For this purpose, it is recommended to use the PHP built-in function htmlentities() that performs this transformation.

Issue #375348 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 278

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 278
Sink: echo
Source: _GET
Taint: HTTP

Code Summary
The GET parameter "name" is received in line 6 of DVWA-master/vulnerabilities/xss_r/source/low.php.
It is concatenated into html markup in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.
The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/xss_r/index.php

```
4  require_once DVWA_WEB_PAGE_TO_ROOT . 'dvwa/includes/dvwaPage.inc.php';
⋮  ⋮
8  $page = dvwaPageNewGrab();
9  $page['title'] = 'Vulnerability: Reflected Cross Site Scripting (XSS)' . $page['title_separator'] . $page['title'];
10 $page['page_id'] = 'xss_r';
11 $page['help_button'] = 'xss_r';
12 $page['source_button'] = 'xss_r';
⋮  ⋮
32 require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/xss_r/source/{$vulnerabilityFile}";
⋮  ⋮
34 $page['body'] .= "\r\n<div class=\"body_padded\">\r\n\t<h1>Vulnerability: Reflected Cross Site Scripting (XSS)</h1>\r\n\r\n\t<div class=\"vulnerable_code_area\">\r\n\t\t<form name=\"XSS\" action=\"#\" method=\"GET\">\r\n\t\t\t<p>\r\n\t\t\tWhat's your name?\r\n\t\t\t<input type=\"text\" name=\"name\">\r\n\t\t\t<input type=\"submit\" value=\"Submit\">\r\n\t\t\t</p>\n";
⋮  ⋮
47 $page['body'] .= "\t\t\t" . tokenField();
⋮  ⋮
49 $page['body'] .= "\r\n\t\t</form>\r\n\t\t{$html}\r\n\t</div>\r\n\r\n\t<h2>More Information</h2>\r\n\t<ul>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https://en.wikipedia.org/wiki/Cross-site_scripting') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://www.cgisecurity.com/xss-faq.html') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://www.scriptalert1.com/') . "</li>\r\n\t</ul>\r\n</div>\n";
⋮  ⋮
64 dvwaHtmlEcho($page);
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../../');
⋮   ⋮
183 function dvwaHtmlEcho($pPage)
```

```
184 {
    $menuBlocks = array();

    :
186 $menuBlocks['home'] = array();
    :
188 $menuBlocks['home'][] = array('id' => 'home', 'name' => 'Home', 'url' => '.');
189 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
190 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup / Reset DB', 'url' => 'setup.php');
    :
193 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup DVWA', 'url' => 'setup.php');
194 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
    :
198 $menuBlocks['vulnerabilities'] = array();
199 $menuBlocks['vulnerabilities'][] = array('id' => 'brute', 'name' => 'Brute Force', 'url' => 'vulnerabilities/brute/');
200 $menuBlocks['vulnerabilities'][] = array('id' => 'exec', 'name' => 'Command Injection', 'url' => 'vulnerabilities/exec/
    ');
201 $menuBlocks['vulnerabilities'][] = array('id' => 'csrf', 'name' => 'CSRF', 'url' => 'vulnerabilities/csrf/');
202 $menuBlocks['vulnerabilities'][] = array('id' => 'fi', 'name' => 'File Inclusion', 'url' => 'vulnerabilities/fi/.?page=
    include.php');
203 $menuBlocks['vulnerabilities'][] = array('id' => 'upload', 'name' => 'File Upload', 'url' => 'vulnerabilities/upload/')
    ;
204 $menuBlocks['vulnerabilities'][] = array('id' => 'captcha', 'name' => 'Insecure CAPTCHA', 'url' => 'vulnerabilities/cap
    tcha/');
205 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli', 'name' => 'SQL Injection', 'url' => 'vulnerabilities/sqli/');
206 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli_blind', 'name' => 'SQL Injection (Blind)', 'url' => 'vulnerabili
    ties/sqli_blind/');
207 $menuBlocks['vulnerabilities'][] = array('id' => 'weak_id', 'name' => 'Weak Session IDs', 'url' => 'vulnerabilities/wea
    k_id/');
208 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_d', 'name' => 'XSS (DOM)', 'url' => 'vulnerabilities/xss_d/');
209 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_r', 'name' => 'XSS (Reflected)', 'url' => 'vulnerabilities/xss_r/
    ');
210 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_s', 'name' => 'XSS (Stored)', 'url' => 'vulnerabilities/xss_s/');
    :
213 $menuBlocks['meta'] = array();
    :
215 $menuBlocks['meta'][] = array('id' => 'security', 'name' => 'DVWA Security', 'url' => 'security.php');
216 $menuBlocks['meta'][] = array('id' => 'phpinfo', 'name' => 'PHP Info', 'url' => 'phpinfo.php');
    :
218 $menuBlocks['meta'][] = array('id' => 'about', 'name' => 'About', 'url' => 'about.php');
    :
221 $menuBlocks['logout'] = array();
222 $menuBlocks['logout'][] = array('id' => 'logout', 'name' => 'Logout', 'url' => 'logout.php');
    :
225 $menuHtml = '';
    :
227 foreach ($menuBlocks as $menuBlock) {
    :
229 foreach ($menuBlock as $menuItem) {
230 $selectedClass = $menuItem['id'] == $pPage['page_id'] ? 'selected' : '';
231 $fixedUrl = DVWA_WEB_PAGE_TO_ROOT . $menuItem['url'];
232 $menuBlockHtml .= "<li onclick=\"window.location='{$fixedUrl}'\" class=\"{$selectedClass}\"><a href=\"{$fixedUrl}\">{$m
    enuItem['name']}</a></li>\n";
    :
234 $menuHtml .= "<ul class=\"menuBlocks\">{$menuBlockHtml}</ul>";
    :
241 $securityLevelHtml = 'low';
    :
244 $securityLevelHtml = 'medium';
    :
247 $securityLevelHtml = 'high';
    :
250 $securityLevelHtml = 'impossible';
    :
255 // -- END (security cookie)
    $phpIdsHtml = '<em>PHPIDS:</em> ' . (dvwaPhpIdsIsEnabled() ? 'enabled' : 'disabled');
    :
259 $messagesHtml;
    :
263 $systemInfoHtml = "";
    :
265 $systemInfoHtml = "<div align=\"left\">{$userInfoHtml}<br /><em>Security Level:</em> {$securityLevelHtml}<br />{$phpIds
    Html}</div>";
    :
267 $systemInfoHtml = dvwaButtonSourceHtmlGet($pPage['source_button']) . " {$systemInfoHtml}";
    :
270 $systemInfoHtml = dvwaButtonHelpHtmlGet($pPage['help_button']) . " {$systemInfoHtml}";
    :
    // Date in the past
    echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dt
    d\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv=\"Content-Type\" cont
```

```
278  ent=\"text/html; charset=UTF-8\" />\r\n\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\r\n\t\t<link rel=\"stylesheet\" type=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/main.css\" />\r\n\r\n\r\n\t\t<link rel=\"icon\" type=\"\\image/ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\r\n\t\t<script type=\"text/javascript\" src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/js/dvwaPage.js\"></script>\r\n\r\n\r\n\t</head>\r\n\r\n\r\n\t<body class=\"home\">\r\n\t\t<div id=\"container\">\r\n\r\n\r\n\t\t\t<div id=\"header\">\r\n\r\n\r\n\t\t\t\t<img src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/images/logo.png\" alt=\"Damn Vulnerable Web Application\" />\r\n\r\n\r\n\t\t\t</div>\r\n\r\n\r\n\t\t\t<div id=\"main_menu\">\r\n\r\n\r\n\t\t\t\t<div id=\"main_menu_padded\">\r\n\t\t\t\t{$menuHtml}\r\n\t\t\t</div>\r\n\r\n\r\n\t\t</div>\r\n\r\n\r\n\t\t<div id=\"main_body\">\r\n\r\n\r\n\t\t\t{$pPage['body']}\r\n\t\t\t<br /><br />\r\n\t\t\t{$messagesHtml}\r\n\r\n\r\n\t\t</div>\r\n\r\n\r\n\t\t<div class=\"clear\">\r\n\t\t</div>\r\n\r\n\r\n\t\t<div id=\"system_info\">\r\n\t\t{$systemInfoHtml}\r\n\t\t</div>\r\n\r\n\r\n\t\t<div id=\"footer\">\r\n\r\n\r\n\t\t<p>Damn Vulnerable Web Application (DVWA) v" . dvwaVersionGet() . "</p>\r\n\r\n\r\n\t</div>\r\n\r\n\r\n\t</body>\r\n\r\n</html>";
⋮
427  $security = dvwaSecurityLevelGet();
⋮
433  $security = dvwaSecurityLevelGet();
```

DVWA-master/vulnerabilities/xss_r/source/low.php

```
6  // Feedback for end user
   $html .= '<pre>Hello ' . $_GET['name'] . '</pre>';
```

HTML Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
<script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script> </head> <body class="home"> <div id="container">
<div id="header"> <img src="../../dvwa/images/logo.png" alt="Damn Vulnerable Web Application" /> </div> <div id="main_menu">
<div id="main_menu_padded"> <ul class="menuBlocks"><li onclick="window.location='../../vulnerabilities/captcha/'"
class="selected"><a href="../../logout">logout</a></li> </ul> </div> </div> <div id="main_body"> <div class="body_padded">
<h1>Vulnerability: Reflected Cross Site Scripting (XSS)</h1> <div class="vulnerable_code_area"> <form name="XSS" action="#"
method="GET"> <p> What's your name? <input type="text" name="name"> <input type="submit" value="Submit"> </p> <input
type='hidden' name='user_token' value='.+' /> </form> .+<pre>Hello $_GET['name']</pre><pre>Hello $_GET['name']</pre>
<pre>Hello $_GET['name']</pre><pre>Hello $_GET['name']</pre> </div> <h2>More Information</h2> <ul> <li><a
href="https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)" target="_blank">https://www.owasp.org/index.php/Cross-
site_Scripting_(XSS)</a></li> <li><a href="https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet"
target="_blank">https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet</a></li> <li><a
href="https://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">https://en.wikipedia.org/wiki/Cross-
site_scripting</a></li> <li><a href="http://www.cgisecurity.com/xss-faq.html"
target="_blank">http://www.cgisecurity.com/xss-faq.html</a></li> <li><a href="http://www.scriptalert1.com/"
target="_blank">http://www.scriptalert1.com/</a></li> </ul> </div> <br /><br /> 0 </div> <div class="clear"> </div> <div
id="system_info"> <input type="button" value="View Source" class="popup_button" onClick="javascript:popUp(
'../../vulnerabilities/view_source.php?id=P_29_&security=U_28_' )"> <div align="left"><em>Username:</em> Array<br />
<em>Security Level:</em> low<br /><em>PHPIDS:</em> enabled</div> </div> <div id="footer"> <p>Damn Vulnerable Web Application
(DVWA) v1.10 *Development*</p> </div> </div> </body> </html>
```

Issue #375366 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 278

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 278
Sink: echo
Source: _FILES
Taint: HTTP

Code Summary

The name of the file "uploaded[name]" is received in line 6 of DVWA-master/vulnerabilities/upload/source/low.php.

It is concatenated into html markup in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/upload/index.php

```
4  require_once DVWA_WEB_PAGE_TO_ROOT . 'dvwa/includes/dvwaPage.inc.php';
⋮
8  $page = dvwaPageNewGrab();
9  $page['title'] = 'Vulnerability: File Upload' . $page['title_separator'] . $page['title'];
10 $page['page_id'] = 'upload';
11 $page['help_button'] = 'upload';
12 $page['source_button'] = 'upload';
⋮
32 require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/upload/source/{$vulnerabilityFile}";
⋮
35 // Check if folder is writeable
   $WarningHtml = '';
⋮
37 $WarningHtml .= "<div class=\"warning\">Incorrect folder permissions: {$PHPUploadPath}<br /><em>Folder is not writable.</em></div>";
⋮
```

```
41  $WarningHtml .= "<div class=\"warning\">The PHP module <em>GD is not installed</em>.</div>";
    :
    $page['body'] .= "\r\n<div class=\"body_padded\">\r\n\t<h1>Vulnerability: File Upload</h1>\r\n\r\n\t{$WarningHtml}\r\n\r
    \n\t<div class=\"vulnerable_code_area\">\r\n\t\t<form enctype=\"multipart/form-data\" action=\"#\" method=\"POST\">\r\n\
44  t\t\t<input type=\"hidden\" name=\"MAX_FILE_SIZE\" value=\"100000\" />\r\n\t\t\tChoose an image to upload:<br /><br />\r
    \n\t\t\t<input name=\"uploaded\" type=\"file\" /><br />\r\n\t\t\t<br />\r\n\t\t\t<input type=\"submit\" name=\"Upload\"
    value=\"Upload\" />\n";
    :
59  $page['body'] .= "\t\t\t" . tokenField();
    :
    $page['body'] .= "\r\n\t\t</form>\r\n\t\t{$html}\r\n\t</div>\r\n\r\n\t<h2>More Information</h2>\r\n\t<ul>\r\n\t\t<li>" .
61  dvwaExternalLinkUrlGet('https://www.owasp.org/index.php/Unrestricted_File_Upload') . "</li>\r\n\t\t<li>" . dvwaExternalL
    inkUrlGet('https://blogs.securiteam.com/index.php/archives/1268') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https:
    //www.acunetix.com/websitesecurity/upload-forms-threat/') . "</li>\r\n\t</ul>\r\n</div>";
    :
74  dvwaHtmlEcho($page);
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../../');
    :
183 function dvwaHtmlEcho($pPage)
184 {
184 $menuBlocks = array();
    :
186 $menuBlocks['home'] = array();
    :
188 $menuBlocks['home'][] = array('id' => 'home', 'name' => 'Home', 'url' => '.');
189 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
190 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup / Reset DB', 'url' => 'setup.php');
    :
193 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup DVWA', 'url' => 'setup.php');
194 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
    :
198 $menuBlocks['vulnerabilities'] = array();
199 $menuBlocks['vulnerabilities'][] = array('id' => 'brute', 'name' => 'Brute Force', 'url' => 'vulnerabilities/brute/');
200 $menuBlocks['vulnerabilities'][] = array('id' => 'exec', 'name' => 'Command Injection', 'url' => 'vulnerabilities/exec/
    ');
201 $menuBlocks['vulnerabilities'][] = array('id' => 'csrf', 'name' => 'CSRF', 'url' => 'vulnerabilities/csrf/');
202 $menuBlocks['vulnerabilities'][] = array('id' => 'fi', 'name' => 'File Inclusion', 'url' => 'vulnerabilities/fi/.?page=
    include.php');
203 $menuBlocks['vulnerabilities'][] = array('id' => 'upload', 'name' => 'File Upload', 'url' => 'vulnerabilities/upload/')
    ;
204 $menuBlocks['vulnerabilities'][] = array('id' => 'captcha', 'name' => 'Insecure CAPTCHA', 'url' => 'vulnerabilities/cap
    tcha/');
205 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli', 'name' => 'SQL Injection', 'url' => 'vulnerabilities/sqli/');
206 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli_blind', 'name' => 'SQL Injection (Blind)', 'url' => 'vulnerabili
    ties/sqli_blind/');
207 $menuBlocks['vulnerabilities'][] = array('id' => 'weak_id', 'name' => 'Weak Session IDs', 'url' => 'vulnerabilities/wea
    k_id/');
208 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_d', 'name' => 'XSS (DOM)', 'url' => 'vulnerabilities/xss_d/');
209 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_r', 'name' => 'XSS (Reflected)', 'url' => 'vulnerabilities/xss_r/
    ');
210 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_s', 'name' => 'XSS (Stored)', 'url' => 'vulnerabilities/xss_s/');
    :
213 $menuBlocks['meta'] = array();
    :
215 $menuBlocks['meta'][] = array('id' => 'security', 'name' => 'DVWA Security', 'url' => 'security.php');
216 $menuBlocks['meta'][] = array('id' => 'phpinfo', 'name' => 'PHP Info', 'url' => 'phpinfo.php');
    :
218 $menuBlocks['meta'][] = array('id' => 'about', 'name' => 'About', 'url' => 'about.php');
    :
221 $menuBlocks['logout'] = array();
222 $menuBlocks['logout'][] = array('id' => 'logout', 'name' => 'Logout', 'url' => 'logout.php');
    :
225 $menuHtml = '';
    :
227 foreach ($menuBlocks as $menuBlock) {
    :
229 foreach ($menuBlock as $menuItem) {
230 $selectedClass = $menuItem['id'] == $pPage['page_id'] ? 'selected' : '';
231 $fixedUrl = DVWA_WEB_PAGE_TO_ROOT . $menuItem['url'];
232 $menuBlockHtml .= "<li onclick=\"window.location='{$fixedUrl}'\" class=\"{$selectedClass}\"><a href=\"{$fixedUrl}\">{$m
    enuItem['name']}</a></li>\n";
    :
234 $menuHtml .= "<ul class=\"menuBlocks\">{$menuBlockHtml}</ul>";
    :
241 $securityLevelHtml = 'low';
    :
244 $securityLevelHtml = 'medium';
```

```
247 $securityLevelHtml = 'high';
250 $securityLevelHtml = 'impossible';
255 // -- END (security cookie)
$phpIdsHtml = '<em>PHPIDS:</em> ' . (dvwaPhpIdsIsEnabled() ? 'enabled' : 'disabled');
259 $messagesHtml;
263 $systemInfoHtml = "";
265 $systemInfoHtml = "<div align=\"left\">{$userInfoHtml}<br /><em>Security Level:</em> {$securityLevelHtml}<br />{$phpIds
Html}</div>";
267 $systemInfoHtml = dvwaButtonSourceHtmlGet($pPage['source_button']) . " {$systemInfoHtml}";
270 $systemInfoHtml = dvwaButtonHelpHtmlGet($pPage['help_button']) . " {$systemInfoHtml}";
278 // Date in the past
echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dt
d\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv=\"Content-Type\" cont
ent=\"text/html; charset=UTF-8\" />\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\t\t<link rel=\"stylesheet\" typ
e=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/main.css\" />\r\n\r\n\t\t<link rel=\"icon\" type=\"\\image/
ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t\t<script type=\"text/javascript\" src=\"" . DVWA_W
EB_PAGE_TO_ROOT . "dvwa/js/dvwaPage.js\"></script>\r\n\r\n\t</head>\r\n\r\n\t<body class=\"home\">\r\n\t\t<div id=\"con
tainer\">\r\n\r\n\t\t<div id=\"header\">\r\n\r\n\t\t<img src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/images/logo.png\
" alt=\"Damn Vulnerable Web Application\" />\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_menu\">\r\n\r\n\t\t<di
v id=\"main_menu_padded\">\r\n\t\t{$menuHtml}\r\n\t\t</div>\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_bod
y\">\r\n\r\n\t\t{$pPage['body']}\r\n\t\t<br /><br />\r\n\t\t{$messagesHtml}\r\n\r\n\t\t</div>\r\n\r\n\t
\t<div class=\"clear\">\r\n\t\t</div>\r\n\r\n\t\t<div id=\"system_info\">\r\n\t\t{$systemInfoHtml}\r\n\t\t</d
iv>\r\n\r\n\t\t<div id=\"footer\">\r\n\r\n\t\t<p>Damn Vulnerable Web Application (DVWA) v" . dvwaVersionGet() . "
</p>\r\n\r\n\t\t</div>\r\n\r\n\t</div>\r\n\r\n\t</body>\r\n\r\n</html>";
427 $security = dvwaSecurityLevelGet();
433 $security = dvwaSecurityLevelGet();
```

DVWA-master/vulnerabilities/upload/source/low.php

```
6 $target_path .= basename($_FILES['uploaded']['name']);
```

HTML Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
<script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script> </head> <body class="home"> <div id="container">
<div id="header"> <img src="../../dvwa/images/logo.png" alt="Damn Vulnerable Web Application" /> </div> <div id="main_menu">
<div id="main_menu_padded"> <ul class="menuBlocks"><li onclick="window.location='../../vulnerabilities/captcha/'"
class="selected"><a href="../../logout">logout</a></li> </ul> </div> </div> <div id="main_body"> <div class="body_padded">
<h1>Vulnerability: File Upload</h1> <div class="warning">Incorrect folder permissions: <br /><em>Folder is not writable.
</em></div><div class="warning">The PHP module <em>GD is not installed</em>.</div> <div class="vulnerable_code_area"> <form
enctype="multipart/form-data" action="#" method="POST"> <input type="hidden" name="MAX_FILE_SIZE" value="100000" /> Choose
an image to upload:<br /><br /> <input name="uploaded" type="file" /><br /> <br /> <input type="submit" name="Upload"
value="Upload" /> <input type='hidden' name='user_token' value='.+' /> </form> .+<pre>Your image was not uploaded.</pre>
<pre>Your image was not uploaded.</pre><pre>Your image was not uploaded.</pre> </div> <h2>More Information</h2> <ul> <li><a
href="https://www.owasp.org/index.php/Unrestricted_File_Upload"
target="_blank">https://www.owasp.org/index.php/Unrestricted_File_Upload</a></li> <li><a
href="https://blogs.securiteam.com/index.php/archives/1268"
target="_blank">https://blogs.securiteam.com/index.php/archives/1268</a></li> <li><a
href="https://www.acunetix.com/websitesecurity/upload-forms-threat/"
target="_blank">https://www.acunetix.com/websitesecurity/upload-forms-threat/</a></li> </ul> </div> <br /><br /> 0 </div>
<div class="clear"> </div> <div id="system_info"> <input type="button" value="View Source" class="popup_button"
onClick="javascript:popUp( '../../vulnerabilities/view_source.php?id=P_29_&security=U_28_' )"> <div align="left">
<em>Username:</em> Array<br /><em>Security Level:</em> low<br /><em>PHPIDS:</em> enabled</div> </div> <div id="footer">
<p>Damn Vulnerable Web Application (DVWA) v1.10 *Development*</p> </div> </div> </body> </html>
```

Issue #375368 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 385

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 385
Sink: echo
Source: _POST
Taint: HTTP

Code Summary

The POST parameter "id" is received in line 12 of DVWA-master/vulnerabilities/sqli/session-input.php.

It is concatenated into html markup in function dvwaSourceHtmlEcho() in line 385 of DVWA-master/dvwa/includes/dvwaPage.inc.php.
The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaSourceHtmlEcho() in line 385 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/sqli/session-input.php

```
8  $page = dvwaPageNewGrab();
9  $page['title'] = 'SQL Injection Session Input' . $page['title_separator'] . $page['title'];
⋮
12 $_SESSION['id'] = $_POST['id'];
⋮
14 //$page[ 'body' ] .= "Session ID set!<br /><br /><br />";
   $page['body'] .= "Session ID: {$_SESSION['id']}<br /><br /><br />";
15 $page['body'] .= "<script>window.opener.location.reload(true);</script>";
⋮
18 $page['body'] .= "\r\n<form action=\"#\" method=\"POST\">\r\n\t<input type=\"text\" size=\"15\" name=\"id\">\r\n\t<input
   type=\"submit\" name=\"Submit\" value=\"Submit\">\r\n</form>\r\n<hr />\r\n<br />\r\n\r\n<button onclick=\"self.close();\
   ">Close</button>";
⋮
28 dvwaSourceHtmlEcho($page);
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../');
⋮
379 function dvwaSourceHtmlEcho($pPage)
    {
⋮
    // Date in the past
    echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dt
    d\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\r\n\t\t<meta http-equiv=\"Content-Type\"
385 content=\"text/html; charset=UTF-8\" />\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\t\t<link rel=\"stylesheet\"
    type=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/source.css\" />\r\n\r\n\t\t<link rel=\"icon\" type=\"\\i
    mage/ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t</head>\r\n\r\n\t<body>\r\n\r\n\t\t<div id=\"c
    ontainer\">\r\n\r\n\t\t\t{$pPage['body']}\r\n\r\n\t\t</div>\r\n\r\n\t</body>\r\n\r\n</html>";
⋮
427 $security = dvwaSecurityLevelGet();
⋮
433 $security = dvwaSecurityLevelGet();
```

HTML Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
<div id="container"> Session ID: $_POST['id']<br /><br /><br /><script>window.opener.location.reload(true);</script> <form
action="#" method="POST"> <input type="text" size="15" name="id"> <input type="submit" name="Submit" value="Submit"> </form>
<hr /> <br /> <button onclick="self.close();">Close</button> </div> </body> </html>
```

Issue #375370 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 278

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 278
Sink: echo
Source: _REQUEST
Taint: HTTP

Code Summary
The GET parameter "id" is received in line 5 of DVWA-master/vulnerabilities/sqli/source/low.php.
It is concatenated into html markup in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.
The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/sqli/index.php

```
4  require_once DVWA_WEB_PAGE_TO_ROOT . 'dvwa/includes/dvwaPage.inc.php';
⋮
8  $page = dvwaPageNewGrab();
9  $page['title'] = 'Vulnerability: SQL Injection' . $page['title_separator'] . $page['title'];
10 $page['page_id'] = 'sqli';
11 $page['help_button'] = 'sqli';
12 $page['source_button'] = 'sqli';
⋮
16 $method = 'GET';
⋮
24 $method = 'POST';
⋮
34 require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/sqli/source/{$vulnerabilityFile}";
⋮
```

```
37  // Is PHP function magic_quotee enabled?
    $WarningHtml = '';
⋮   ⋮
39  $WarningHtml .= "<div class=\"warning\">The PHP function \"<em>Magic Quotes</em>\" is enabled.</div>";
⋮   ⋮
43  $WarningHtml .= "<div class=\"warning\">The PHP function \"<em>Safe mode</em>\" is enabled.</div>";
⋮   ⋮
46  $page['body'] .= "\r\n<div class=\"body_padded\">\r\n\t<h1>Vulnerability: SQL Injection</h1>\r\n\r\n\t{$WarningHtml}\r\n
    \r\n\t<div class=\"vulnerable_code_area\">";
⋮   ⋮
54  $page['body'] .= "Click <a href=\"#\" onClick=\"javascript:popUp('session-input.php');return false;\">here to change you
    r ID</a>.";
⋮   ⋮
57  $page['body'] .= "\r\n\t\t<form action=\"#\" method=\"{$method}\">\r\n\t\t\t<p>\r\n\t\t\t\tUser ID:";
⋮   ⋮
62  $page['body'] .= "\n\t\t\t\t<select name=\"id\">";
⋮   ⋮
64  for ($i = 1; $i < $number_of_rows + 1; $i++) {
65  $page['body'] .= "</select>";
⋮   ⋮
68  $page['body'] .= "\n\t\t\t\t<input type=\"text\" size=\"15\" name=\"id\">";
⋮   ⋮
70  $page['body'] .= "\n\t\t\t\t<input type=\"submit\" name=\"Submit\" value=\"Submit\">\r\n\t\t\t</p>\n";
⋮   ⋮
74  $page['body'] .= "\t\t\t" . tokenField();
⋮   ⋮
76  $page['body'] .= "\r\n\t\t</form>";
⋮   ⋮
    $page['body'] .= "\r\n\t\t{$html}\r\n\t</div>\r\n\r\n\t<h2>More Information</h2>\r\n\t<ul>\r\n\t\t<li>" . dvwaExternalLi
    nkUrlGet('http://www.securiteam.com/securityreviews/5DP0N1P76E.html') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('ht
    tps://en.wikipedia.org/wiki/SQL_injection') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://ferruh.mavituna.com/s
79  ql-injection-cheatsheet-oku/') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://pentestmonkey.net/cheat-sheet/sql-
    injection/mysql-sql-injection-cheat-sheet') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https://www.owasp.org/index.
    php/SQL_Injection') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://bobby-tables.com/') . "</li>\r\n\t</ul>\r\n</
    div>\n";
⋮   ⋮
94  dvwaHtmlEcho($page);
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../../');
⋮   ⋮
183 function dvwaHtmlEcho($pPage)
184 {
185 $menuBlocks = array();
⋮   ⋮
186 $menuBlocks['home'] = array();
⋮   ⋮
188 $menuBlocks['home'][] = array('id' => 'home', 'name' => 'Home', 'url' => '.');
189 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
190 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup / Reset DB', 'url' => 'setup.php');
⋮   ⋮
193 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup DVWA', 'url' => 'setup.php');
194 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
⋮   ⋮
198 $menuBlocks['vulnerabilities'] = array();
199 $menuBlocks['vulnerabilities'][] = array('id' => 'brute', 'name' => 'Brute Force', 'url' => 'vulnerabilities/brute/');
200 $menuBlocks['vulnerabilities'][] = array('id' => 'exec', 'name' => 'Command Injection', 'url' => 'vulnerabilities/exec/
    ');
201 $menuBlocks['vulnerabilities'][] = array('id' => 'csrf', 'name' => 'CSRF', 'url' => 'vulnerabilities/csrf/');
202 $menuBlocks['vulnerabilities'][] = array('id' => 'fi', 'name' => 'File Inclusion', 'url' => 'vulnerabilities/fi/.?page=
    include.php');
203 $menuBlocks['vulnerabilities'][] = array('id' => 'upload', 'name' => 'File Upload', 'url' => 'vulnerabilities/upload/')
    ;
204 $menuBlocks['vulnerabilities'][] = array('id' => 'captcha', 'name' => 'Insecure CAPTCHA', 'url' => 'vulnerabilities/cap
    tcha/');
205 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli', 'name' => 'SQL Injection', 'url' => 'vulnerabilities/sqli/');
206 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli_blind', 'name' => 'SQL Injection (Blind)', 'url' => 'vulnerabili
    ties/sqli_blind/');
207 $menuBlocks['vulnerabilities'][] = array('id' => 'weak_id', 'name' => 'Weak Session IDs', 'url' => 'vulnerabilities/wea
    k_id/');
208 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_d', 'name' => 'XSS (DOM)', 'url' => 'vulnerabilities/xss_d/');
209 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_r', 'name' => 'XSS (Reflected)', 'url' => 'vulnerabilities/xss_r/
    ');
210 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_s', 'name' => 'XSS (Stored)', 'url' => 'vulnerabilities/xss_s/');
⋮   ⋮
213 $menuBlocks['meta'] = array();
⋮   ⋮
215 $menuBlocks['meta'][] = array('id' => 'security', 'name' => 'DVWA Security', 'url' => 'security.php');
```

```
216 $menuBlocks['meta'][] = array('id' => 'phpinfo', 'name' => 'PHP Info', 'url' => 'phpinfo.php');
218 $menuBlocks['meta'][] = array('id' => 'about', 'name' => 'About', 'url' => 'about.php');
221 $menuBlocks['logout'] = array();
222 $menuBlocks['logout'][] = array('id' => 'logout', 'name' => 'Logout', 'url' => 'logout.php');
225 $menuHtml = '';
227 foreach ($menuBlocks as $menuBlock) {
229 foreach ($menuBlock as $menuItem) {
230 $selectedClass = $menuItem['id'] == $pPage['page_id'] ? 'selected' : '';
231 $fixedUrl = DVWA_WEB_PAGE_TO_ROOT . $menuItem['url'];
232 $menuBlockHtml .= "<li onclick=\"window.location='{$fixedUrl}'\" class=\"{$selectedClass}\"><a href=\"{$fixedUrl}\">{$menuItem['name']}</a></li>\n";
234 $menuHtml .= "<ul class=\"menuBlocks\">{$menuBlockHtml}</ul>";
241 $securityLevelHtml = 'low';
244 $securityLevelHtml = 'medium';
247 $securityLevelHtml = 'high';
250 $securityLevelHtml = 'impossible';
255 // -- END (security cookie)
    $phpIdsHtml = '<em>PHPIDS:</em> ' . (dvwaPhpIdsIsEnabled() ? 'enabled' : 'disabled');
259 $messagesHtml;
263 $systemInfoHtml = "";
265 $systemInfoHtml = "<div align=\"left\">{$userInfoHtml}<br /><em>Security Level:</em> {$securityLevelHtml}<br />{$phpIdsHtml}</div>";
267 $systemInfoHtml = dvwaButtonSourceHtmlGet($pPage['source_button']) . " {$systemInfoHtml}";
270 $systemInfoHtml = dvwaButtonHelpHtmlGet($pPage['help_button']) . " {$systemInfoHtml}";
278 // Date in the past
    echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\" />\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\t\t<link rel=\"stylesheet\" type=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/main.css\" />\r\n\r\n\t\t<link rel=\"icon\" type=\"\\image/ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t\t<script type=\"text/javascript\" src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/js/dvwaPage.js\"></script>\r\n\r\n\t</head>\r\n\r\n\t<body class=\"home\">\r\n\t\t<div id=\"container\">\r\n\r\n\t\t<div id=\"header\">\r\n\r\n\t\t<img src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/images/logo.png\" alt=\"Damn Vulnerable Web Application\" />\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_menu\">\r\n\r\n\t\t<div id=\"main_menu_padded\">\r\n\t\t{$menuHtml}\r\n\t\t</div>\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_body\">\r\n\r\n\t\t{$pPage['body']}\r\n\t\t<br /><br />\r\n\t\t{$messagesHtml}\r\n\r\n\t\t</div>\r\n\r\n\t\t<div class=\"clear\">\r\n\t\t</div>\r\n\r\n\t\t<div id=\"system_info\">\r\n\t\t{$systemInfoHtml}\r\n\t\t</div>\r\n\r\n\t\t<div id=\"footer\">\r\n\r\n\t\t<p>Damn Vulnerable Web Application (DVWA) v" . dvwaVersionGet() . "</p>\r\n\r\n\t\t</div>\r\n\r\n\t</body>\r\n\r\n</html>";
427 $security = dvwaSecurityLevelGet();
433 $security = dvwaSecurityLevelGet();
```

DVWA-master/vulnerabilities/sqli/source/low.php

```
5 // Get input
  $id = $_REQUEST['id'];
```

HTML Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
<script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script> </head> <body class="home"> <div id="container">
<div id="header"> <img src="../../dvwa/images/logo.png" alt="Damn Vulnerable Web Application" /> </div> <div id="main_menu">
<div id="main_menu_padded"> <ul class="menuBlocks"><li onclick="window.location='../../vulnerabilities/captcha/'"
class="selected"><a href="../../logout">logout</a></li> </ul> </div> </div> <div id="main_body"> <div class="body_padded">
<h1>Vulnerability: SQL Injection</h1> <div class="warning">The PHP function "<em>Magic Quotes</em>" is enabled.</div><div
class="warning">The PHP function "<em>Safe mode</em>" is enabled.</div> <div class="vulnerable_code_area">Click <a href="#"
onClick="javascript:popUp('session-input.php');return false;">here to change your ID</a>. <br />First name: .+<br />Surname:
.+</pre> </div> <h2>More Information</h2> <ul> <li><a href="http://www.securiteam.com/securityreviews/5DP0N1P76E.html"
target="_blank">http://www.securiteam.com/securityreviews/5DP0N1P76E.html</a></li> <li><a
href="https://en.wikipedia.org/wiki/SQL_injection" target="_blank">https://en.wikipedia.org/wiki/SQL_injection</a></li> <li>
<a href="http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/" target="_blank">http://ferruh.mavituna.com/sql-injection-
```

```
cheatsheet-oku/</a></li> <li><a href="http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet"
target="_blank">http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet</a></li> <li><a
href="https://www.owasp.org/index.php/SQL_Injection" target="_blank">https://www.owasp.org/index.php/SQL_Injection</a></li>
<li><a href="http://bobby-tables.com/" target="_blank">http://bobby-tables.com/</a></li> </ul> </div> <br /><br /> 0 </div>
<div class="clear"> </div> <div id="system_info"> <input type="button" value="View Source" class="popup_button"
onClick="javascript:popUp( '../../vulnerabilities/view_source.php?id=P_29_&security=U_28_' )"> <div align="left">
<em>Username:</em> Array<br /><em>Security Level:</em> low<br /><em>PHPIDS:</em> enabled</div> </div> <div id="footer">
<p>Damn Vulnerable Web Application (DVWA) v1.10 *Development*</p> </div> </div> </body> </html>
```

Issue #375388 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 278

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 278
Sink: echo
Source: _GET
Taint: HTTP

Code Summary

The GET parameter "username" is received in line 5 of DVWA-master/vulnerabilities/brute/source/low.php.

It is concatenated into html markup in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/brute/index.php

```
4   require_once DVWA_WEB_PAGE_TO_ROOT . 'dvwa/includes/dvwaPage.inc.php';
⋮
8   $page = dvwaPageNewGrab();
9   $page['title'] = 'Vulnerability: Brute Force' . $page['title_separator'] . $page['title'];
10  $page['page_id'] = 'brute';
11  $page['help_button'] = 'brute';
12  $page['source_button'] = 'brute';
⋮
29  $method = 'POST';
⋮
33  require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/brute/source/{$vulnerabilityFile}";
⋮
35  $page['body'] .= "\r\n<div class=\"body_padded\">\r\n\t<h1>Vulnerability: Brute Force</h1>\r\n\r\n\t<div class=\"vulnera
    ble_code_area\">\r\n\t\t<h2>Login</h2>\r\n\r\n\t\t<form action=\"#\" method=\"{$method}\">\r\n\t\t\tUsername:<br />\r\n\
    t\t\t<input type=\"text\" name=\"username\"><br />\r\n\t\t\tPassword:<br />\r\n\t\t\t<input type=\"password\" AUTOCOMPLE
    TE=\"off\" name=\"password\"><br />\r\n\t\t\t<br />\r\n\t\t\t<input type=\"submit\" value=\"Login\" name=\"Login\">\n";
⋮
51  $page['body'] .= "\t\t\t" . tokenField();
⋮
53  $page['body'] .= "\r\n\t\t</form>\r\n\t\t{$html}\r\n\t</div>\r\n\r\n\t<h2>More Information</h2>\r\n\t<ul>\r\n\t\t<li>" .
    dvwaExternalLinkUrlGet('https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)') . "</li>\r\n\t\t<li>" .
    dvwaExternalLinkUrlGet('http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password') . "
    </li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-win
    dows.html') . "</li>\r\n\t</ul>\r\n</div>\n";
⋮
66  dvwaHtmlEcho($page);
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../../');
⋮
183 function dvwaHtmlEcho($pPage)
184 {
185 $menuBlocks = array();
⋮
186 $menuBlocks['home'] = array();
⋮
188 $menuBlocks['home'][] = array('id' => 'home', 'name' => 'Home', 'url' => '.');
189 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
190 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup / Reset DB', 'url' => 'setup.php');
⋮
193 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup DVWA', 'url' => 'setup.php');
194 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
⋮
198 $menuBlocks['vulnerabilities'] = array();
199 $menuBlocks['vulnerabilities'][] = array('id' => 'brute', 'name' => 'Brute Force', 'url' => 'vulnerabilities/brute/');
200 $menuBlocks['vulnerabilities'][] = array('id' => 'exec', 'name' => 'Command Injection', 'url' => 'vulnerabilities/exec/
    ');
201 $menuBlocks['vulnerabilities'][] = array('id' => 'csrf', 'name' => 'CSRF', 'url' => 'vulnerabilities/csrf/');
202 $menuBlocks['vulnerabilities'][] = array('id' => 'fi', 'name' => 'File Inclusion', 'url' => 'vulnerabilities/fi/.?page=
    include.php');
203 $menuBlocks['vulnerabilities'][] = array('id' => 'upload', 'name' => 'File Upload', 'url' => 'vulnerabilities/upload/')
```

```
;
204 $menuBlocks['vulnerabilities'][] = array('id' => 'captcha', 'name' => 'Insecure CAPTCHA', 'url' => 'vulnerabilities/captcha/');
205 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli', 'name' => 'SQL Injection', 'url' => 'vulnerabilities/sqli/');
206 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli_blind', 'name' => 'SQL Injection (Blind)', 'url' => 'vulnerabilities/sqli_blind/');
207 $menuBlocks['vulnerabilities'][] = array('id' => 'weak_id', 'name' => 'Weak Session IDs', 'url' => 'vulnerabilities/weak_id/');
208 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_d', 'name' => 'XSS (DOM)', 'url' => 'vulnerabilities/xss_d/');
209 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_r', 'name' => 'XSS (Reflected)', 'url' => 'vulnerabilities/xss_r/');
210 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_s', 'name' => 'XSS (Stored)', 'url' => 'vulnerabilities/xss_s/');
⋮
213 $menuBlocks['meta'] = array();
⋮
215 $menuBlocks['meta'][] = array('id' => 'security', 'name' => 'DVWA Security', 'url' => 'security.php');
216 $menuBlocks['meta'][] = array('id' => 'phpinfo', 'name' => 'PHP Info', 'url' => 'phpinfo.php');
⋮
218 $menuBlocks['meta'][] = array('id' => 'about', 'name' => 'About', 'url' => 'about.php');
⋮
221 $menuBlocks['logout'] = array();
222 $menuBlocks['logout'][] = array('id' => 'logout', 'name' => 'Logout', 'url' => 'logout.php');
⋮
225 $menuHtml = '';
⋮
227 foreach ($menuBlocks as $menuBlock) {
⋮
229 foreach ($menuBlock as $menuItem) {
230 $selectedClass = $menuItem['id'] == $pPage['page_id'] ? 'selected' : '';
231 $fixedUrl = DVWA_WEB_PAGE_TO_ROOT . $menuItem['url'];
232 $menuBlockHtml .= "<li onclick=\"window.location='{$fixedUrl}'\" class=\"{$selectedClass}\"><a href=\"{$fixedUrl}\">{$menuItem['name']}</a></li>\n";
⋮
234 $menuHtml .= "<ul class=\"menuBlocks\">{$menuBlockHtml}</ul>";
⋮
241 $securityLevelHtml = 'low';
⋮
244 $securityLevelHtml = 'medium';
⋮
247 $securityLevelHtml = 'high';
⋮
250 $securityLevelHtml = 'impossible';
⋮
255 // -- END (security cookie)
$phpIdsHtml = '<em>PHPIDS:</em> ' . (dvwaPhpIdsIsEnabled() ? 'enabled' : 'disabled');
⋮
259 $messagesHtml;
⋮
263 $systemInfoHtml = "";
⋮
265 $systemInfoHtml = "<div align=\"left\">{$userInfoHtml}<br /><em>Security Level:</em> {$securityLevelHtml}<br />{$phpIdsHtml}</div>";
⋮
267 $systemInfoHtml = dvwaButtonSourceHtmlGet($pPage['source_button']) . " {$systemInfoHtml}";
⋮
270 $systemInfoHtml = dvwaButtonHelpHtmlGet($pPage['help_button']) . " {$systemInfoHtml}";
⋮
278 // Date in the past
echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\" />\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\t\t<link rel=\"stylesheet\" type=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/main.css\" />\r\n\r\n\t\t<link rel=\"icon\" type=\"\\image/ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t\t<script type=\"text/javascript\" src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/js/dvwaPage.js\"></script>\r\n\r\n\t</head>\r\n\r\n\t<body class=\"home\">\r\n\t\t<div id=\"container\">\r\n\r\n\t\t<div id=\"header\">\r\n\r\n\t\t<img src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/images/logo.png\" alt=\"Damn Vulnerable Web Application\" />\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_menu\">\r\n\r\n\t\t<div id=\"main_menu_padded\">\r\n\t\t{$menuHtml}\r\n\t\t</div>\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_body\">\r\n\r\n\t\t{$pPage['body']}\r\n\t\t<br /><br />\r\n\t\t{$messagesHtml}\r\n\r\n\t\t</div>\r\n\r\n\t\t<div class=\"clear\">\r\n\t\t</div>\r\n\r\n\t\t<div id=\"system_info\">\r\n\t\t{$systemInfoHtml}\r\n\t\t</div>\r\n\r\n\t\t<div id=\"footer\">\r\n\r\n\t\t<p>Damn Vulnerable Web Application (DVWA) v" . dvwaVersionGet() . "</p>\r\n\r\n\t</div>\r\n\r\n</body>\r\n</html>";
⋮
427 $security = dvwaSecurityLevelGet();
⋮
433 $security = dvwaSecurityLevelGet();
```

DVWA-master/vulnerabilities/brute/source/low.php

```
5 // Get username
```

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

HTML Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
<script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script> </head> <body class="home"> <div id="container">
<div id="header"> <img src="../../dvwa/images/logo.png" alt="Damn Vulnerable Web Application" /> </div> <div id="main_menu">
<div id="main_menu_padded"> <ul class="menuBlocks"><li onclick="window.location='../../vulnerabilities/captcha/'"
class="selected"><a href="../../logout">logout</a></li> </ul> </div> </div> <div id="main_body"> <div class="body_padded">
<h1>Vulnerability: Brute Force</h1> <div class="vulnerable_code_area"> <h2>Login</h2> <form action="#" method="GET">
Username:<br /> <input type="text" name="username"><br /> Password:<br /> <input type="password" AUTOCOMPLETE="off"
name="password"><br /> <br /> <input type="submit" value="Login" name="Login"> <input type='hidden' name='user_token'
value='.+' /> </form> .+<p>Welcome to the password protected area $_GET['username']</p><img src=".+" /><p>Welcome to the
password protected area $_GET['username']</p><img src=".+" /><p>Welcome to the password protected area $_GET['username']</p>
<img src=".+" /><p>Welcome to the password protected area <em>$_POST['username']</em></p><img src=".+" /><p>
<em>Warning</em>: Someone might of been brute forcing your account.</p><p>Number of login attempts: <em>1</em>.<br />Last
login attempt was at: <em>1</em>.</p> </div> <h2>More Information</h2> <ul> <li><a
href="https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)"
target="_blank">https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)</a></li> <li><a
href="http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password"
target="_blank">http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password</a></li> <li><a
href="http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html"
target="_blank">http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html</a></li> </ul> </div>
<br /><br /> 0 </div> <div class="clear"> </div> <div id="system_info"> <input type="button" value="View Source"
class="popup_button" onClick="javascript:popUp( '../../vulnerabilities/view_source.php?id=P_29_&security=U_28_' )"> <div
align="left"><em>Username:</em> Array<br /><em>Security Level:</em> low<br /><em>PHPIDS:</em> enabled</div> </div> <div
id="footer"> <p>Damn Vulnerable Web Application (DVWA) v1.10 *Development*</p> </div> </div> </body> </html>
```

Issue #375394 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 278

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 278
Sink: echo
Source:
Taint: Multi-Step

Code Summary

A SQL injection allows to taint column data of table "users.first_name" that is received in line 15 of DVWA-master/vulnerabilities/sqli/source/impossible.php.

It is concatenated into html markup in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/sqli/index.php

```
4  require_once DVWA_WEB_PAGE_TO_ROOT . 'dvwa/includes/dvwaPage.inc.php';
8  $page = dvwaPageNewGrab();
9  $page['title'] = 'Vulnerability: SQL Injection' . $page['title_separator'] . $page['title'];
10 $page['page_id'] = 'sqli';
11 $page['help_button'] = 'sqli';
12 $page['source_button'] = 'sqli';
16 $method = 'GET';
24 $method = 'POST';
34 require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/sqli/source/{$vulnerabilityFile}";
37 // Is PHP function magic_quotee enabled?
   $WarningHtml = '';
39 $WarningHtml .= "<div class=\"warning\">The PHP function \"<em>Magic Quotes</em>\" is enabled.</div>";
43 $WarningHtml .= "<div class=\"warning\">The PHP function \"<em>Safe mode</em>\" is enabled.</div>";
46 $page['body'] .= "\r\n<div class=\"body_padded\">\r\n\t<h1>Vulnerability: SQL Injection</h1>\r\n\r\n\t{$WarningHtml}\r\n
   \r\n\t<div class=\"vulnerable_code_area\">";
54 $page['body'] .= "Click <a href=\"#\" onClick=\"javascript:popUp('session-input.php');return false;\">here to change you
   r ID</a>.";
57 $page['body'] .= "\r\n\t\t<form action=\"#\" method=\"{$method}\">\r\n\t\t\t<p>\r\n\t\t\t\tUser ID:";
62 $page['body'] .= "\n\t\t\t\t<select name=\"id\">";
64 for ($i = 1; $i < $number_of_rows + 1; $i++) {
```

```
65  $page['body'] .= "</select>";
68  $page['body'] .= "\n\t\t\t\t<input type=\"text\" size=\"15\" name=\"id\">";
70  $page['body'] .= "\n\t\t\t\t<input type=\"submit\" name=\"Submit\" value=\"Submit\">\r\n\t\t\t</p>\n";
74  $page['body'] .= "\t\t\t" . tokenField();
76  $page['body'] .= "\r\n\t\t</form>";
    $page['body'] .= "\r\n\t\t{$html}\r\n\t</div>\r\n\r\n\t<h2>More Information</h2>\r\n\t<ul>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://www.securiteam.com/securityreviews/5DP0N1P76E.html') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https://en.wikipedia.org/wiki/SQL_injection') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https://www.owasp.org/index.php/SQL_Injection') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://bobby-tables.com/') . "</li>\r\n\t</ul>\r\n</div>\n";
94  dvwaHtmlEcho($page);
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../../');
183 function dvwaHtmlEcho($pPage)
184 {
186 $menuBlocks = array();
188 $menuBlocks['home'] = array();
189 $menuBlocks['home'][] = array('id' => 'home', 'name' => 'Home', 'url' => '.');
190 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
193 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup / Reset DB', 'url' => 'setup.php');
194 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup DVWA', 'url' => 'setup.php');
198 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
199 $menuBlocks['vulnerabilities'] = array();
200 $menuBlocks['vulnerabilities'][] = array('id' => 'brute', 'name' => 'Brute Force', 'url' => 'vulnerabilities/brute/');
201 $menuBlocks['vulnerabilities'][] = array('id' => 'exec', 'name' => 'Command Injection', 'url' => 'vulnerabilities/exec/');
202 $menuBlocks['vulnerabilities'][] = array('id' => 'csrf', 'name' => 'CSRF', 'url' => 'vulnerabilities/csrf/');
203 $menuBlocks['vulnerabilities'][] = array('id' => 'fi', 'name' => 'File Inclusion', 'url' => 'vulnerabilities/fi/.?page=include.php');
204 $menuBlocks['vulnerabilities'][] = array('id' => 'upload', 'name' => 'File Upload', 'url' => 'vulnerabilities/upload/');
205 $menuBlocks['vulnerabilities'][] = array('id' => 'captcha', 'name' => 'Insecure CAPTCHA', 'url' => 'vulnerabilities/captcha/');
206 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli', 'name' => 'SQL Injection', 'url' => 'vulnerabilities/sqli/');
207 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli_blind', 'name' => 'SQL Injection (Blind)', 'url' => 'vulnerabilities/sqli_blind/');
208 $menuBlocks['vulnerabilities'][] = array('id' => 'weak_id', 'name' => 'Weak Session IDs', 'url' => 'vulnerabilities/weak_id/');
209 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_d', 'name' => 'XSS (DOM)', 'url' => 'vulnerabilities/xss_d/');
210 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_r', 'name' => 'XSS (Reflected)', 'url' => 'vulnerabilities/xss_r/');
213 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_s', 'name' => 'XSS (Stored)', 'url' => 'vulnerabilities/xss_s/');
215 $menuBlocks['meta'] = array();
216 $menuBlocks['meta'][] = array('id' => 'security', 'name' => 'DVWA Security', 'url' => 'security.php');
218 $menuBlocks['meta'][] = array('id' => 'phpinfo', 'name' => 'PHP Info', 'url' => 'phpinfo.php');
221 $menuBlocks['meta'][] = array('id' => 'about', 'name' => 'About', 'url' => 'about.php');
222 $menuBlocks['logout'] = array();
225 $menuBlocks['logout'][] = array('id' => 'logout', 'name' => 'Logout', 'url' => 'logout.php');
227 $menuHtml = '';
229 foreach ($menuBlocks as $menuBlock) {
230 foreach ($menuBlock as $menuItem) {
231 $selectedClass = $menuItem['id'] == $pPage['page_id'] ? 'selected' : '';
232 $fixedUrl = DVWA_WEB_PAGE_TO_ROOT . $menuItem['url'];
234 $menuBlockHtml .= "<li onclick=\"window.location='{$fixedUrl}'\" class=\"{$selectedClass}\"><a href=\"{$fixedUrl}\">{$menuItem['name']}</a></li>\n";
    $menuHtml .= "<ul class=\"menuBlocks\">{$menuBlockHtml}</ul>";
```

```
241 $securityLevelHtml = 'low';
    ⋮
244 $securityLevelHtml = 'medium';
    ⋮
247 $securityLevelHtml = 'high';
    ⋮
250 $securityLevelHtml = 'impossible';
    ⋮
255 // -- END (security cookie)
    $phpIdsHtml = '<em>PHPIDS:</em> ' . (dvwaPhpIdsIsEnabled() ? 'enabled' : 'disabled');
    ⋮
259 $messagesHtml;
    ⋮
263 $systemInfoHtml = "";
    ⋮
265 $systemInfoHtml = "<div align=\"left\">{$userInfoHtml}<br /><em>Security Level:</em> {$securityLevelHtml}<br />{$phpIds
    Html}</div>";
    ⋮
267 $systemInfoHtml = dvwaButtonSourceHtmlGet($pPage['source_button']) . " {$systemInfoHtml}";
    ⋮
270 $systemInfoHtml = dvwaButtonHelpHtmlGet($pPage['help_button']) . " {$systemInfoHtml}";
    ⋮
    // Date in the past
    echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dt
    d\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv=\"Content-Type\" cont
    ent=\"text/html; charset=UTF-8\" />\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\t\t<link rel=\"stylesheet\" typ
    e=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/main.css\" />\r\n\r\n\t\t<link rel=\"icon\" type=\"\\image/
    ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t\t<script type=\"text/javascript\" src=\"" . DVWA_W
278 EB_PAGE_TO_ROOT . "dvwa/js/dvwaPage.js\"></script>\r\n\r\n\t</head>\r\n\r\n<body class=\"home\">\r\n\t<div id=\"con
    tainer\">\r\n\r\n\t<div id=\"header\">\r\n\r\n\t\t<img src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/images/logo.png\
    " alt=\"Damn Vulnerable Web Application\" />\r\n\r\n\t</div>\r\n\r\n<div id=\"main_menu\">\r\n\r\n\t<di
    v id=\"main_menu_padded\">\r\n\t{$menuHtml}\r\n\t</div>\r\n\r\n\t<div id=\"main_bod
    y\">\r\n\r\n\t{$pPage['body']}\r\n\t<br /><br />\r\n\t{$messagesHtml}\r\n\r\n\t</div>\r\n\r\n\t
    \t<div class=\"clear\">\r\n\t</div>\r\n\r\n\t<div id=\"system_info\">\r\n\t{$systemInfoHtml}\r\n\t</d
    iv>\r\n\r\n\t<div id=\"footer\">\r\n\r\n\t<p>Damn Vulnerable Web Application (DVWA) v" . dvwaVersionGet() . "
    </p>\r\n\r\n\t</div>\r\n\r\n</div>\r\n\r\n</body>\r\n\r\n</html>";
    ⋮
427 $security = dvwaSecurityLevelGet();
    ⋮
433 $security = dvwaSecurityLevelGet();
```

DVWA-master/vulnerabilities/sqli/source/impossible.php

```
15 $data->execute();
```

HTML Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT `first_name` FROM `users`
```

Issue #375395 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 278

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 278
Sink: echo
Source:
Taint: Multi-Step

Code Summary

A SQL injection allows to taint column data of table "users.last_name" that is received in line 15 of DVWA-master/vulnerabilities/sqli/source/impossible.php.

It is concatenated into html markup in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/sqli/index.php

```
4  require_once DVWA_WEB_PAGE_TO_ROOT . 'dvwa/includes/dvwaPage.inc.php';
   ⋮
8  $page = dvwaPageNewGrab();
9  $page['title'] = 'Vulnerability: SQL Injection' . $page['title_separator'] . $page['title'];
10 $page['page_id'] = 'sqli';
11 $page['help_button'] = 'sqli';
12 $page['source_button'] = 'sqli';
   ⋮
16 $method = 'GET';
   ⋮
```

```
24  $method = 'POST';
⋮
34  require_once DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/sqli/source/{$vulnerabilityFile}";
⋮
37  // Is PHP function magic_quotee enabled?
    $WarningHtml = '';
⋮
39  $WarningHtml .= "<div class=\"warning\">The PHP function \"<em>Magic Quotes</em>\" is enabled.</div>";
⋮
43  $WarningHtml .= "<div class=\"warning\">The PHP function \"<em>Safe mode</em>\" is enabled.</div>";
⋮
46  $page['body'] .= "\r\n<div class=\"body_padded\">\r\n\t<h1>Vulnerability: SQL Injection</h1>\r\n\r\n\t{$WarningHtml}\r\n
    \r\n\t<div class=\"vulnerable_code_area\">";
⋮
54  $page['body'] .= "Click <a href=\"#\" onClick=\"javascript:popUp('session-input.php');return false;\">here to change you
    r ID</a>.";
⋮
57  $page['body'] .= "\r\n\t\t<form action=\"#\" method=\"{$method}\">\r\n\t\t\t<p>\r\n\t\t\t\tUser ID:";
⋮
62  $page['body'] .= "\n\t\t\t\t<select name=\"id\">";
⋮
64  for ($i = 1; $i < $number_of_rows + 1; $i++) {
65  $page['body'] .= "</select>";
⋮
68  $page['body'] .= "\n\t\t\t\t<input type=\"text\" size=\"15\" name=\"id\">";
⋮
70  $page['body'] .= "\n\t\t\t\t<input type=\"submit\" name=\"Submit\" value=\"Submit\">\r\n\t\t\t</p>\n";
⋮
74  $page['body'] .= "\t\t\t" . tokenField();
⋮
76  $page['body'] .= "\r\n\t\t</form>";
⋮
    $page['body'] .= "\r\n\t\t{$html}\r\n\t</div>\r\n\r\n\t<h2>More Information</h2>\r\n\t<ul>\r\n\t\t<li>" . dvwaExternalLi
    nkUrlGet('http://www.securiteam.com/securityreviews/5DP0N1P76E.html') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('ht
    tps://en.wikipedia.org/wiki/SQL_injection') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://ferruh.mavituna.com/s
79  ql-injection-cheatsheet-oku/') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://pentestmonkey.net/cheat-sheet/sql-
    injection/mysql-sql-injection-cheat-sheet') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https://www.owasp.org/index.
    php/SQL_Injection') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://bobby-tables.com/') . "</li>\r\n\t</ul>\r\n</
    div>\n";
⋮
94  dvwaHtmlEcho($page);
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../../');
⋮
183 function dvwaHtmlEcho($pPage)
184 {
184 $menuBlocks = array();
⋮
186 $menuBlocks['home'] = array();
⋮
188 $menuBlocks['home'][] = array('id' => 'home', 'name' => 'Home', 'url' => '.');
189 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
190 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup / Reset DB', 'url' => 'setup.php');
⋮
193 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup DVWA', 'url' => 'setup.php');
194 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
⋮
198 $menuBlocks['vulnerabilities'] = array();
199 $menuBlocks['vulnerabilities'][] = array('id' => 'brute', 'name' => 'Brute Force', 'url' => 'vulnerabilities/brute/');
200 $menuBlocks['vulnerabilities'][] = array('id' => 'exec', 'name' => 'Command Injection', 'url' => 'vulnerabilities/exec/
    ');
201 $menuBlocks['vulnerabilities'][] = array('id' => 'csrf', 'name' => 'CSRF', 'url' => 'vulnerabilities/csrf/');
202 $menuBlocks['vulnerabilities'][] = array('id' => 'fi', 'name' => 'File Inclusion', 'url' => 'vulnerabilities/fi/.?page=
    include.php');
203 $menuBlocks['vulnerabilities'][] = array('id' => 'upload', 'name' => 'File Upload', 'url' => 'vulnerabilities/upload/')
    ;
204 $menuBlocks['vulnerabilities'][] = array('id' => 'captcha', 'name' => 'Insecure CAPTCHA', 'url' => 'vulnerabilities/cap
    tcha/');
205 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli', 'name' => 'SQL Injection', 'url' => 'vulnerabilities/sqli/');
206 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli_blind', 'name' => 'SQL Injection (Blind)', 'url' => 'vulnerabili
    ties/sqli_blind/');
207 $menuBlocks['vulnerabilities'][] = array('id' => 'weak_id', 'name' => 'Weak Session IDs', 'url' => 'vulnerabilities/wea
    k_id/');
208 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_d', 'name' => 'XSS (DOM)', 'url' => 'vulnerabilities/xss_d/');
209 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_r', 'name' => 'XSS (Reflected)', 'url' => 'vulnerabilities/xss_r/
    ');
210 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_s', 'name' => 'XSS (Stored)', 'url' => 'vulnerabilities/xss_s/');
```

```
213 $menuBlocks['meta'] = array();
215 $menuBlocks['meta'][] = array('id' => 'security', 'name' => 'DVWA Security', 'url' => 'security.php');
216 $menuBlocks['meta'][] = array('id' => 'phpinfo', 'name' => 'PHP Info', 'url' => 'phpinfo.php');
218 $menuBlocks['meta'][] = array('id' => 'about', 'name' => 'About', 'url' => 'about.php');
221 $menuBlocks['logout'] = array();
222 $menuBlocks['logout'][] = array('id' => 'logout', 'name' => 'Logout', 'url' => 'logout.php');
225 $menuHtml = '';
227 foreach ($menuBlocks as $menuBlock) {
229 foreach ($menuBlock as $menuItem) {
230 $selectedClass = $menuItem['id'] == $pPage['page_id'] ? 'selected' : '';
231 $fixedUrl = DVWA_WEB_PAGE_TO_ROOT . $menuItem['url'];
232 $menuBlockHtml .= "<li onclick=\"window.location='{$fixedUrl}'\" class=\"{$selectedClass}\"><a href=\"{$fixedUrl}\">{$menuItem['name']}</a></li>\n";
234 $menuHtml .= "<ul class=\"menuBlocks\">{$menuBlockHtml}</ul>";
241 $securityLevelHtml = 'low';
244 $securityLevelHtml = 'medium';
247 $securityLevelHtml = 'high';
250 $securityLevelHtml = 'impossible';
255 // -- END (security cookie)
    $phpIdsHtml = '<em>PHPIDS:</em> ' . (dvwaPhpIdsIsEnabled() ? 'enabled' : 'disabled');
259 $messagesHtml;
263 $systemInfoHtml = "";
265 $systemInfoHtml = "<div align=\"left\">{$userInfoHtml}<br /><em>Security Level:</em> {$securityLevelHtml}<br />{$phpIdsHtml}</div>";
267 $systemInfoHtml = dvwaButtonSourceHtmlGet($pPage['source_button']) . " {$systemInfoHtml}";
270 $systemInfoHtml = dvwaButtonHelpHtmlGet($pPage['help_button']) . " {$systemInfoHtml}";
278 // Date in the past
    echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\" />\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\t\t<link rel=\"stylesheet\" type=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/main.css\" />\r\n\r\n\t\t<link rel=\"icon\" type=\"\\image/ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t\t<script type=\"text/javascript\" src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/js/dvwaPage.js\"></script>\r\n\r\n\t</head>\r\n\r\n\t<body class=\"home\">\r\n\t\t<div id=\"container\">\r\n\r\n\t\t<div id=\"header\">\r\n\r\n\t\t<img src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/images/logo.png\" alt=\"Damn Vulnerable Web Application\" />\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_menu\">\r\n\r\n\t\t<div id=\"main_menu_padded\">\r\n\t\t{$menuHtml}\r\n\t\t</div>\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_body\">\r\n\r\n\t\t{$pPage['body']}\r\n\t\t<br /><br />\r\n\t\t{$messagesHtml}\r\n\r\n\t\t</div>\r\n\r\n\t\t<div class=\"clear\">\r\n\t\t</div>\r\n\r\n\t\t<div id=\"system_info\">\r\n\t\t{$systemInfoHtml}\r\n\t\t</div>\r\n\r\n\t\t<div id=\"footer\">\r\n\r\n\t\t<p>Damn Vulnerable Web Application (DVWA) v" . dvwaVersionGet() . "</p>\r\n\r\n\t\t</div>\r\n\r\n\t</body>\r\n\r\n</html>";
427 $security = dvwaSecurityLevelGet();
433 $security = dvwaSecurityLevelGet();
```

DVWA-master/vulnerabilities/sqli/source/impossible.php

```
15 $data->execute();
```

HTML Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
SELECT `last_name` FROM `users`
```

3.6.2. Cross-Site Scripting (eventhandler)

CWE: 83 **OWASP Top 10:** A3 **SANS 25 Rank:** 4 **PCI DSS:** 6.5.7 **Severity:** Medium

The detected injection occurs within an eventhandler attribute. Data within this attribute's value is executed as JavaScript code. Thus, an attacker can inject malicious JavaScript code for execution.

Issue #375347 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 278

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 278
Sink: echo
Source: _COOKIE
Taint: HTTP

Code Summary

The cookie "security" is received in function dvwaSecurityLevelGet() in line 136 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

It is concatenated into html markup in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../../');
    :
183 function dvwaHtmlEcho($pPage)
184 {
184 $menuBlocks = array();
    :
186 $menuBlocks['home'] = array();
    :
188 $menuBlocks['home'][] = array('id' => 'home', 'name' => 'Home', 'url' => '.');
189 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
190 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup / Reset DB', 'url' => 'setup.php');
    :
193 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup DVWA', 'url' => 'setup.php');
194 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
    :
198 $menuBlocks['vulnerabilities'] = array();
199 $menuBlocks['vulnerabilities'][] = array('id' => 'brute', 'name' => 'Brute Force', 'url' => 'vulnerabilities/brute/');
200 $menuBlocks['vulnerabilities'][] = array('id' => 'exec', 'name' => 'Command Injection', 'url' => 'vulnerabilities/exec/
    ');
201 $menuBlocks['vulnerabilities'][] = array('id' => 'csrf', 'name' => 'CSRF', 'url' => 'vulnerabilities/csrf/');
202 $menuBlocks['vulnerabilities'][] = array('id' => 'fi', 'name' => 'File Inclusion', 'url' => 'vulnerabilities/fi/.?page=
    include.php');
203 $menuBlocks['vulnerabilities'][] = array('id' => 'upload', 'name' => 'File Upload', 'url' => 'vulnerabilities/upload/')
    ;
204 $menuBlocks['vulnerabilities'][] = array('id' => 'captcha', 'name' => 'Insecure CAPTCHA', 'url' => 'vulnerabilities/cap
    tcha/');
205 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli', 'name' => 'SQL Injection', 'url' => 'vulnerabilities/sqli/');
206 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli_blind', 'name' => 'SQL Injection (Blind)', 'url' => 'vulnerabili
    ties/sqli_blind/');
207 $menuBlocks['vulnerabilities'][] = array('id' => 'weak_id', 'name' => 'Weak Session IDs', 'url' => 'vulnerabilities/wea
    k_id/');
208 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_d', 'name' => 'XSS (DOM)', 'url' => 'vulnerabilities/xss_d/');
209 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_r', 'name' => 'XSS (Reflected)', 'url' => 'vulnerabilities/xss_r/
    ');
210 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_s', 'name' => 'XSS (Stored)', 'url' => 'vulnerabilities/xss_s/');
    :
213 $menuBlocks['meta'] = array();
    :
215 $menuBlocks['meta'][] = array('id' => 'security', 'name' => 'DVWA Security', 'url' => 'security.php');
216 $menuBlocks['meta'][] = array('id' => 'phpinfo', 'name' => 'PHP Info', 'url' => 'phpinfo.php');
    :
218 $menuBlocks['meta'][] = array('id' => 'about', 'name' => 'About', 'url' => 'about.php');
    :
221 $menuBlocks['logout'] = array();
222 $menuBlocks['logout'][] = array('id' => 'logout', 'name' => 'Logout', 'url' => 'logout.php');
    :
225 $menuHtml = '';
    :
227 foreach ($menuBlocks as $menuBlock) {
    :
229 foreach ($menuBlock as $menuItem) {
230 $selectedClass = $menuItem['id'] == $pPage['page_id'] ? 'selected' : '';
231 $fixedUrl = DVWA_WEB_PAGE_TO_ROOT . $menuItem['url'];
232 $menuBlockHtml .= "<li onclick=\"window.location='{$fixedUrl}'\" class=\"{$selectedClass}\"><a href=\"{$fixedUrl}\">{$m
    enuItem['name']}</a></li>\n";
    :
234 $menuHtml .= "<ul class=\"menuBlocks\">{$menuBlockHtml}</ul>";
    :
```

```
241 $securityLevelHtml = 'low';
    :
244 $securityLevelHtml = 'medium';
    :
247 $securityLevelHtml = 'high';
    :
250 $securityLevelHtml = 'impossible';
    :
255 // -- END (security cookie)
    $phpIdsHtml = '<em>PHPIDS:</em> ' . (dvwaPhpIdsIsEnabled() ? 'enabled' : 'disabled');
    :
259 $messagesHtml;
    :
263 $systemInfoHtml = "";
    :
265 $systemInfoHtml = "<div align=\"left\">{$userInfoHtml}<br /><em>Security Level:</em> {$securityLevelHtml}<br />{$phpIds
    Html}</div>";
    :
267 $systemInfoHtml = dvwaButtonSourceHtmlGet($pPage['source_button']) . " {$systemInfoHtml}";
    :
270 $systemInfoHtml = dvwaButtonHelpHtmlGet($pPage['help_button']) . " {$systemInfoHtml}";
    :
    // Date in the past
    echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dt
    d\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv=\"Content-Type\" cont
    ent=\"text/html; charset=UTF-8\" />\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\t\t<link rel=\"stylesheet\" typ
    e=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/main.css\" />\r\n\r\n\t\t<link rel=\"icon\" type=\"\\image/
    ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t\t<script type=\"text/javascript\" src=\"" . DVWA_W
278 EB_PAGE_TO_ROOT . "dvwa/js/dvwaPage.js\"></script>\r\n\r\n\t</head>\r\n\r\n<body class=\"home\">\r\n\t<div id=\"con
    tainer\">\r\n\r\n\t\t<div id=\"header\">\r\n\r\n\t\t<img src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/images/logo.png\
    " alt=\"Damn Vulnerable Web Application\" />\r\n\r\n\t</div>\r\n\r\n\t<div id=\"main_menu\">\r\n\r\n\t<di
    v id=\"main_menu_padded\">\r\n\t{$menuHtml}\r\n\t</div>\r\n\r\n\t<div id=\"main_bod
    y\">\r\n\r\n\t{$pPage['body']}\r\n<br /><br />\r\n{$messagesHtml}\r\n\r\n</div>\r\n\r\n<div
    \t<div class=\"clear\">\r\n</div>\r\n\r\n<div id=\"system_info\">\r\n{$systemInfoHtml}\r\n</d
    iv>\r\n\r\n<div id=\"footer\">\r\n\r\n<p>Damn Vulnerable Web Application (DVWA) v" . dvwaVersionGet() . "
    </p>\r\n\r\n</div>\r\n\r\n</div>\r\n\r\n</body>\r\n</html>";
    :
427 $security = dvwaSecurityLevelGet();
    :
433 $security = dvwaSecurityLevelGet();
    :
136 $_COOKIE['security'];
    :
135 function dvwaSecurityLevelGet()
    {
```

JavaScript Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
<title>.+</title> <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" /> <link rel="icon"
type="\image/ico" href="../../favicon.ico" /> <script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script>
</head> <body class="home"> <div id="container"> <div id="header"> <img src="../../dvwa/images/logo.png" alt="Damn
Vulnerable Web Application" /> </div> <div id="main_menu"> <div id="main_menu_padded"> <ul class="menuBlocks"><li
onclick="window.location='../../logout'" class="selected"><a href="../../logout">logout</a></li> </ul> </div> </div> <div
id="main_body"> .+ <br /><br /> 0 </div> <div class="clear"> </div> <div id="system_info"> <input type="button" value="View
Help" class="popup_button" onClick="javascript:popUp( '../../vulnerabilities/view_help.php?
id=.+&security=$_COOKIE['security']' )"> <input type="button" value="View Source" class="popup_button"
onClick="javascript:popUp( '../../vulnerabilities/view_source.php?id=.+&security=$_COOKIE['security']' )"> <div
align="left"><em>Username:</em> Array<br /><em>Security Level:</em> low<br /><em>PHPIDS:</em> enabled</div> </div> <div
id="footer"> <p>Damn Vulnerable Web Application (DVWA) v1.10 *Development*</p> </div> </div> </body> </html>
```

Issue #375357 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 385

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 385
Sink: echo
Source: _GET
Taint: HTTP

Code Summary

The GET parameter "id" is received in line 11 of DVWA-master/vulnerabilities/view_source.php.

It is concatenated into html markup in function dvwaSourceHtmlEcho() in line 385 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in function dvwaSourceHtmlEcho() in line 385 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/view_source.php

```
8   $page = dvwaPageNewGrab();
9   $page['title'] .= 'Source' . $page['title_separator'] . $page['title'];
    :
11  $id = $_GET['id'];
    :
17  $vuln = 'File Inclusion';
    :
20  $vuln = 'Brute Force';
    :
23  $vuln = 'CSRF';
    :
26  $vuln = 'Command Injection';
    :
29  $vuln = 'SQL Injection';
    :
32  $vuln = 'SQL Injection (Blind)';
    :
35  $vuln = 'File Upload';
    :
38  $vuln = 'Reflected XSS';
    :
41  $vuln = 'Stored XSS';
    :
44  $vuln = 'Weak Session IDs';
    :
47  $vuln = "Unknown Vulnerability";
    :
50  $source = @file_get_contents(DVWA_WEB_PAGE_TO_ROOT . "vulnerabilities/{$id}/source/{$security}.php");
51  $source = str_replace(array('$html .='), array('echo'), $source);
    :
    $page['body'] .= "\r\n<div class=\"body_padded\">\r\n\t<h1>{$vuln} Source</h1>\r\n\r\n\t<div id=\"code\">\r\n\t\t<table
    width='100%' bgcolor='white' style=\"border:2px #C0C0C0 solid\">\r\n\t\t\t<tr>\r\n\t\t\t\t<td><div id=\"code\">" . highl
53  ight_string($source, true) . "</div></td>\r\n\t\t\t</tr>\r\n\t\t</table>\r\n\t</div>\r\n\t<br /> <br />\r\n\r\n\t<form>\
    r\n\t\t<input type=\"button\" value=\"Compare All Levels\" onclick=\"window.location.href='view_source_all.php?id={$id}'
    \">\r\n\t</form>\r\n</div>\n";
    :
71  dvwaSourceHtmlEcho($page);
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3   define('DVWA_WEB_PAGE_TO_ROOT', '../');
    :
379 function dvwaSourceHtmlEcho($pPage)
    {
    :
    // Date in the past
    echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dt
    d\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\r\n\t<meta http-equiv=\"Content-Type\"
385 content=\"text/html; charset=UTF-8\" />\r\n\r\n\t<title>{$pPage['title']}</title>\r\n\r\n\t<link rel=\"stylesheet\"
    type=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/source.css\" />\r\n\r\n\t<link rel=\"icon\" type=\"\\i
    mage/ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t</head>\r\n\r\n\t<body>\r\n\r\n\t<div id=\"c
    ontainer\">\r\n\r\n\t\t{$pPage['body']}\r\n\r\n\t</div>\r\n\r\n\t</body>\r\n\r\n</html>";
    :
427 $security = dvwaSecurityLevelGet();
    :
433 $security = dvwaSecurityLevelGet();
```

JavaScript Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
<input type="button" value="Compare All Levels" onclick="window.location.href='view_source_all.php?id=$_GET['id']'"> </form>
</div> </div> </body> </html>
```

3.7. CVE

CWE: **OWASP Top 10:** A9 **PCI DSS:** 6.2 **Severity:** Medium

The Common Vulnerabilities and Exposures (CVE) dictionary is an industry standard that references a collection of publicly known security vulnerabilities to an unique CVE identifier. Next to security issues added by the developer of a PHP code, it is possible that built-in features of the PHP core (written in the C language) are affected by security issues due to an outdated PHP version. A publicly known security vulnerability in the PHP core itself was found to be exploitable via the analyzed application. It is recommended to upgrade to the latest PHP version or to avoid the usage of the affected feature.

3.7.1. Use After Free

CWE: 416 **OWASP Top 10:** A9 **PCI DSS:** 6.2 **Severity:** Medium

In the selected PHP version, the affected PHP built-in feature is vulnerable to an use after free vulnerability. This security issue occurs when memory is referenced after is has been freed. An attacker can abuse this to execute arbitrary code or to crash the server.

Issue #375346 - DVWA-master/external/phpids/0.6/lib/IDS/Converter.php: 632

Path: DVWA-master/external/phpids/0.6/lib/IDS/Converter.php
Line: 632
Sink: unserialize
Source: _REQUEST
Taint: HTTP

Code Summary

This issue depends on the selected PHP version (7.0.0). The GET parameter is received in method IDS_Monitor::run() in line 236 of DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php.

The operation unserialize() has assigned CVEs and may be vulnerable in PHP 7.0.0. It is located in method IDS_Converter::runCentrifuge() in line 632 of DVWA-master/external/phpids/0.6/lib/IDS/Converter.php.

DVWA-master/external/phpids/0.6/lib/IDS/Monitor.php

```
    class IDS_Monitor
233 {
    public function run()
    {
⋮   ⋮
236 foreach ($this->request as $key => $value) {
237 $this->_iterate($key, $value);
⋮   ⋮
    class IDS_Monitor
253 {
    private function _iterate($key, $value)
    {
⋮   ⋮
259 $this->_detect($key, $value);
⋮   ⋮
    class IDS_Monitor
285 {
    private function _detect($key, $value)
    {
⋮   ⋮
308 list($key, $value) = $this->_purifyValues($key, $value);
⋮   ⋮
313 list($key, $value) = $this->_jsonDecodeValues($key, $value);
⋮   ⋮
322 // scan keys if activated via config
    $key = $this->scanKeys ? IDS_Converter::runAll($key) : $key;
⋮   ⋮
324 IDS_Converter::runCentrifuge($key, $this);
```

DVWA-master/external/phpids/0.6/lib/IDS/Converter.php

```
    class IDS_Converter
627 {
    public static function runCentrifuge($value, IDS_Monitor $monitor = null)
    {
⋮   ⋮
632 unserialize($value);
```

Info Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
$_REQUEST[.*]
```

3.8. Session Fixation

CWE: 384 **OWASP Top 10:** A2 **PCI DSS:** 6.5.10 **Severity:** Low

A session fixation vulnerability occurs when user input is used as a session token. By crafting a malicious link, an attacker can set a session token of his knowledge into the browser of a victim. When the victim authenticates himself over the web application, the attacker is in the possession of a valid token and can hijack the session of the victim. To prevent this, a fresh session token

should be generated by the web application before every login.

Issue #375373 - DVWA-master/vulnerabilities/sqli_blind/cookie-input.php: 12

Path: DVWA-master/vulnerabilities/sqli_blind/cookie-input.php
Line: 12
Sink: setcookie
Source: _POST
Taint: HTTP

Code Summary

The POST parameter "id" is received in line 12 of DVWA-master/vulnerabilities/sqli_blind/cookie-input.php.

The user-supplied data is then used unsanitized in the sensitive operation setcookie() in line 12 of DVWA-master/vulnerabilities/sqli_blind/cookie-input.php.

DVWA-master/vulnerabilities/sqli_blind/cookie-input.php

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

Cookie Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
$_POST['id']
```

3.9. Weak Cryptography

CWE: 310 **OWASP Top 10:** A6 **PCI DSS:** 6.5.3 **Severity:** Low

3.9.1. Weak Cryptography (broken algorithm)

CWE: 327 **OWASP Top 10:** A6 **SANS 25 Rank:** 19 **PCI DSS:** 6.5.3 **Severity:** Low

Issue #375341 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 535

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 535
Sink: md5
Source:
Taint: HTTP

Code Summary

The operation md5() is used incorrectly in function generateSessionToken() in line 535 of DVWA-master/dvwa/includes/dvwaPage.inc.php. This may result in weakened cryptography.

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
535 md5(uniqid());
```

Info Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
weak hash algorithm: MD5
```

Issue #375353 - DVWA-master/vulnerabilities/weak_id/source/impossible.php: 6

Path: DVWA-master/vulnerabilities/weak_id/source/impossible.php
Line: 6
Sink: sha1
Source:
Taint: HTTP

Code Summary

The operation sha1() is used incorrectly in line 6 of DVWA-master/vulnerabilities/weak_id/source/impossible.php. This may result in weakened cryptography.

DVWA-master/vulnerabilities/weak_id/source/impossible.php

```
6 sha1(mt_rand() . time() . "Impossible");
```

Info Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
weak hash algorithm: SHA1
```

Issue #375354 - DVWA-master/vulnerabilities/weak_id/source/high.php: 10

Path: DVWA-master/vulnerabilities/weak_id/source/high.php
Line: 10
Sink: md5
Source:
Taint: HTTP

Code Summary
The operation md5() is used incorrectly in line 10 of DVWA-master/vulnerabilities/weak_id/source/high.php. This may result in weakened cryptography.

DVWA-master/vulnerabilities/weak_id/source/high.php

```
10 | md5($_SESSION['last_session_id_high']);
```

Info Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
weak hash algorithm: MD5
```

Issue #375360 - DVWA-master/vulnerabilities/upload/source/impossible.php: 18

Path: DVWA-master/vulnerabilities/upload/source/impossible.php
Line: 18
Sink: md5
Source:
Taint: HTTP

Code Summary
The operation md5() is used incorrectly in line 18 of DVWA-master/vulnerabilities/upload/source/impossible.php. This may result in weakened cryptography.

DVWA-master/vulnerabilities/upload/source/impossible.php

```
18 | md5(uniqid() . $uploaded_name);
```

Info Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
weak hash algorithm: MD5
```

Issue #375361 - DVWA-master/vulnerabilities/upload/source/impossible.php: 20

Path: DVWA-master/vulnerabilities/upload/source/impossible.php
Line: 20
Sink: md5
Source:
Taint: HTTP

Code Summary
The operation md5() is used incorrectly in line 20 of DVWA-master/vulnerabilities/upload/source/impossible.php. This may result in weakened cryptography.

DVWA-master/vulnerabilities/upload/source/impossible.php

```
20 | md5(uniqid() . $uploaded_name);
```

Info Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
weak hash algorithm: MD5
```

Issue #375384 - DVWA-master/vulnerabilities/brute/source/low.php: 9

Path: DVWA-master/vulnerabilities/brute/source/low.php
Line: 9
Sink: md5
Source:
Taint: HTTP

Code Summary

The operation md5() is used incorrectly in line 9 of DVWA-master/vulnerabilities/brute/source/low.php. This may result in weakened cryptography.

DVWA-master/vulnerabilities/brute/source/low.php

```
9 | md5($pass);
```

Info Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
weak hash algorithm: MD5
```

Issue #375393 - DVWA-master/external/phpids/0.6/lib/IDS/Log/Email.php: 238

Path: DVWA-master/external/phpids/0.6/lib/IDS/Log/Email.php
Line: 238
Sink: md5
Source:
Taint: HTTP

Code Summary

The operation md5() is used incorrectly in method IDS_Log_Email::isSpamAttempt() in line 238 of DVWA-master/external/phpids/0.6/lib/IDS/Log/Email.php. This may result in weakened cryptography.

DVWA-master/external/phpids/0.6/lib/IDS/Log/Email.php

```
238 | md5($remoteAddr . $userAgent);
```

Info Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
weak hash algorithm: MD5
```

3.10. Cookie Misconfiguration

CWE: 494 **OWASP Top 10:** A5 **PCI DSS:** 6.5.10 **Severity:** Low

Cookies are often used to store a session identifier of the web application's state for a specific user. In order to harden against related attacks, such as session riding that can lead to the compromise of user data or the takeover of adminstrator sessions, a secure configuration of sensitive cookies is crucial. It has been detected that a new cookie is set insecure within the HTTP response header which can be improved by enabling additional flags.

3.10.1. Cookie Misconfiguration (secure flag)

CWE: 614 **OWASP Top 10:** A5 **PCI DSS:** 6.5.10 **Severity:** Low

The secure flag was not set for this cookie. This flag prevents that the cookie is transmitted in cleartext and is only send when HTTPS is used. For additional security, the secure flag can be set to TRUE in the 6th parameter.

Issue #375343 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 147

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 147
Sink: setcookie
Source:
Taint: HTTP

Code Summary

A cookie is set insecurely in function dvwaSecurityLevelSet() in line 147 of DVWA-master/dvwa/includes/dvwaPage.inc.php. Please refer to the context and description for further information.

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
147 | setcookie(session_name(), session_id(), null, '/', null, null, $httponly);
```

Flag Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
6th parameter not set to TRUE
```

Issue #375344 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 148

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 148
Sink: setcookie
Source:
Taint: HTTP

Code Summary
A cookie is set insecurely in function dvwaSecurityLevelSet() in line 148 of DVWA-master/dvwa/includes/dvwaPage.inc.php. Please refer to the context and description for further information.

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
148 setcookie('security', $pSecurityLevel, NULL, NULL, NULL, NULL, $httponly);
```

Flag Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
6th parameter not set to TRUE
```

Issue #375349 - DVWA-master/vulnerabilities/weak_id/source/medium.php: 7

Path: DVWA-master/vulnerabilities/weak_id/source/medium.php
Line: 7
Sink: setcookie
Source:
Taint: HTTP

Code Summary
A cookie is set insecurely in line 7 of DVWA-master/vulnerabilities/weak_id/source/medium.php. Please refer to the context and description for further information.

DVWA-master/vulnerabilities/weak_id/source/medium.php

```
7 setcookie("dvwaSession", $cookie_value);
```

Flag Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
missing 6th parameter
```

Issue #375351 - DVWA-master/vulnerabilities/weak_id/source/low.php: 11

Path: DVWA-master/vulnerabilities/weak_id/source/low.php
Line: 11
Sink: setcookie
Source:
Taint: HTTP

Code Summary
A cookie is set insecurely in line 11 of DVWA-master/vulnerabilities/weak_id/source/low.php. Please refer to the context and description for further information.

DVWA-master/vulnerabilities/weak_id/source/low.php

```
11 setcookie("dvwaSession", $cookie_value);
```

Flag Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
missing 6th parameter
```

Issue #375355 - DVWA-master/vulnerabilities/weak_id/source/high.php: 11

Path: DVWA-master/vulnerabilities/weak_id/source/high.php
Line: 11
Sink: setcookie
Source:
Taint: HTTP

Code Summary
A cookie is set insecurely in line 11 of DVWA-master/vulnerabilities/weak_id/source/high.php. Please refer to the context and description for further information.

DVWA-master/vulnerabilities/weak_id/source/high.php

```
11 setcookie("dvwaSession", $cookie_value, time() + 3600, "/vulnerabilities/weak_id/", $_SERVER['HTTP_HOST'], false, false)
;
```

Flag Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
6th parameter not set to TRUE
```

Issue #375374 - DVWA-master/vulnerabilities/sqli_blind/cookie-input.php: 12

Path: DVWA-master/vulnerabilities/sqli_blind/cookie-input.php
Line: 12
Sink: setcookie
Source:
Taint: HTTP

Code Summary
A cookie is set insecurely in line 12 of DVWA-master/vulnerabilities/sqli_blind/cookie-input.php. Please refer to the context and description for further information.

DVWA-master/vulnerabilities/sqli_blind/cookie-input.php

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

Flag Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
missing 6th parameter
```

3.10.2. Cookie Misconfiguration (path)

CWE: 287 **OWASP Top 10:** A5 **PCI DSS:** 6.5.10 **Severity:** Low

The specified path for the cookie is overly broad. In order to increase security, the validity of the cookie can be limited to a more specific application path in the 4th parameter.

Issue #375342 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 147

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 147
Sink: setcookie
Source:
Taint: HTTP

Code Summary
A cookie is set insecurely in function dvwaSecurityLevelSet() in line 147 of DVWA-master/dvwa/includes/dvwaPage.inc.php. Please refer to the context and description for further information.

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
147 setcookie(session_name(), session_id(), null, '/', null, null, $httponly);
```

Path Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
/
```

3.10.3. Cookie Misconfiguration (httpOnly flag)

CWE: 200 **OWASP Top 10:** A5 **PCI DSS:** 6.5.10 **Severity:** Low

The httpOnly flag was not set for this cookie. This flag prevents that the cookie is accessed by malicious scripts, for example during a Cross-Site Scripting attack. For additional security, the httpOnly flag can be set to TRUE in the 7th parameter.

Issue #375350 - DVWA-master/vulnerabilities/weak_id/source/medium.php: 7

Path: DVWA-master/vulnerabilities/weak_id/source/medium.php
Line: 7
Sink: setcookie
Source:
Taint: HTTP

Code Summary

A cookie is set insecurely in line 7 of DVWA-master/vulnerabilities/weak_id/source/medium.php. Please refer to the context and description for further information.

DVWA-master/vulnerabilities/weak_id/source/medium.php

```
7 setcookie("dvwaSession", $cookie_value);
```

Flag Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
missing 7th parameter
```

Issue #375352 - DVWA-master/vulnerabilities/weak_id/source/low.php: 11

Path: DVWA-master/vulnerabilities/weak_id/source/low.php
Line: 11
Sink: setcookie
Source:
Taint: HTTP

Code Summary

A cookie is set insecurely in line 11 of DVWA-master/vulnerabilities/weak_id/source/low.php. Please refer to the context and description for further information.

DVWA-master/vulnerabilities/weak_id/source/low.php

```
11 setcookie("dvwaSession", $cookie_value);
```

Flag Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
missing 7th parameter
```

Issue #375356 - DVWA-master/vulnerabilities/weak_id/source/high.php: 11

Path: DVWA-master/vulnerabilities/weak_id/source/high.php
Line: 11
Sink: setcookie
Source:
Taint: HTTP

Code Summary

A cookie is set insecurely in line 11 of DVWA-master/vulnerabilities/weak_id/source/high.php. Please refer to the context and description for further information.

DVWA-master/vulnerabilities/weak_id/source/high.php

```
11 setcookie("dvwaSession", $cookie_value, time() + 3600, "/vulnerabilities/weak_id/", $_SERVER['HTTP_HOST'], false, false)
   ;
```

Flag Context

The following snippet(s) do not represent actual code but the tainted markup context.

```
7th parameter not set to TRUE
```

Issue #375375 - DVWA-master/vulnerabilities/sqli_blind/cookie-input.php: 12

Path: DVWA-master/vulnerabilities/sqli_blind/cookie-input.php
Line: 12
Sink: setcookie
Source:
Taint: HTTP

Code Summary
A cookie is set insecurely in line 12 of DVWA-master/vulnerabilities/sqli_blind/cookie-input.php. Please refer to the context and description for further information.

DVWA-master/vulnerabilities/sqli_blind/cookie-input.php

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

Flag Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
missing 7th parameter
```

3.11. Information Leakage

CWE: 209 **OWASP Top 10:** A6 **PCI DSS:** 6.5.5 **Severity:** Low

An information leakage vulnerability occurs when confidential information about the web server's setup is leaked to the application's user. Although the issue might not be exploitable, it can help an attacker to prepare other attacks.

3.11.1. Information Leakage (System)

CWE: 209 **OWASP Top 10:** A6 **PCI DSS:** 6.5.5 **Severity:** Low

The affected code leaks information about the system that allows an attacker to learn about used software versions or installation paths.

Issue #375383 - DVWA-master/dvwa/includes/dvwaPage.inc.php: 278

Path: DVWA-master/dvwa/includes/dvwaPage.inc.php
Line: 278
Sink: echo
Source:
Taint: HTTP

Code Summary
The operation echo() leaks sensitive system information in function dvwaHtmlEcho() in line 278 of DVWA-master/dvwa/includes/dvwaPage.inc.php.

DVWA-master/vulnerabilities/captcha/index.php

```
3  define('DVWA_WEB_PAGE_TO_ROOT', '../../');
⋮
9  $page = dvwaPageNewGrab();
10 $page['title'] = 'Vulnerability: Insecure CAPTCHA' . $page['title_separator'] . $page['title'];
11 $page['page_id'] = 'captcha';
12 $page['help_button'] = 'captcha';
13 $page['source_button'] = 'captcha';
⋮
39 $WarningHtml = "<div class=\"warning\"><em>reCAPTCHA API key missing</em> from config file: " . realpath(getcwd() . DIR
   ECTORY_SEPARATOR . DVWA_WEB_PAGE_TO_ROOT . "config" . DIRECTORY_SEPARATOR . "config.inc.php") . "</div>";
⋮
44 $page['body'] .= "\r\n\t<div class=\"body_padded\">\r\n\t<h1>Vulnerability: Insecure CAPTCHA</h1>\r\n\r\n\t{$WarningHtm
   l}\r\n\r\n\t<div class=\"vulnerable_code_area\">\r\n\t\t<form action=\"#\" method=\"POST\" ";
⋮
54 $page['body'] .= "style=\"display:none;\"";
⋮
56 $page['body'] .= ">\r\n\t\t\t<h3>Change your password:</h3>\r\n\t\t\t<br />\r\n\r\n\t\t\t<input type=\"hidden\" name=\"
   step\" value=\"1\" />\n";
⋮
63 $page['body'] .= "\r\n\t\t\tCurrent password:<br />\r\n\t\t\t<input type=\"password\" AUTOCOMPLETE=\"off\" name=\"passw
   ord_current\"><br />";
```

```
⋮
$page['body'] .= "\t\t\tNew password:<br />\r\n\t\t\t<input type=\"password\" AUTOCOMPLETE=\"off\" name=\"password_new\
68 "><br />\r\n\t\t\tConfirm new password:<br />\r\n\t\t\t<input type=\"password\" AUTOCOMPLETE=\"off\" name=\"password_co
nf\"><br />\r\n\r\n\t\t\t" . recaptcha_get_html($_DVWA['recaptcha_public_key']);
⋮
75 $page['body'] .= "\n\n\t\t\t<!-- **DEV NOTE**   Response: 'hidd3n_valu3'   &&   User-Agent: 'reCAPTCHA'   **/DEV NOTE**
-->\n";
⋮
78 $page['body'] .= "\n\t\t\t" . tokenField();
⋮
$page['body'] .= "\r\n\t\t\t<br />\r\n\r\n\t\t\t<input type=\"submit\" value=\"Change\" name=\"Change\">\r\n\t\t</form>
\r\n\t\t{$html}\r\n\t</div>\r\n\r\n\t<h2>More Information</h2>\r\n\t<ul>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('http://
80 www.captcha.net/') . "</li>\r\n\t\t<li>" . dvwaExternalLinkUrlGet('https://www.google.com/recaptcha/') . "</li>\r\n\t\t
<li>" . dvwaExternalLinkUrlGet('https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)') . "</li>\r\n\t</ul
>\r\n</div>\n";
⋮
96 dvwaHtmlEcho($page);
⋮
427 $security = dvwaSecurityLevelGet();
⋮
433 $security = dvwaSecurityLevelGet();
```

DVWA-master/dvwa/includes/dvwaPage.inc.php

```
3  define('DVWA_WEB_PAGE_TO_ROOT', '../../');
⋮
183 function dvwaHtmlEcho($pPage)
184 {
185 $menuBlocks = array();
⋮
186 $menuBlocks['home'] = array();
⋮
188 $menuBlocks['home'][] = array('id' => 'home', 'name' => 'Home', 'url' => '.');
189 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
190 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup / Reset DB', 'url' => 'setup.php');
⋮
193 $menuBlocks['home'][] = array('id' => 'setup', 'name' => 'Setup DVWA', 'url' => 'setup.php');
194 $menuBlocks['home'][] = array('id' => 'instructions', 'name' => 'Instructions', 'url' => 'instructions.php');
⋮
198 $menuBlocks['vulnerabilities'] = array();
199 $menuBlocks['vulnerabilities'][] = array('id' => 'brute', 'name' => 'Brute Force', 'url' => 'vulnerabilities/brute/');
200 $menuBlocks['vulnerabilities'][] = array('id' => 'exec', 'name' => 'Command Injection', 'url' => 'vulnerabilities/exec/
');
201 $menuBlocks['vulnerabilities'][] = array('id' => 'csrf', 'name' => 'CSRF', 'url' => 'vulnerabilities/csrf/');
202 $menuBlocks['vulnerabilities'][] = array('id' => 'fi', 'name' => 'File Inclusion', 'url' => 'vulnerabilities/fi/.?page=
include.php');
203 $menuBlocks['vulnerabilities'][] = array('id' => 'upload', 'name' => 'File Upload', 'url' => 'vulnerabilities/upload/')
;
204 $menuBlocks['vulnerabilities'][] = array('id' => 'captcha', 'name' => 'Insecure CAPTCHA', 'url' => 'vulnerabilities/cap
tcha/');
205 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli', 'name' => 'SQL Injection', 'url' => 'vulnerabilities/sqli/');
206 $menuBlocks['vulnerabilities'][] = array('id' => 'sqli_blind', 'name' => 'SQL Injection (Blind)', 'url' => 'vulnerabili
ties/sqli_blind/');
207 $menuBlocks['vulnerabilities'][] = array('id' => 'weak_id', 'name' => 'Weak Session IDs', 'url' => 'vulnerabilities/wea
k_id/');
208 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_d', 'name' => 'XSS (DOM)', 'url' => 'vulnerabilities/xss_d/');
209 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_r', 'name' => 'XSS (Reflected)', 'url' => 'vulnerabilities/xss_r/
');
210 $menuBlocks['vulnerabilities'][] = array('id' => 'xss_s', 'name' => 'XSS (Stored)', 'url' => 'vulnerabilities/xss_s/');
⋮
213 $menuBlocks['meta'] = array();
⋮
215 $menuBlocks['meta'][] = array('id' => 'security', 'name' => 'DVWA Security', 'url' => 'security.php');
216 $menuBlocks['meta'][] = array('id' => 'phpinfo', 'name' => 'PHP Info', 'url' => 'phpinfo.php');
⋮
218 $menuBlocks['meta'][] = array('id' => 'about', 'name' => 'About', 'url' => 'about.php');
⋮
221 $menuBlocks['logout'] = array();
222 $menuBlocks['logout'][] = array('id' => 'logout', 'name' => 'Logout', 'url' => 'logout.php');
⋮
225 $menuHtml = '';
⋮
227 foreach ($menuBlocks as $menuBlock) {
⋮
229 foreach ($menuBlock as $menuItem) {
230 $selectedClass = $menuItem['id'] == $pPage['page_id'] ? 'selected' : '';
231 $fixedUrl = DVWA_WEB_PAGE_TO_ROOT . $menuItem['url'];
232 $menuBlockHtml .= "<li onclick=\"window.location='{$fixedUrl}'\" class=\"{$selectedClass}\"><a href=\"{$fixedUrl}\">{$m
enuItem['name']}</a></li>\n";
```

```
234 $menuHtml .= "<ul class=\"menuBlocks\">{$menuBlockHtml}</ul>";
241 $securityLevelHtml = 'low';
244 $securityLevelHtml = 'medium';
247 $securityLevelHtml = 'high';
250 $securityLevelHtml = 'impossible';
255 // -- END (security cookie)
    $phpIdsHtml = '<em>PHPIDS:</em> ' . (dvwaPhpIdsIsEnabled() ? 'enabled' : 'disabled');
259 $messagesHtml;
263 $systemInfoHtml = "";
265 $systemInfoHtml = "<div align=\"left\">{$userInfoHtml}<br /><em>Security Level:</em> {$securityLevelHtml}<br />{$phpIds
    Html}</div>";
267 $systemInfoHtml = dvwaButtonSourceHtmlGet($pPage['source_button']) . " {$systemInfoHtml}";
270 $systemInfoHtml = dvwaButtonHelpHtmlGet($pPage['help_button']) . " {$systemInfoHtml}";
    // Date in the past
278 echo "\r\n<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Strict//EN\" \"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dt
    d\">\r\n\r\n<html xmlns=\"http://www.w3.org/1999/xhtml\">\r\n\r\n\t<head>\r\n\t\t<meta http-equiv=\"Content-Type\" cont
    ent=\"text/html; charset=UTF-8\" />\r\n\r\n\t\t<title>{$pPage['title']}</title>\r\n\r\n\t\t<link rel=\"stylesheet\" typ
    e=\"text/css\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/css/main.css\" />\r\n\r\n\t\t<link rel=\"icon\" type=\"\\image/
    ico\" href=\"" . DVWA_WEB_PAGE_TO_ROOT . "favicon.ico\" />\r\n\r\n\t\t<script type=\"text/javascript\" src=\"" . DVWA_W
    EB_PAGE_TO_ROOT . "dvwa/js/dvwaPage.js\"></script>\r\n\r\n\t</head>\r\n\r\n\t<body class=\"home\">\r\n\t\t<div id=\"con
    tainer\">\r\n\r\n\t\t<div id=\"header\">\r\n\r\n\t\t\t<img src=\"" . DVWA_WEB_PAGE_TO_ROOT . "dvwa/images/logo.png\
    " alt=\"Damn Vulnerable Web Application\" />\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_menu\">\r\n\r\n\t\t<di
    v id=\"main_menu_padded\">\r\n\t\t{$menuHtml}\r\n\t\t</div>\r\n\r\n\t\t</div>\r\n\r\n\t\t<div id=\"main_bod
    y\">\r\n\r\n\t\t{$pPage['body']}\r\n\t\t<br /><br />\r\n\t\t{$messagesHtml}\r\n\r\n\t\t</div>\r\n\r\n\t
    \t<div class=\"clear\">\r\n\t\t</div>\r\n\r\n\t\t<div id=\"system_info\">\r\n\t\t{$systemInfoHtml}\r\n\t\t</d
    iv>\r\n\r\n\t\t<div id=\"footer\">\r\n\r\n\t\t<p>Damn Vulnerable Web Application (DVWA) v" . dvwaVersionGet() . "
    </p>\r\n\r\n\t</div>\r\n\r\n</body>\r\n\r\n</html>";
427 $security = dvwaSecurityLevelGet();
433 $security = dvwaSecurityLevelGet();
```

Info Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
info
```

Issue #375390 - DVWA-master/phpinfo.php: 8

Path: DVWA-master/phpinfo.php
Line: 8
Sink: phpinfo
Source:
Taint: HTTP

Code Summary
The operation phpinfo() leaks sensitive system information in line 8 of DVWA-master/phpinfo.php.

DVWA-master/phpinfo.php

```
8 phpinfo();
```

Info Context
The following snippet(s) do not represent actual code but the tainted markup context.

```
Server Configuration
```