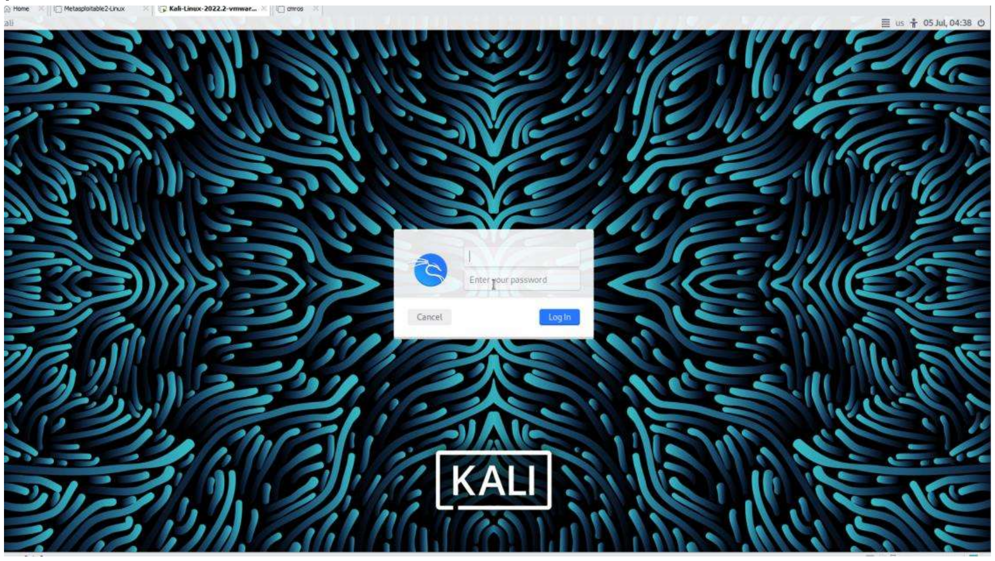
Experiment 3: Examination of a website to test the vulnerability of attacks. – DVWA setup & SQLi

Step 1: Download VMWare or virtual box and Install kali linux

Step2: Login to the kali linux by using the

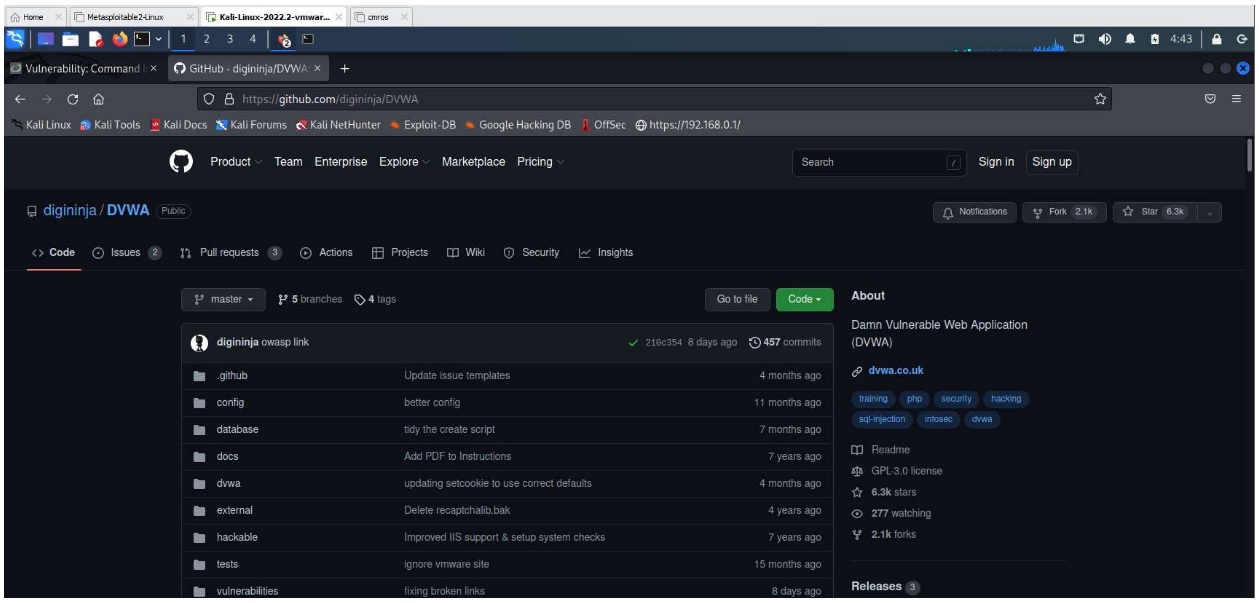
Username: kali

password: kali



Step 3: go to browser and search for DVWA in Kali Linux

DVWA → is a vulnerable website


Vulnerability: Command I
GitHub - digininja/DVWA
https://github.com/digininja/DVWA
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec https://192.168.0.1/
Product Team Enterprise Explore Marketplace Pricing
Search
Sign in Sign up
digininja / DVWA Public
Notifications Fork 2.1k Star 6.3k
Code Issues 2 Pull requests 3 Actions Projects Wiki Security Insights
master 5 branches 4 tags
Go to file Code

digininja owasp link ✓ 210c354 8 days ago 457 commits

.github	Update issue templates	4 months ago
config	better config	11 months ago
database	tidy the create script	7 months ago
docs	Add PDF to Instructions	7 years ago
dvwa	updating setcookie to use correct defaults	4 months ago
external	Delete recaptchalib.bak	4 years ago
hackable	Improved IIS support & setup system checks	7 years ago
tests	ignore vmware site	15 months ago
vulnerabilities	fixing broken links	8 days ago

About

Damn Vulnerable Web Application (DVWA)

dvwa.co.uk

training php security hacking sql-injection infosec dvwa

Readme
GPL-3.0 license
6.3k stars
277 watching
2.1k forks

Releases 3

Installing DVWA:

git clone https://github.com/digininja/DVWA.git

// if any error occurs use sudo in front of git clone

mv DVWA dvwa

chmod -R 777 dvwa/

// to get recursive permission we use -R

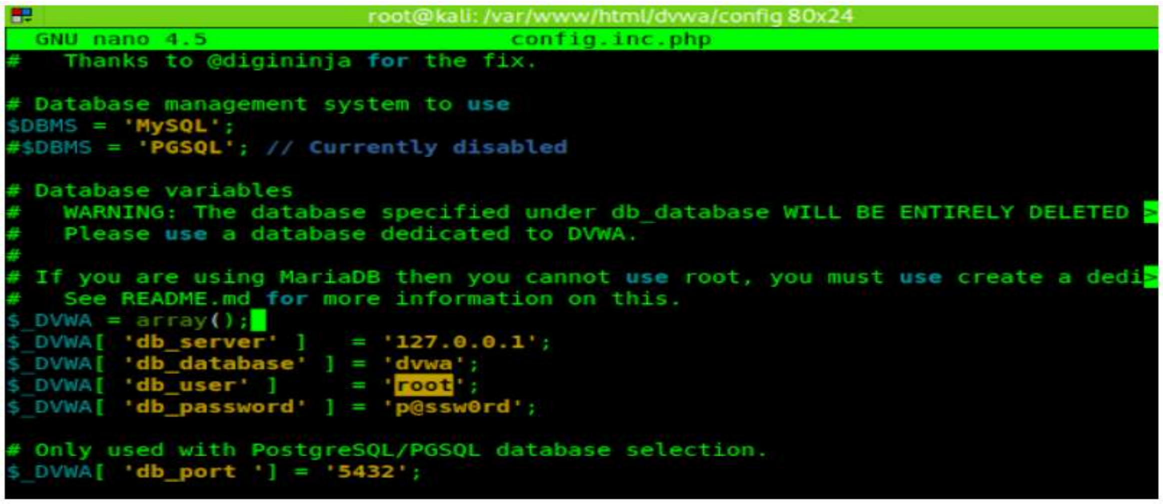
cd dvwa/config

//there will be a dummy file so we can copy to get a new file

//cp used to copy the content of the file

cp config.inc.php.dist config.inc.php

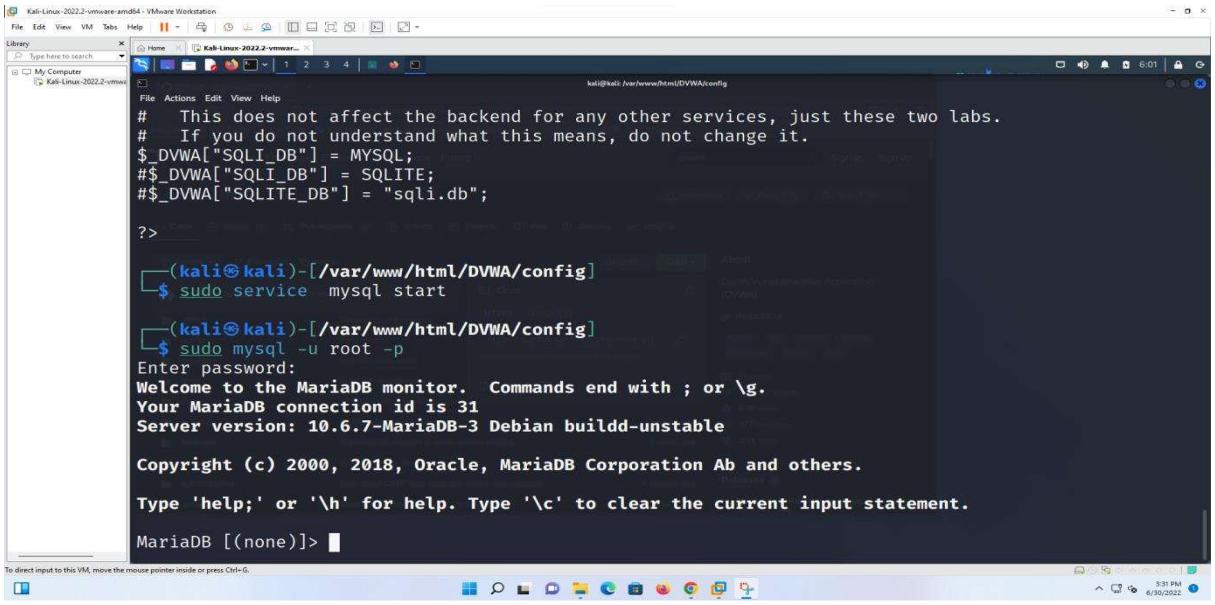
cat or nano config.inc.php



```
root@kali: /var/www/html/dvwa/config 80x24
  GNU nano 4.5                    config.inc.php
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
#   WARNING: The database specified under db_database WILL BE ENTIRELY DELETED >
#   Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedi>
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'root';
$_DVWA[ 'db_password' ] = 'p@ssw0rd';

# Only used with PostgreSQL/PGSQL database selection.
$_DVWA[ 'db_port '] = '5432';
```

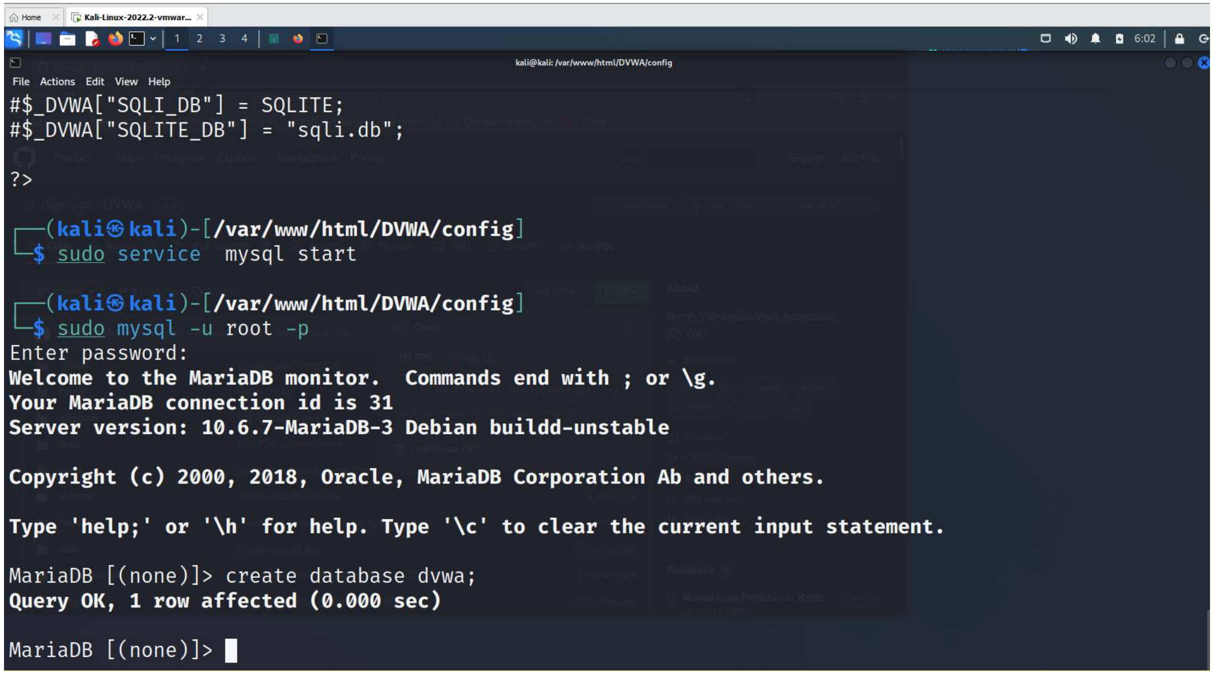
sudo service mysql start

sudo mysql -u root -p



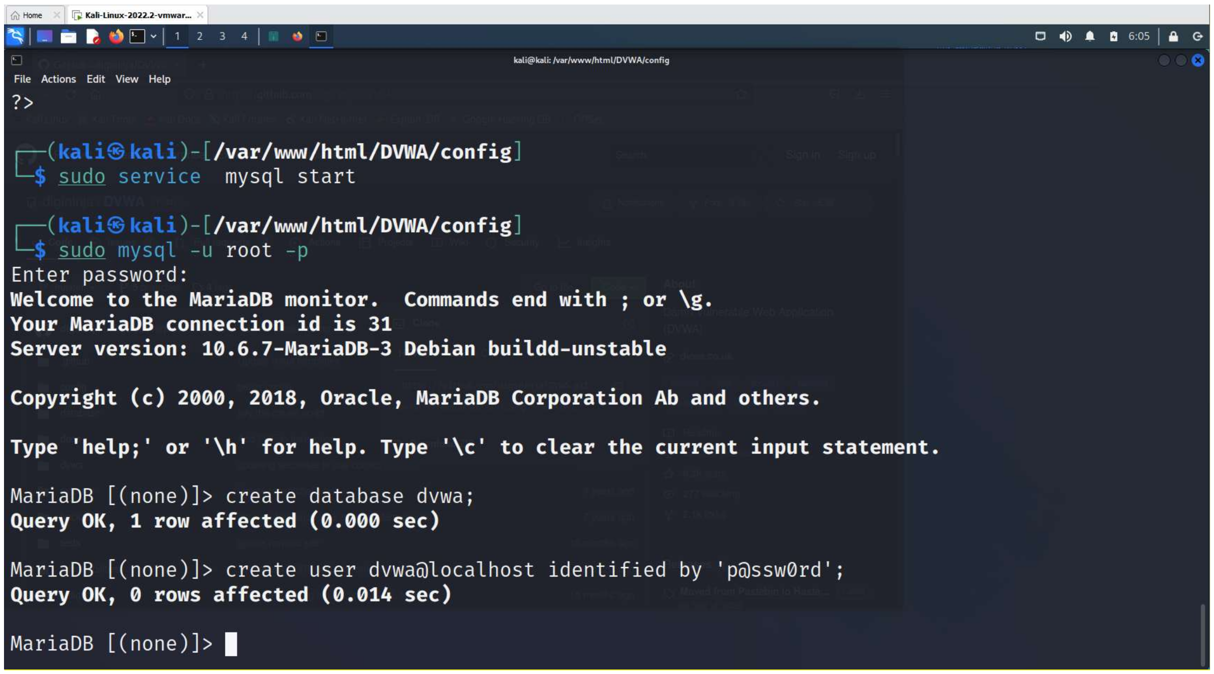
```
#   This does not affect the backend for any other services, just these two labs.
#   If you do not understand what this means, do not change it.
$_DVWA["SQLI_DB"] = MYSQL;
#$_DVWA["SQLI_DB"] = SQLITE;
#$_DVWA["SQLITE_DB"] = "sqli.db";

?>

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo service  mysql start

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.7-MariaDB-3 Debian buildd-unstable

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

create database dvwa;



```
#$_DVWA["SQLI_DB"] = SQLITE;
#$_DVWA["SQLITE_DB"] = "sqli.db";

?>

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo service  mysql start

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.7-MariaDB-3 Debian buildd-unstable

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]>
```

create user dvwa@localhost identified by 'p@ssw0rd';

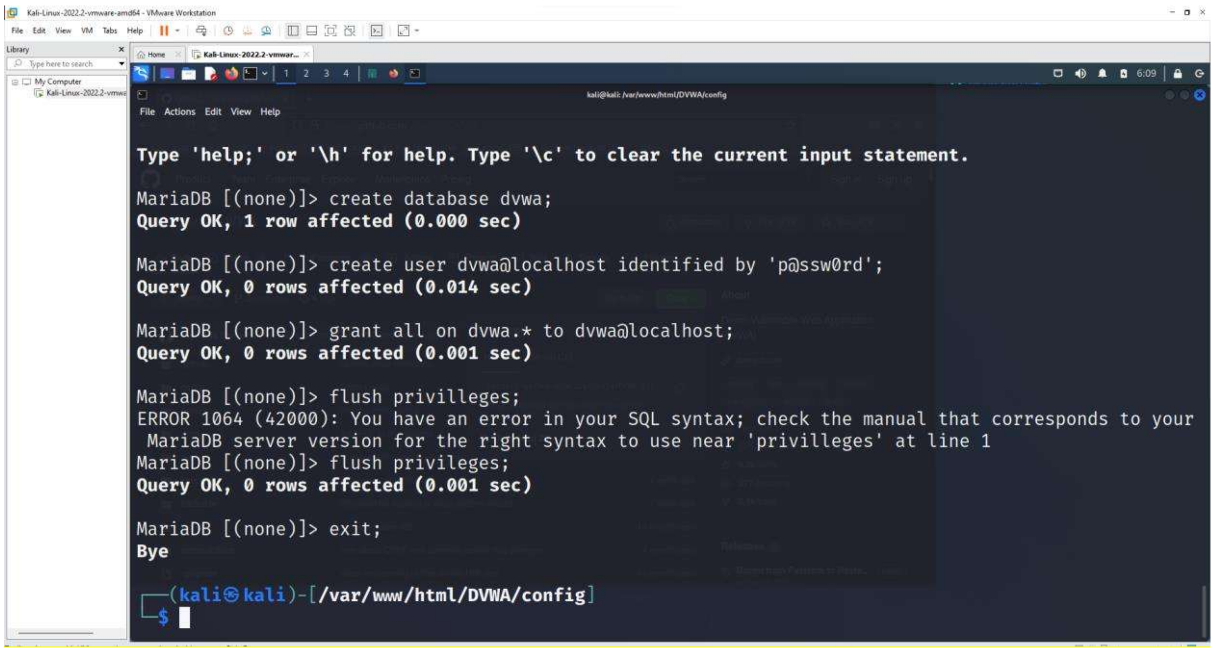


```
?>

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo service  mysql start

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.7-MariaDB-3 Debian buildd-unstable

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user dvwa@localhost identified by 'p@ssw0rd';
Query OK, 0 rows affected (0.014 sec)

MariaDB [(none)]>
```

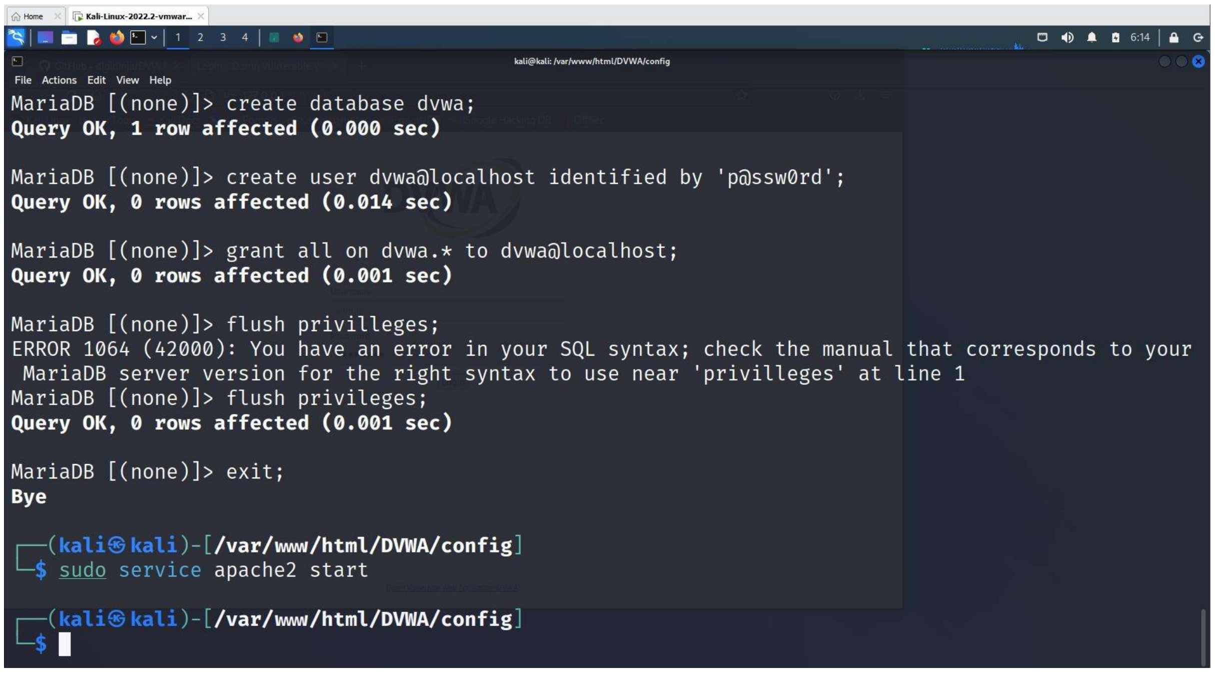
grant all on dvwa.* to dvwa@localhost;

flush privileges;

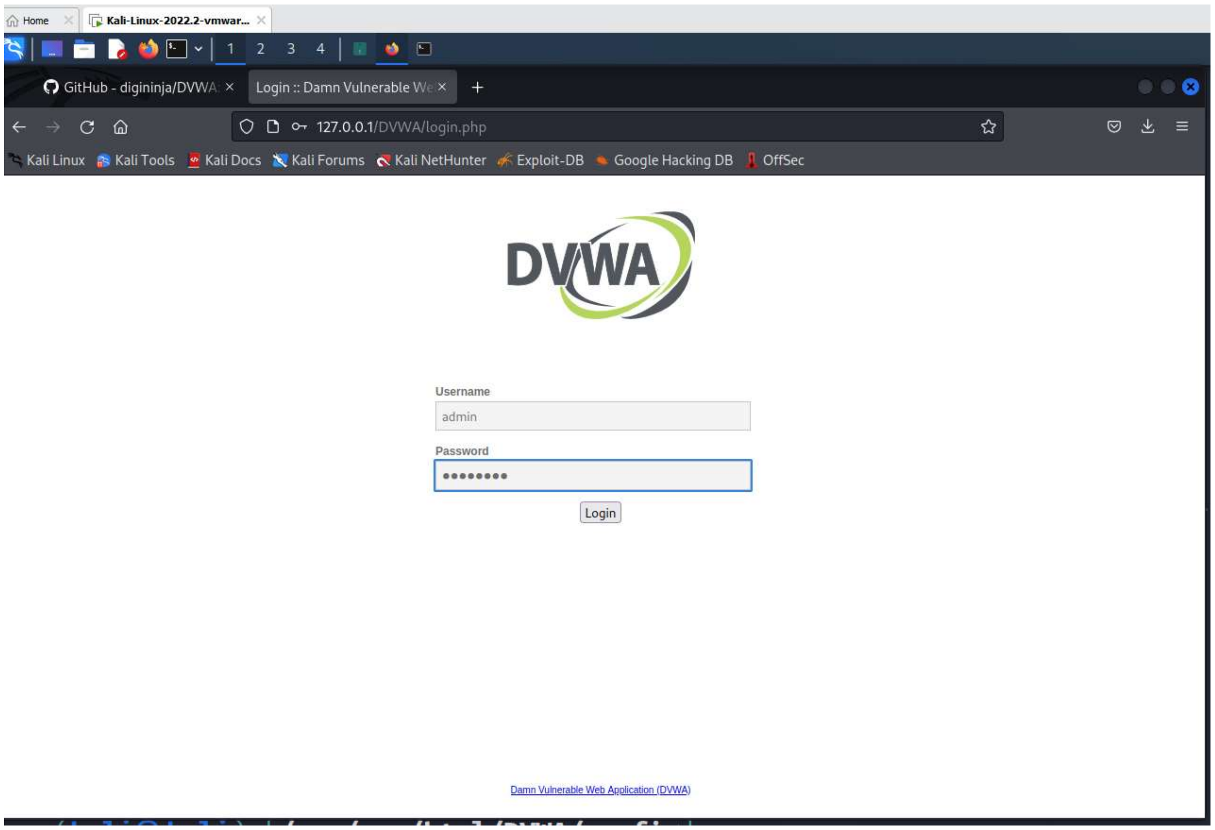
exit;


```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user dvwa@localhost identified by 'p@ssw0rd';
Query OK, 0 rows affected (0.014 sec)

MariaDB [(none)]> grant all on dvwa.* to dvwa@localhost;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> flush privilleges;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
 MariaDB server version for the right syntax to use near 'privilleges' at line 1
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit;
Bye

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ 
```

sudo service apache2 start


```
MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user dvwa@localhost identified by 'p@ssw0rd';
Query OK, 0 rows affected (0.014 sec)

MariaDB [(none)]> grant all on dvwa.* to dvwa@localhost;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> flush privilleges;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
 MariaDB server version for the right syntax to use near 'privilleges' at line 1
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit;
Bye

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo service apache2 start

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ 
```

goto browser and give http://localhost/DVWA or http://127.0.0.1/DVWA/login.php

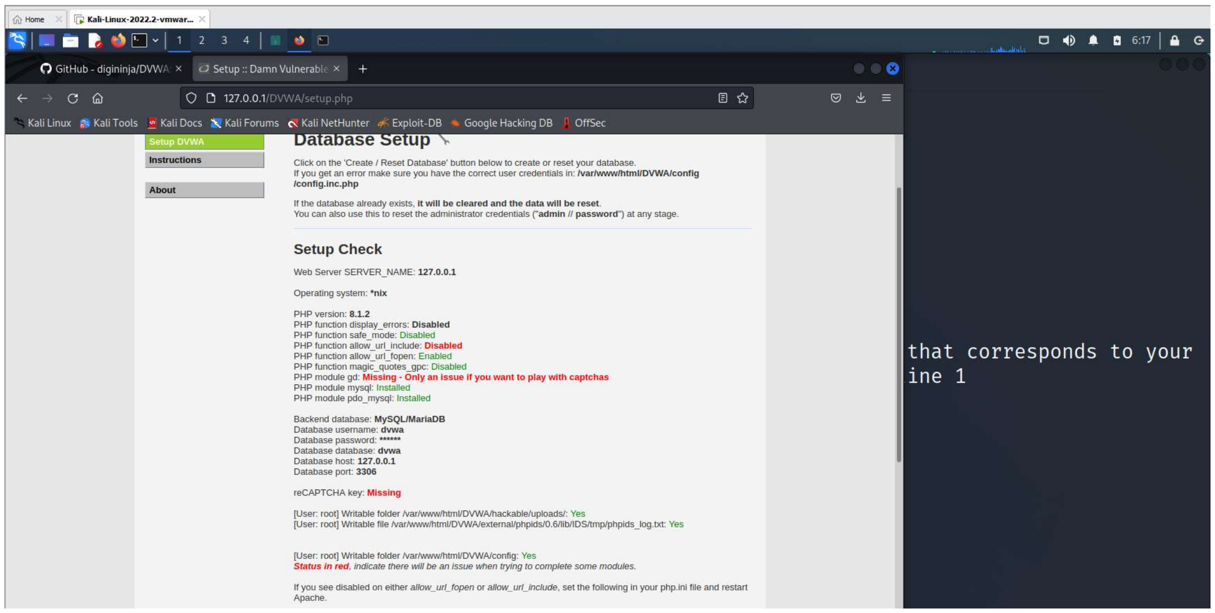

127.0.0.1/DVWA/login.php

Username: admin

Password: ••••••••

username: admin

password: password


127.0.0.1/DVWA/setup.php

Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: **/var/www/html/DVWA/config/config.inc.php**

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

Setup Check

Web Server SERVER_NAME: **127.0.0.1**

Operating system: ***nix**

PHP version: **8.1.2**
PHP function display_errors: **Disabled**
PHP function safe_mode: Disabled
PHP function allow_url_include: **Disabled**
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: **Missing - Only an issue if you want to play with captchas**
PHP module mysql: Installed
PHP module pdo_mysql: Installed

Backend database: **MySQL/MariaDB**
Database username: **dvwa**
Database password: **********
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

reCAPTCHA key: **Missing**

[User: root] Writable folder /var/www/html/DVWA/hackable/uploads/: Yes
[User: root] Writable file /var/www/html/DVWA/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

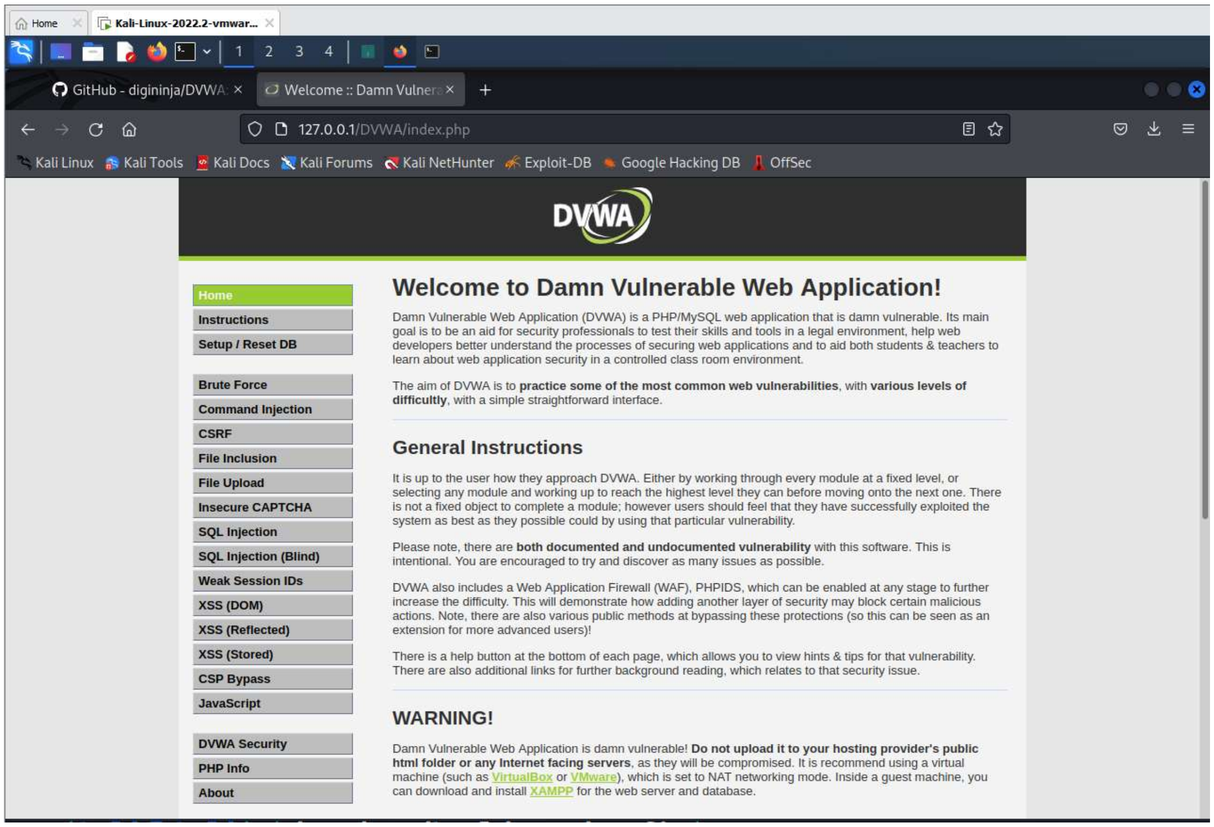
[User: root] Writable folder /var/www/html/DVWA/config: Yes
***Status in red**, indicate there will be an issue when trying to complete some modules.*

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

```
that corresponds to your
ine 1
```

click create database

we get http://127.0.0.1/DVWA/index.php



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 vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

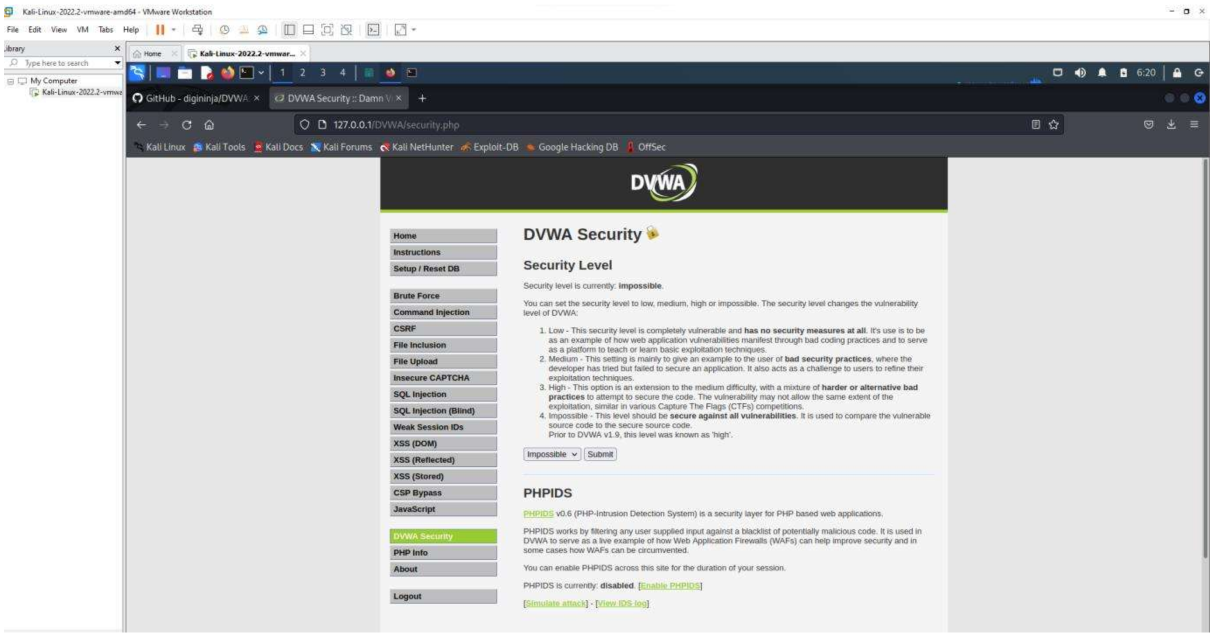
DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

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!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.

Goto DVWA security



DVWA Security

Security Level

Security level is currently: **impossible**.

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'.

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

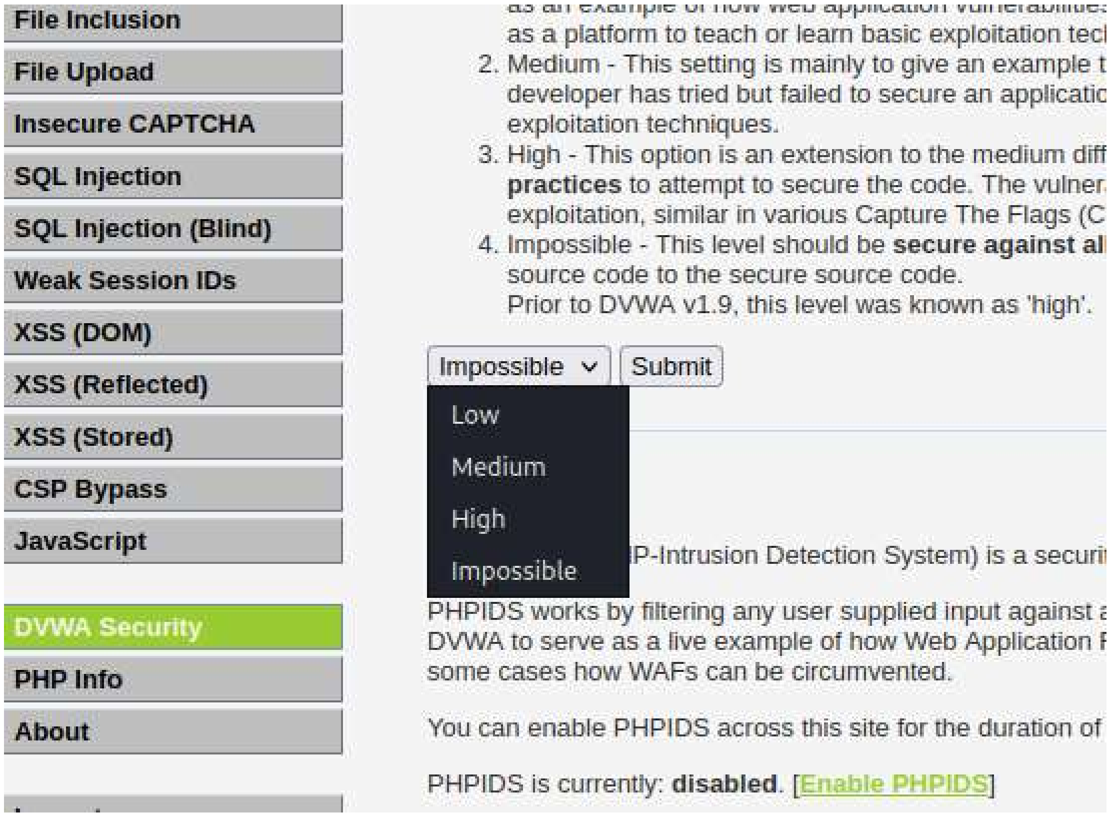
PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

Click on impossible



as a platform to teach or learn basic exploitation tec

2. Medium - This setting is mainly to give an example t developer has tried but failed to secure an applicatio exploitation techniques.
3. High - This option is an extension to the medium diff **practices** to attempt to secure the code. The vulner exploitation, similar in various Capture The Flags (C
4. Impossible - This level should be **secure against al** source code to the secure source code. Prior to DVWA v1.9, this level was known as 'high'.

Impossible Submit

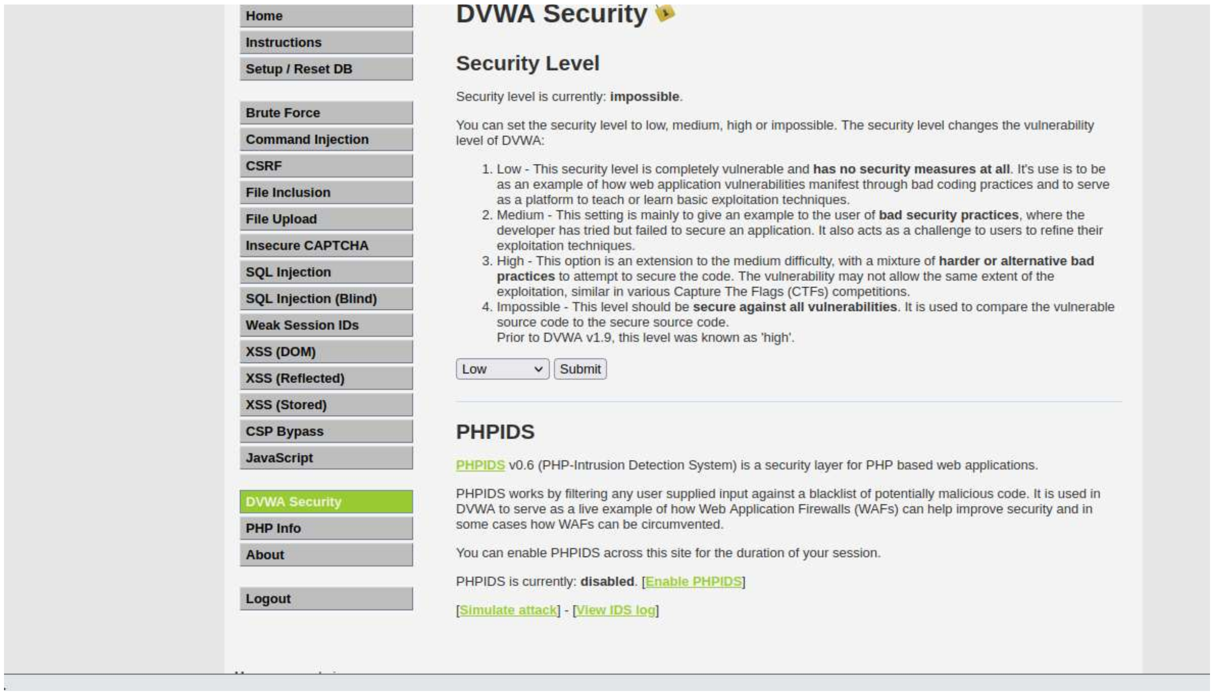
P-Intrusion Detection System) is a securi

PHPIDS works by filtering any user supplied input against a DVWA to serve as a live example of how Web Application F some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of

PHPIDS is currently: **disabled**. [Enable PHPIDS]

set as LOW.



DVWA Security

Security Level

Security level is currently: **impossible**.

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

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [Enable PHPIDS]

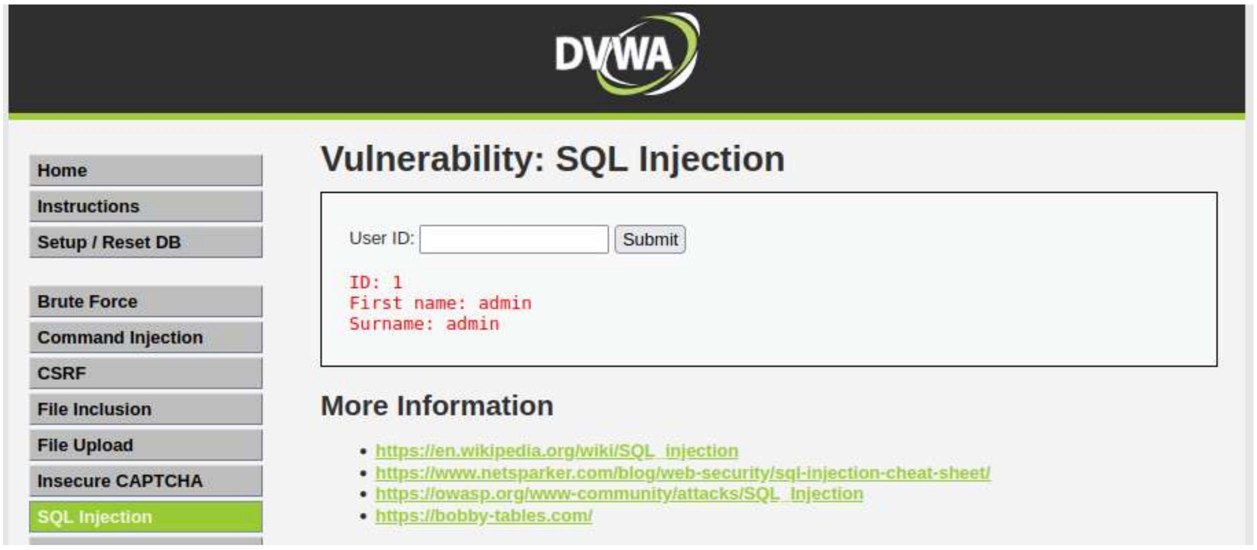
[Simulate attack] - [View IDS log]

Click submit.

Attacking the system:

- SQLInjection:

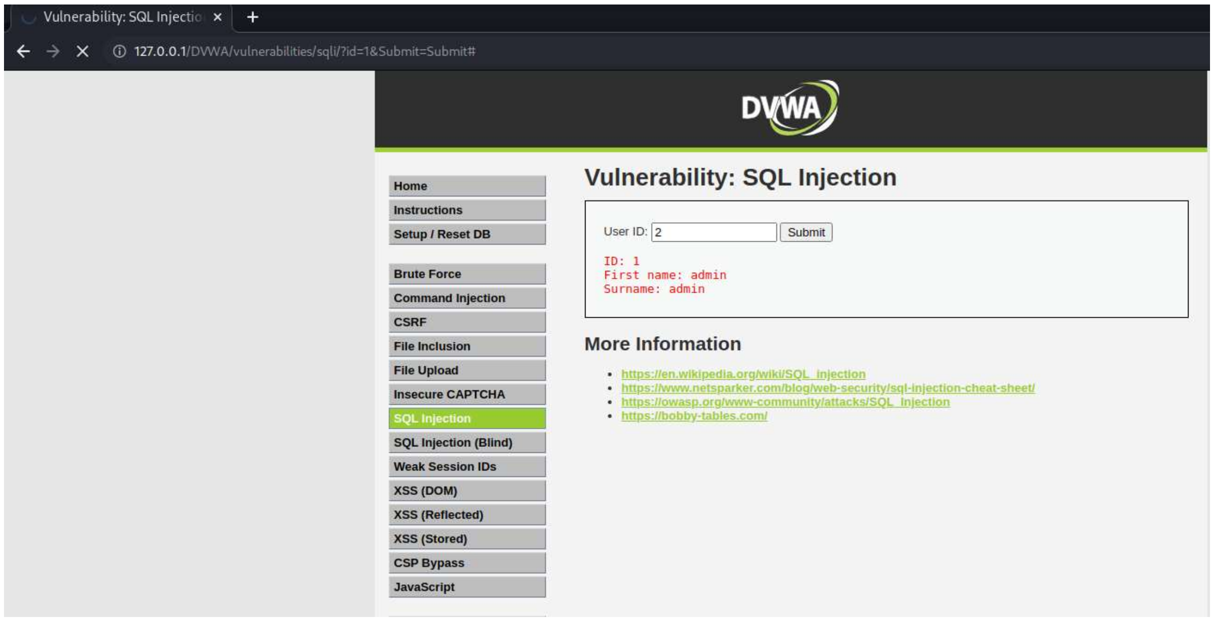
Enter 1 and Click submit



```
ID: 1
First name: admin
Surname: admin
```

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/

Enter 2 and Click submit

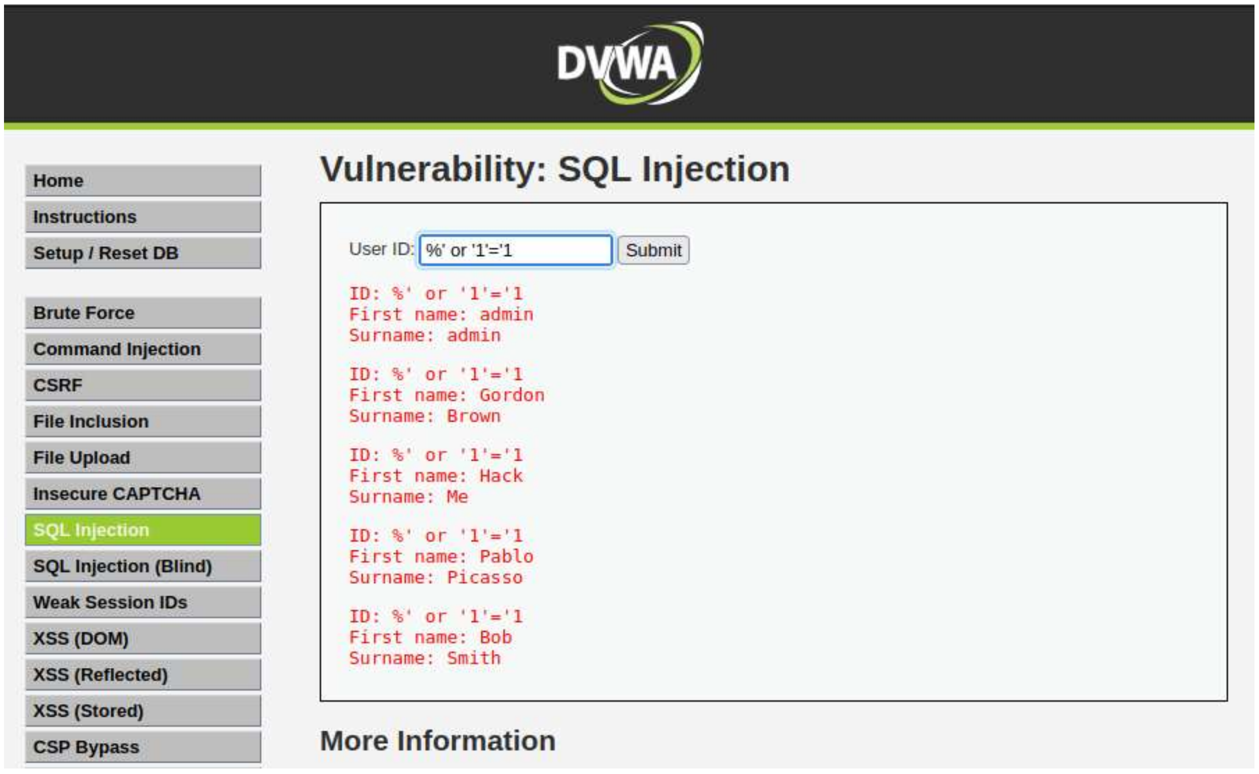


```
ID: 1
First name: admin
Surname: admin
```

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/

Enter %' or '1'='1

It displays all the information.



Vulnerability: SQL Injection

User ID: %' or '1'='1

```
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
```

More Information