

Индивидуальный проект. Этап 2

Основы информационной безопасности

Тойчубекова Асель Нурлановна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение индивидуального проекта	8
5	Выводы	15

Список иллюстраций

4.1	Клонирование репозитория	8
4.2	Копирования конфигурационного файла	9
4.3	Установка имени пользователя и пароля	9
4.4	Создание базы данных	10
4.5	Подключение к базе данных	10
4.6	Открытие файла для редактирования	11
4.7	Редактирование конфигурационного файла php	11
4.8	Редактирование конфигурационного файла php	12
4.9	Установка php-gd	12
4.10	Добавление модуля в apache	13
4.11	Открытие страницы DVWA	13
4.12	Работа с базой данных	14
4.13	Работа с базой данных	14

Список таблиц

1 Цель работы

Целью индивидуального проекта является научиться основным способам тестирования веб приложений.

2 Задание

- Установка DVWA в гостевую систему к Kali Linux

3 Теоретическое введение

Damn Vulnerable Web Application (DVWA) — это учебное веб-приложение, специально разработанное для тестирования уязвимостей и отработки навыков в области веб-безопасности. Оно представляет собой платформу с преднамеренно уязвимым кодом, позволяя исследователям и разработчикам анализировать и исправлять типичные ошибки, встречающиеся в веб-приложениях.

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.

4 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. (рис. 4.1).

```
(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:

(kali㉿kali)-[~]
└─$ cd /var/www/html

(kali㉿kali)-[/var/www/html]
└─$ ls
DVWA  index.html  index.nginx-debian.html
```

Рис. 4.1: Клонирование репозитория

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. (рис. 4.2).


```

(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/config.inc.php.dist config/config.inc.php
cp: cannot stat 'config/config.inc.php.dist': No such file or directory
(kali@kali)-[/var/www/html/DVWA/config]
$ cp config.inc.php.dist config.inc.php
(kali@kali)-[/var/www/html/DVWA/config]

```

Рис. 4.2: Копирования конфигурационного файла

Далее открываем конфигурационный файл для редактирования, устанавливаем имя пользователя и пароль, сохраняем и закрываем. (рис. 4.3).

```

# 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
$_DVWA['recaptcha_public_key'] = getenv('RECAPTCHA_PUBLIC_KEY') ?: '';
$_DVWA['recaptcha_private_key'] = getenv('RECAPTCHA_PRIVATE_KEY') ?: '';

# Default security level
-- INSERT --

```

Рис. 4.3: Установка имени пользователя и пароля

Затем перейдя в режим суперпользователя настроим базу данных MariaDB для DVWA. Запускаем mysql, создаем базу данных, создаем нового пользователя с паролем, который сможет подключиться только с локального компьютера и имеет все права доступа к базе данных. (рис. 4.4).

```
(kali@kali)-[/var/www/html]
└─$ 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)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> 
```

Рис. 4.4: Создание базы данных

Попробуем подключиться к базе данных с ранее созданным пользователем. Мы видим, что вывелось сообщение Database changed, что подтверждает успешное подключение. (рис. 4.5).

```
(kali@kali)-[~]
└─$ 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]> 
```

Рис. 4.5: Подключение к базе данных

Под именем суперпользователя перейдем в директорию /etc/php/8.4/apache2 и

откроем файл `php.ini` для редактирования. (рис. 4.6).

```
(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. 4.6: Открытие файла для редактирования

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`. (рис. 4.7).

```
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
ng is Security
; for this is empty.
; https://php.net/from
;from="john@doe.com"
```

Рис. 4.7: Редактирование конфигурационного файла `php`

Далее установим значения `display_errors` и `display_startup_errors` на значения `On`, это позволит ошибкам `php` отображаться на экране, также при запуске. (рис. 4.8).

```

; 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.
; Default Value: Off
; Development Value: On
; Production Value: On
; https://php.net/log-errors
log_errors = On

```

Рис. 4.8: Редактирование конфигурационного файла php

Установим php-gd, для работы с графикой. (рис. 4.9).

```

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

```

Рис. 4.9: Установка php-gd

Добавляем в apache модуль rewrite, который позволяет apache перенаправлять url-адреса. (рис. 4.10).

```
(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
Database port: 3306
(root@kali)-[/etc/php/8.2/apache2]
# sudo systemctl restart apache2
This section is only important if you want to use the API module.
(root@kali)-[/etc/php/8.2/apache2]
#
```

Рис. 4.10: Добавление модуля в apache

Запускаем apache2 и в посковой системе введем localhost/setup. Мы видим, что нам вывелась старница DVWA для настройки базы данных. (рис. 4.11).

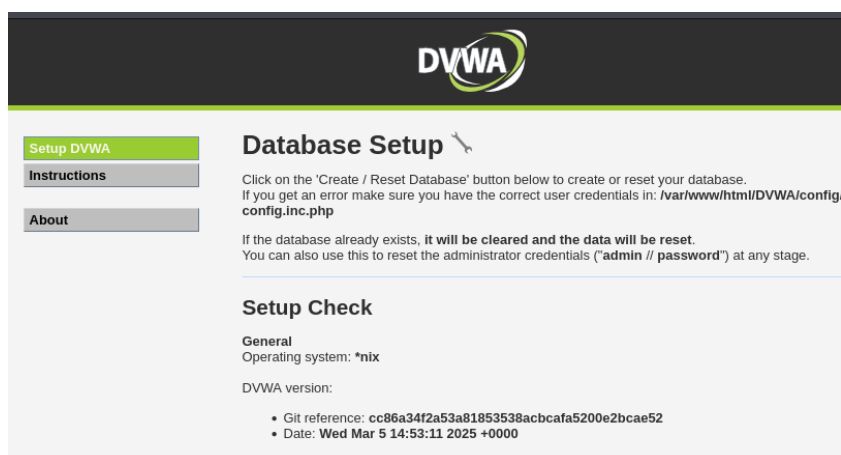


Рис. 4.11: Открытие страницы DVWA

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 в разле имя admin, в разделе пароль password. (рис. 4.12).



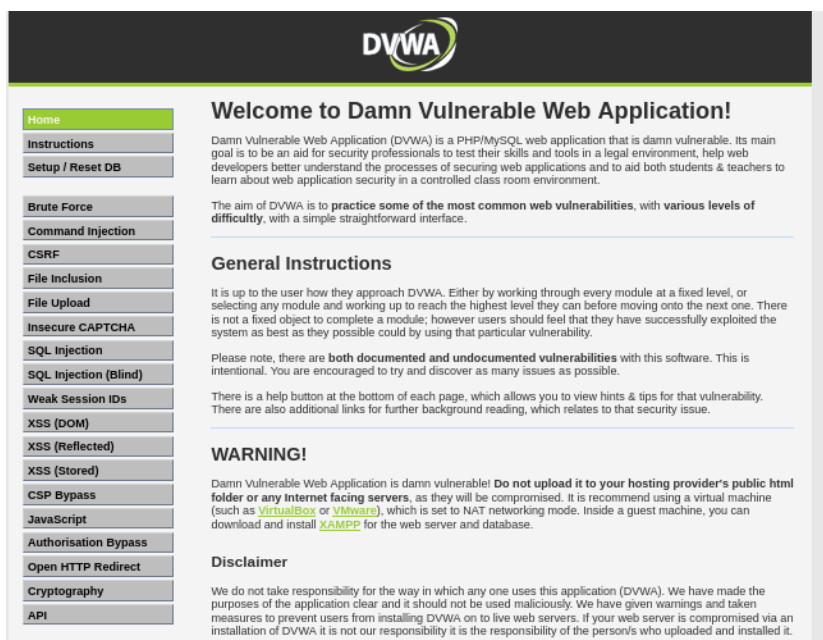
Username

Password

Login

Рис. 4.12: Работа с базой данных

Мы видим, что мы успешно вошли в DVWA все наши настройки корректно установлены. (рис. 4.13).



The image shows the main interface of the Damn Vulnerable Web Application (DVWA). At the top is the DVWA logo. Below it is a navigation menu on the left with links: Home (highlighted), Instructions, Setup / Reset DB, Bruteforce, 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, Open HTTP Redirect, Cryptography, and API. The main content area has a heading "Welcome to Damn Vulnerable Web Application!" followed by a paragraph about DVWA's purpose. Below this is a section titled "General Instructions" with a paragraph explaining the user's goal. Further down is a "WARNING!" section with a paragraph about not uploading DVWA to public servers. At the bottom is a "Disclaimer" section with a paragraph about the application's intended use and the user's responsibility.

Рис. 4.13: Работа с базой данных

5 Выводы

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