

# **Индивидуальный проект**

**Этап 2. Установка DVWA**

Артамонов Тимофей Евгеньевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>4</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

## Список иллюстраций

3.1	git clone . . . . .	8
3.2	Запуск apache2 . . . . .	8
3.3	Веб-страница Apache2 Debian Default Page . . . . .	9
3.4	Новый файл config.inc.php с содержимым config.inc.php.dist . . .	9
3.5	Содержимое файла config.inc.php.dist . . . . .	9
3.6	Стартовую страницу DVWA . . . . .	10
3.7	MariaDB monitor . . . . .	10
3.8	Создание базы данных и пользователя . . . . .	10
3.9	Запуск MariaDB monitor . . . . .	11
3.10	Страница входа DVWA . . . . .	11
3.11	Рабочая область DVWA . . . . .	12

## **Список таблиц**

# 1 Цель работы

Установить DVWA и сделать приготовления для последующей работы.

## 2 Теоретическое введение

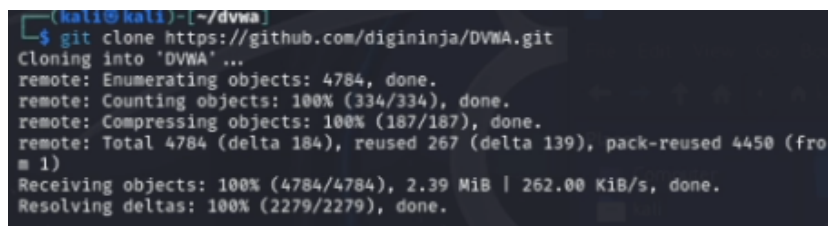
DVWA (Damn Vulnerable Web Application) - это веб-приложение на PHP/MySQL, которое “чертовски” уязвимо. Его основная цель — помочь специалистам по безопасности протестировать свои навыки и инструменты в легальной среде, помочь веб-разработчикам лучше понять процессы обеспечения безопасности веб-приложений, а также помочь студентам и преподавателям изучить вопросы безопасности веб-приложений в контролируемой учебной среде. Цель DVWA — отработать некоторые из наиболее распространенных веб-уязвимостей с различными уровнями сложности, используя простой и понятный интерфейс. Пожалуйста, обратите внимание, что в этом программном обеспечении есть как задокументированные, так и незадокументированные уязвимости. Это сделано намеренно. Вам предлагается попытаться обнаружить как можно больше проблем. [1]

Некоторые из уязвимостей веб приложений, который содержит DVWA: Брутфорс: Брутфорс HTTP формы страницы входа - используется для тестирования инструментов по атаке на пароль методом грубой силы и показывает небезопасность слабых паролей. Исполнение (внедрение) команд: Выполнение команд уровня операционной системы. Межсайтовая подделка запроса (CSRF): Позволяет «атакующему» изменить пароль администратора приложений. Внедрение (инклюд) файлов: Позволяет «атакующему» присоединить удалённые/локальные файлы в веб приложение. SQL внедрение: Позволяет «атакующему» внедрить SQL выражения в HTTP из поля ввода, DVWA включает слепое и основанное на ошибке SQL внедрение. Небезопасная выгрузка файлов: Позволяет «атакующему»

щему» выгрузить вредоносные файлы на веб сервер. Межсайтовый скриптинг (XSS): «Атакующий» может внедрить свои скрипты в веб приложение/базу данных. DVWA включает отражённую и хранимую XSS. Пасхальные яйца: раскрытие полных путей, обход аутентификации и некоторые другие. DVWA имеет три уровня безопасности, они меняют уровень безопасности каждого веб приложения в DVWA: Невозможный — этот уровень должен быть безопасным от всех уязвимостей. Он используется для сравнения уязвимого исходного кода с безопасным исходным кодом. Высокий — это расширение среднего уровня сложности, со смесью более сложных или альтернативных плохих практик в попытке обезопасить код. Уязвимости не позволяют такой простор эксплуатации как на других уровнях. Средний — этот уровень безопасности предназначен главным образом для того, чтобы дать пользователю пример плохих практик безопасности, где разработчик попытался сделать приложение безопасным, но потерпел неудачу. Низкий — этот уровень безопасности совершенно уязвим и совсем не имеет защиты. Его предназначение быть примером среди уязвимых веб приложений, примером плохих практик программирования и служить платформой обучения базовым техникам эксплуатации. [2]

### 3 Выполнение лабораторной работы

Загружаем репозиторий по ссылке. (рис. [3.1])



```
(kali@kali)-[~/dvwa]
$ git clone https://github.com/digininja/DVWA.git
Cloning into 'DVWA' ...
remote: Enumerating objects: 4784, done.
remote: Counting objects: 100% (334/334), done.
remote: Compressing objects: 100% (187/187), done.
remote: Total 4784 (delta 184), reused 267 (delta 139), pack-reused 4450 (from 1)
Receiving objects: 100% (4784/4784), 2.39 MiB | 262.00 KiB/s, done.
Resolving deltas: 100% (2279/2279), done.
```

Рис. 3.1: git clone

Запускаем Apache. (рис. [3.2])



```
(kali@kali)-[~/dvwa]
$ sudo service apache2 start
[sudo] password for kali:
```

Рис. 3.2: Запуск apache2

Открываем веб-страницу Apache. (рис. [3.3])





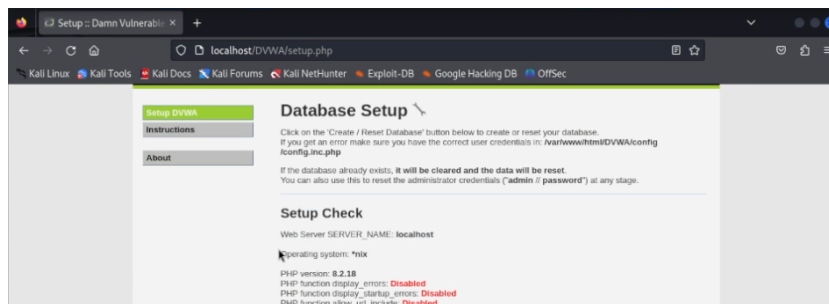


Рис. 3.6: Стартовую страницу DVWA

Запустим mysql. (рис. [3.7])

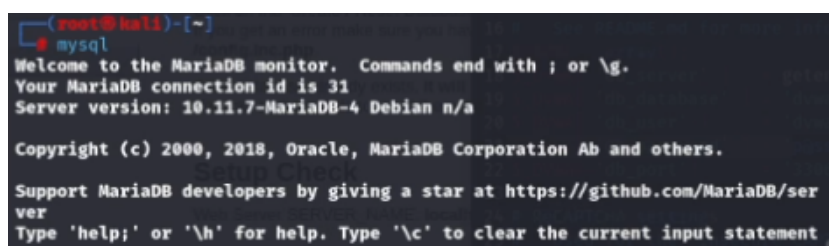


Рис. 3.7: MariaDB monitor

Создадим базу данных и пользователя, указав данные из файла. (рис. [3.8])

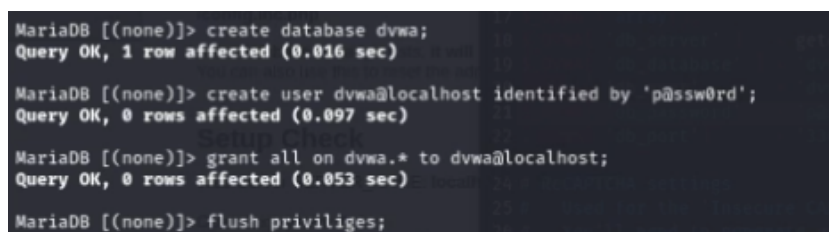


Рис. 3.8: Создание базы данных и пользователя

Запустим MariaDB monitor с созданным пользователем и выберем нашу базу данных. (рис. [3.9])

```
(kali㉿kali)-[~]  
$ mysql -u dvwa -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 31  
Server version: 10.11.7-MariaDB-4 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
```

Рис. 3.9: Запуск MariaDB monitor

Нажмем на странице DVWA create/reset database и попадем на страницу с логином. (рис. [3.10])

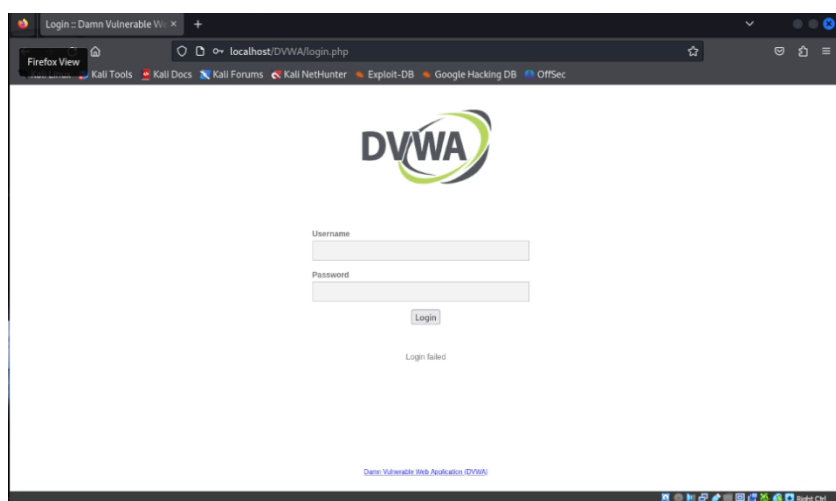


Рис. 3.10: Страница входа DVWA

После входа увидим рабочую область DVWA. (рис. [3.11])

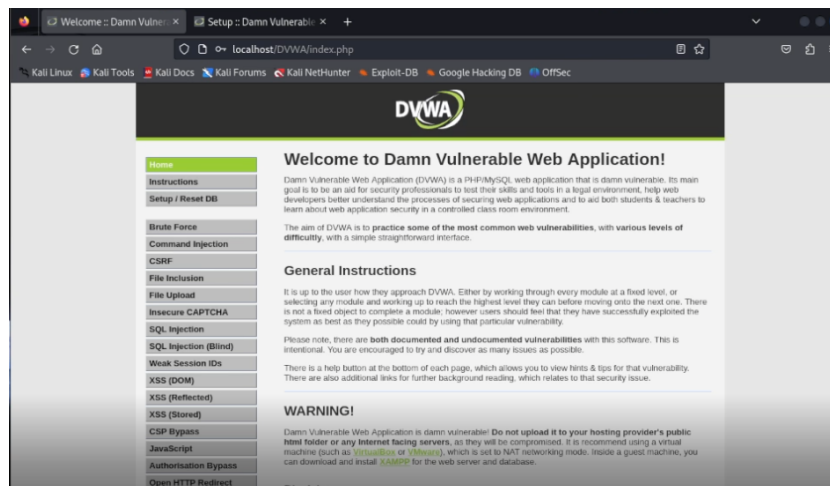


Рис. 3.11: Рабочая область DVWA

## 4 Выводы

Установили DVWA и сделали приготовления для последующей работы.

## Список литературы

1. DVWA [Электронный ресурс]. 2024 GitHub, Inc., 2024. URL: <https://github.com/digininja/DVWA>.
2. Этап 2. DVWA [Электронный ресурс]. RUDN, 2024. URL: <https://esystem.rudn.ru/mod/page/view.php?id=1140704>.