

Персональный проект. Этап №2

Основы информационной безопасности

Астраханцева А. А.

15 марта 2024

Российский университет дружбы народов, Москва, Россия

- Астраханцева Анастасия Александровна
- студентка НКАбд-01-22
- Студ. билет: 1132226437
- Российский университет дружбы народов
- <https://anastasiia7205.github.io/>



Ознакомление с специально предназначенным для поиска уязвимостей веб приложением под названием Damn Vulnerable Web Application (DVWA).

Выполнение работы

Репозиторий github DVWA

Переходим в репозиторий github DVWA.

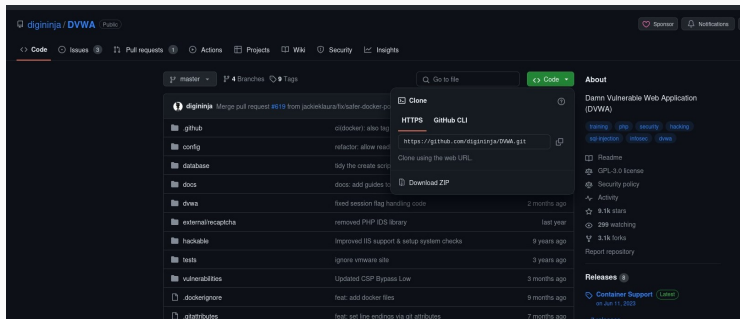
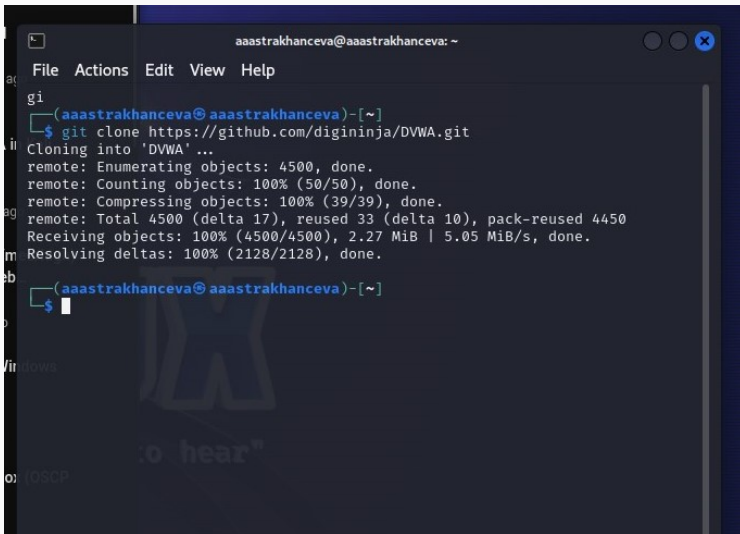


Рис. 1: Репозиторий github DVWA

Клонирование репозитория

Клонируем репозиторий.

A terminal window with a dark blue background and white text. The window title is 'aaastrakhanceva@aaastrakhanceva: ~'. The menu bar shows 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal shows the command 'git clone https://github.com/digininja/DVWA.git' being executed. The output shows the cloning progress, including enumerating, counting, and compressing objects, and receiving the final repository data. The terminal ends with a prompt '\$' and a cursor.

```
aaastrakhanceva@aaastrakhanceva: ~  
File Actions Edit View Help  
gi  
(aaastrakhanceva@aaastrakhanceva)-[~]  
$ git clone https://github.com/digininja/DVWA.git  
Cloning into 'DVWA' ...  
remote: Enumerating objects: 4500, done.  
remote: Counting objects: 100% (50/50), done.  
remote: Compressing objects: 100% (39/39), done.  
remote: Total 4500 (delta 17), reused 33 (delta 10), pack-reused 4450  
Receiving objects: 100% (4500/4500), 2.27 MiB | 5.05 MiB/s, done.  
Resolving deltas: 100% (2128/2128), done.  
(aaastrakhanceva@aaastrakhanceva)-[~]  
$
```

Перемещаем файл DVWA.

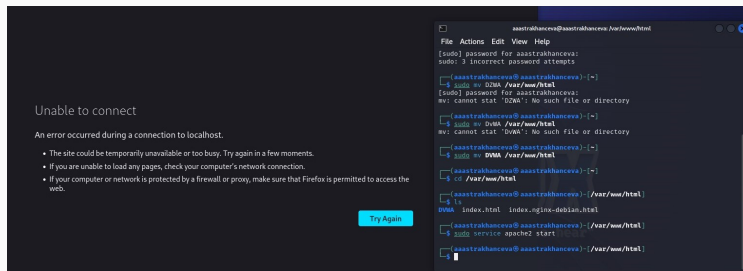


Рис. 3: Перемещение DVWA

Запуск apache сервера

После этого запускаем apache сервер.

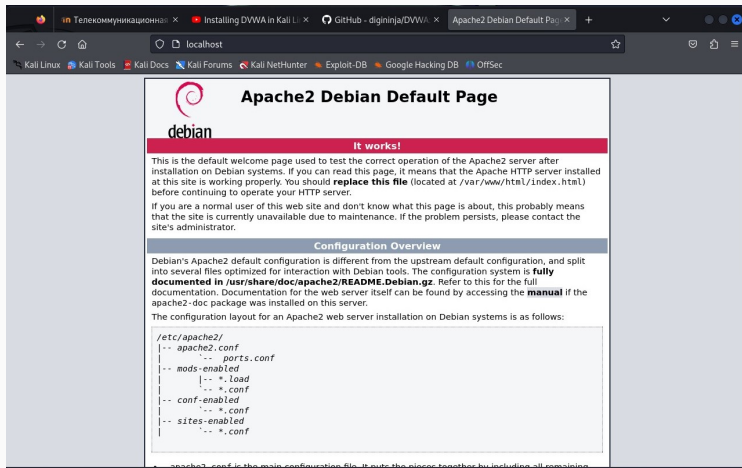


Рис. 4: Запуск apache сервера

Сообщение о конфигуционном файле

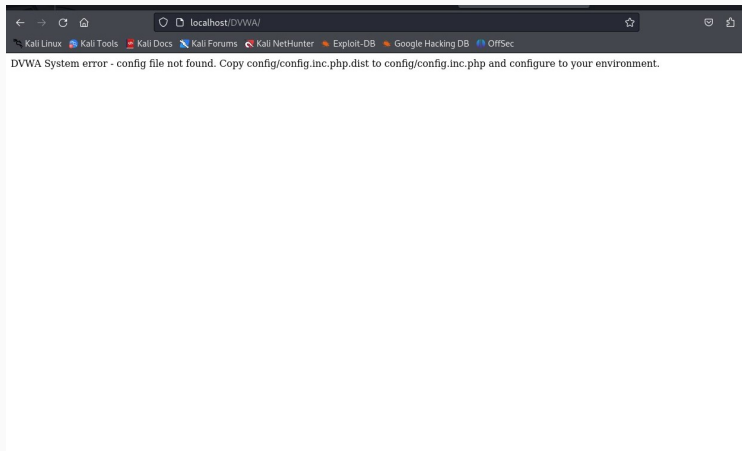


Рис. 5: Сообщение о конфигуционном файле

Копирование конфигурационного файла

Вполняем копирование.

```
(aaastrakhanceva@aaastrakhanceva)-[/var/www/html]
$ cd DVWA

(aaastrakhanceva@aaastrakhanceva)-[/var/www/html/DVWA]
$ ls
CHANGELOG.md  README.id.md  compose.yml  hackable      robots.txt
COPYING.txt   README.md     config       index.php     security.php
Dockerfile    README.pt.md  database     instructions.php security.txt
README.ar.md  README.tr.md  docs         login.php     setup.php
README.es.md  README.zh.md  dvwa         logout.php    tests
README.fa.md  SECURITY.md   external     php.ini       vulnerabilities
README.fr.md  about.php    favicon.ico  phpinfo.php

(aaastrakhanceva@aaastrakhanceva)-[/var/www/html/DVWA]
$ cd config

(aaastrakhanceva@aaastrakhanceva)-[/var/www/html/DVWA/config]
$ ls
config.inc.php.dist

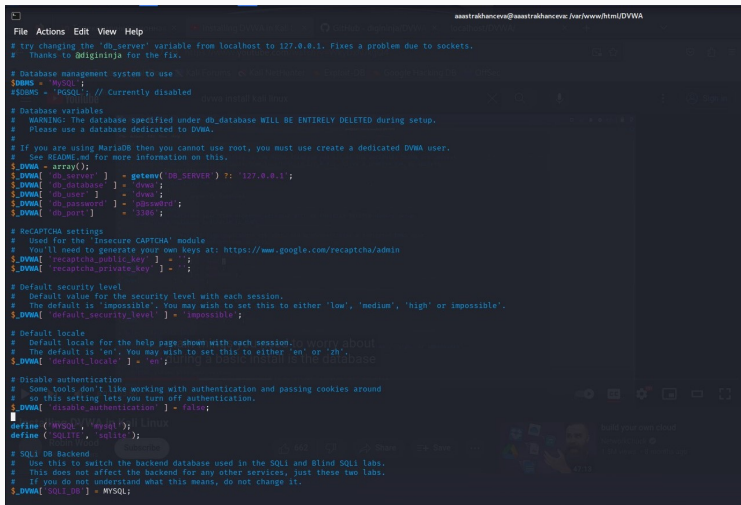
(aaastrakhanceva@aaastrakhanceva)-[/var/www/html/DVWA/config]
$ cd ../.

(aaastrakhanceva@aaastrakhanceva)-[/var/www/html/DVWA]
$ cp config/config.inc.php.dist config/config.inc.php

(aaastrakhanceva@aaastrakhanceva)-[/var/www/html/DVWA]
```

Просмотр конфигурационного файла

Далее просматриваем конфигурационный файл с помощью vim.



```
File Actions Edit View Help
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
# 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 during setup.
# Please use a database dedicated to DVWA.

# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
# See README.md for more information on this.
$_DVWA = array();
$_DVWA['db_server'] = getenv('DB_SERVER') ? '127.0.0.1';
$_DVWA['db_database'] = 'dvwa';
$_DVWA['db_user'] = 'dvwa';
$_DVWA['db_password'] = 'password';
$_DVWA['db_port'] = '3306';

# ReCAPTCHA settings
# Used for the 'Insecure CAPTCHA' module
# You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA['recaptcha_public_key'] = '';
$_DVWA['recaptcha_private_key'] = '';

# Default security level
# Default value for the security level with each session.
# The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or 'impossible'.
$_DVWA['default_security_level'] = 'impossible';

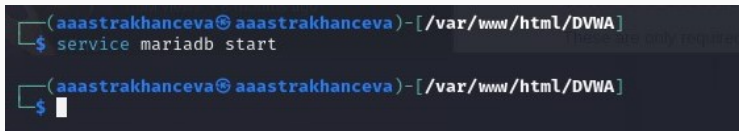
# Default locale
# Default locale for the help page shown with each session.
# The default is 'en'. You may wish to set this to either 'en' or 'zh'.
$_DVWA['default_locale'] = 'en';

# Disable authentication
# Some tools don't like working with authentication and passing cookies around
# so this setting lets you turn off authentication.
$_DVWA['disable_authentication'] = false;

# Define ('MYSQL', 'mysql');
define ('SQLITE', 'sqlite');

# SQLi DB Backend
# Use this to switch the backend database used in the SQLi and Blind SQLi labs.
# 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;
```

Запускаем mariadb для работы с базами данных.

A terminal window with a dark background. The prompt is `(aaastrakhanceva@aaastrakhanceva)-[/var/www/html/DVWA]`. The command `$ service mariadb start` has been entered. The output `mariadb.service: status=running/success, pid=1000, 10s` is visible in the background. Below the first command, the prompt `$` is shown with a cursor.

```
(aaastrakhanceva@aaastrakhanceva)-[/var/www/html/DVWA]
$ service mariadb start
mariadb.service: status=running/success, pid=1000, 10s
$
```

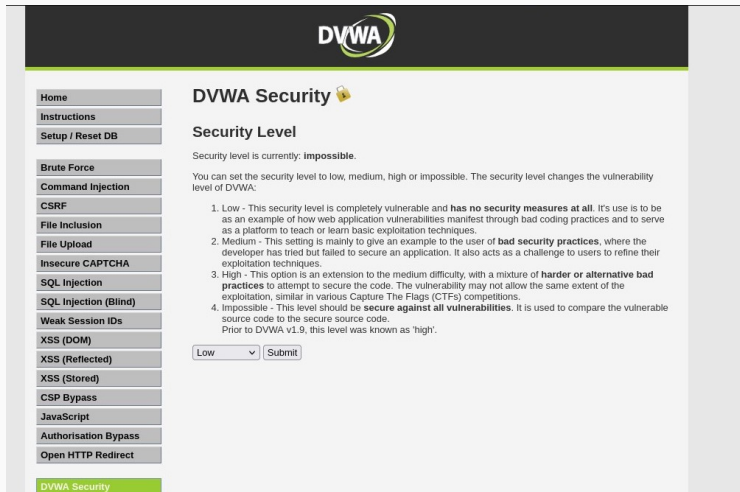
Рис. 8: Запуск mariadb

Необходимые настройки

Создаем новую базу данных.

```
root@aaastrakhanceva: ~  
File Actions Edit View Help  
(aaastrakhanceva@aaastrakhanceva)-[~]  
$ sudo su -  
[sudo] password for aaastrakhanceva:  
(root@aaastrakhanceva)-[~]  
# mysql  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 31  
Server version: 10.11.5-MariaDB-3 Debian n/a  
  
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 (0.002 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 (0.004 sec)  
  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.005 sec)
```

На этом установка окончена, переходим на “http://localhost/DVWA” для дальнейших необходимых настроек.



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The top navigation bar includes links for 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), CSP Bypass, JavaScript, Authorisation Bypass, and Open HTTP Redirect. The main content area is titled "DVWA Security" with a lock icon. Below the title, it states "Security Level" and "Security level is currently: impossible." A paragraph explains that the security level can be set to low, medium, high, or impossible, and that it changes the vulnerability level of DVWA. A list of four levels is provided: 1. Low (completely vulnerable), 2. Medium (bad security practices), 3. High (extension to medium difficulty), and 4. Impossible (secure against all vulnerabilities). At the bottom, there is a dropdown menu currently set to "Low" and a "Submit" button.

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

В ходе выполнения второго этапа индивидуального проекта я ознакомилась с специально предназначенным для поиска уязвимостей веб приложением под названием Damn Vulnerable Web Application (DVWA).

Спасибо за внимание
