

Индивидуальный проект. Этап 2

Основы информационной безопасности

Тойчубекова А.Н.

22 март 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Тойчубекова Асель Нурлановна
- студент 2 курса
- факультет физико-математических и естественных наук
- Российский университет дружбы народов
- 103223035033@pfur.ru

Целью индивидуального проекта является научиться основным способам тестирования веб-приложений.

Damn Vulnerable Web Application (DVWA) — это учебное веб-приложение, специально разработанное для тестирования уязвимостей и отработки навыков в области веб-безопасности.

DVWA включает различные уровни сложности атак, такие как SQL-инъекции, межсайтовый скриптинг (XSS), межсайтовая подделка запросов (CSRF) и другие распространённые уязвимости. Пользователи могут переключать уровень сложности (низкий, средний, высокий) для изучения различных способов атак и методов защиты от них.

Выполнение индивидуального проекта

Выполнение индивидуального проекта

Для начала клонируем репозиторий DVWA из гитхаба, ссылка которой было указано в туисе. Далее перенесем DVWA в /var/www/html, где располагаются все сайты. С помощью команды ls мы видим, что все успешно выполнено.

```
(kali@kali)-[~]
$ git clone https://github.com/digininja/DVWA.git
Cloning into 'DVWA' ...
remote: Enumerating objects: 5105, done.
remote: Counting objects: 100% (91/91), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 5105 (delta 79), reused 67 (delta 67), pack-reused 5014 (from 4
)
Receiving objects: 100% (5105/5105), 2.45 MiB | 3.07 MiB/s, done.
Resolving deltas: 100% (2504/2504), done.

(kali@kali)-[~]
$ sudo mv DVWA /var/www/html
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
```

Выполнение индивидуального проекта

Устанавливаем все права доступа для DVWA. Далее перейдем в конфигурационный файл DVWA и скопируем файл config.inc.php.dist в новый файл config.inc.php.

```
(kali㉿kali)-[/var/www/html]
$ chmod -R 777 DVWA/

(kali㉿kali)-[/var/www/html]
$ cd DVWA

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

(kali㉿kali)-[/var/www/html/DVWA]
$ cd config/

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

(kali㉿kali)-[/var/www/html/DVWA/config]
$ cp config.inc.php.dist config.inc.php
```


Выполнение индивидуального проекта

Далее открываем конфигурационный файл для редактирования, устанавливаем имя пользователя и пароль, сохраняем и закрываем.

```
# Database management system to use
$DBMS = getenv('DBMS') ?: 'MySQL';
#$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'] = getenv('DB_DATABASE') ?: 'dvwa';
$_DVWA['db_user'] = getenv('DB_USER') ?: 'user';
$_DVWA['db_password'] = getenv('DB_PASSWORD') ?: 'password';
$_DVWA['db_port'] = getenv('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
```

Выполнение индивидуального проекта

Затем перейдя в режим суперпользователя настроим базу данных MariaDB для DVWA.
Запускаем mysql.

```
(kali㉿kali)-[/var/www/html] you must create a new database user.  
$ sudo su -  
(root㉿kali)-[~]  
# mysql  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 31  
Server version: 11.4.3-MariaDB-1 Debian n/a  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Support MariaDB developers by giving a star at https://github.com/MariaDB/server  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement  
.  
MariaDB [(none)]> create database dvwa;  
Query OK, 1 row affected (0.004 sec)  
MariaDB [(none)]> create user user@localhost identified by 'password';  
Query OK, 0 rows affected (0.011 sec)  
MariaDB [(none)]> grant all on dvwa.* to user@localhost;  
Query OK, 0 rows affected (0.010 sec)
```

Выполнение индивидуального проекта

Попробуем подключиться к базе данных с ранее созданным пользователем. Мы видим, что вывелось сообщение Database changed, что подтверждает успешное подключение.

```
(kali@kali)-[~/var/www/html]
$ mysql -u user -ppassword
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 11.4.3-MariaDB-1 Debian n/a
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server

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

MariaDB [(none)]> use dvwa;
Database changed
MariaDB [dvwa]>
```

Выполнение индивидуального проекта

Под именем суперпользователя перейдем в директорию `/etc/php/8.4/apache2` и откроем файл `php.ini` для редактирования.

```
(kali㉿kali)-[~]
└─$ sudo su -
[sudo] password for kali:
└─(root㉿kali)-[~]
   └─# cd /etc/php/8.4/apache2

└─(root㉿kali)-[/etc/php/8.4/apache2]
   └─# ls
conf.d  php.ini

└─(root㉿kali)-[/etc/php/8.4/apache2]
   └─# vim php.ini

└─(root㉿kali)-[/etc/php/8.4/apache2]
```

Рис. 6: Открытие файла для редактирования

Выполнение индивидуального проекта

В конфигурационном файле php ставим значения `allow_url_fopen` и `allow_url_include` на `On`, это позволит php обрабатывать удаленные файлы по url и использовать `include`, `require` (позволяют подключать один php файл в другой) для загрузки кода по url.

```
XSS (Reflected)
;;;;;;;;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;;;;;;;;

; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like https:// or ftp://) as
files.
; https://php.net/allow-url-include
allow_url_include = On

; Define the anonymous ftp password (your email address). PHP's default setti
```

Выполнение индивидуального проекта

Далее установим значения `display_errors` и `display_startup_errors` на значения `On`, это позволит ошибкам php отображаться на экране, также при запуске.

```
; sending them to STDOUT.
; Possible Values:
;   Off = Do not display any errors
;   stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
;   On or stdout = Display errors to STDOUT
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-errors
display_errors = On
; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. We strongly recommend you set this to 'off'
; for production servers to avoid leaking configuration details.
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-startup-errors
display_startup_errors = On
; Besides displaying errors, PHP can also log errors to locations such as a
; server-specific log, STDERR, or a location specified by the error_log
; directive found below. While errors should not be displayed on productions
; servers they should still be monitored and logging is a great way to do this
```

Выполнение индивидуального проекта

Установим php-gd, для работы с графикой.

```
(root@kali)-[/etc/php/8.2/apache2] key: Missing
# apt install php-gd
Upgrading:
  libapache2-mod-php php php-common php-mysql
Installing:
  php-gd
Installing dependencies:
  libapache2-mod-php8.4 php8.4 php8.4-common php8.4-gd php8.4-opcache
  php8.4-readline
Suggested packages:
  php-pear
```

Рис. 9: Установка php-gd

Выполнение индивидуального проекта

Добавляем в apache модуль rewrite, который позволяет apache перенаправлять url-адреса.


```
(root@kali)-[/etc/php/8.2/apache2]DB
# sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
systemctl restart apache2
```

```
(root@kali)-[/etc/php/8.2/apache2]
# sudo systemctl restart apache2
```

```
(root@kali)-[/etc/php/8.2/apache2]
#
```

Рис. 10: Добавление модуля в apache

Запускаем apache2 и в посковой системе введем localhost/setup. Мы видим, что нам вывелась старница DVWA для настройки базы данных.



Setup DVWA

Instructions

About

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

General
Operating system: ***nix**

Выполнение индивидуального проекта

Немного опустившись вниз по странице найдем кнопку create database нажмем на нее, чтобы начать работу с базой данных, у нас спросят ввести имя и пароль, введем admin, password.



Username


admin

Password

••••••••

Выполнение индивидуального проекта

Мы видим, что мы успешно вошли в DVWA все наши настройки корректно установлены.



[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](#)

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!

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.

Disclaimer

В ходе выполнения данного этапа индивидуального проекта я научилась устанавливать DVWA на Kali Linux.