

# **Отчёт о выполнении индивидуальны проект Этап 4**

**Использование Nikto для Сканирования Уязвимостей**

*Акондзо Жордани Лади Гаэл*

# Содержание

<b>1</b>	<b>Общая информация</b>	<b>5</b>
1.1	Цель работы . . . . .	5
1.2	Введение . . . . .	5
1.3	Задачи . . . . .	5
1.4	Инструменты . . . . .	5
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
2.1	Введение в Nikto . . . . .	6
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Использование Nikto . . . . .	7
3.2	Дополнительные Опции Сканирования . . . . .	8
3.3	Результаты Сканирования . . . . .	8
<b>4</b>	<b>Выводы</b>	<b>10</b>

## Список иллюстраций

3.1	Использование Nikto . . . . .	7
3.2	Сохранение результатов сканирования в файл . . . . .	8

## List of Tables

# 1 Общая информация

## 1.1 Цель работы

- Научиться основным способам тестирования веб приложений

## 1.2 Введение

- Ищутся уязвимости в специально предназначенном для этого веб приложении под названием **Damn Vulnerable Web Application (DVWA)**.
- Назначение **DVWA** — попрактиковаться в некоторых самых распространённых веб уязвимостях.
- Предлагается попробовать и обнаружить так много уязвимостей, как сможете.

## 1.3 Задачи

- Найти максимальное количество уязвимостей различных типов.
- Реализовать успешную эксплуатацию каждой уязвимости.

## 1.4 Инструменты

- Для тестирования должен использоваться дистрибутив Kali Linux.
- Можно пользоваться любыми инструментами дистрибутива.

## 2 Теоретическое введение

**Nikto** — это инструмент с открытым исходным кодом для сканирования веб-уязвимостей, который позволяет обнаруживать распространенные проблемы безопасности на веб-серверах, такие как небезопасные файлы, уязвимые конфигурации или устаревшие версии программного обеспечения. **Nikto** уже установлен на **Kali Linux**, что делает его использование простым и удобным для вашего проекта.

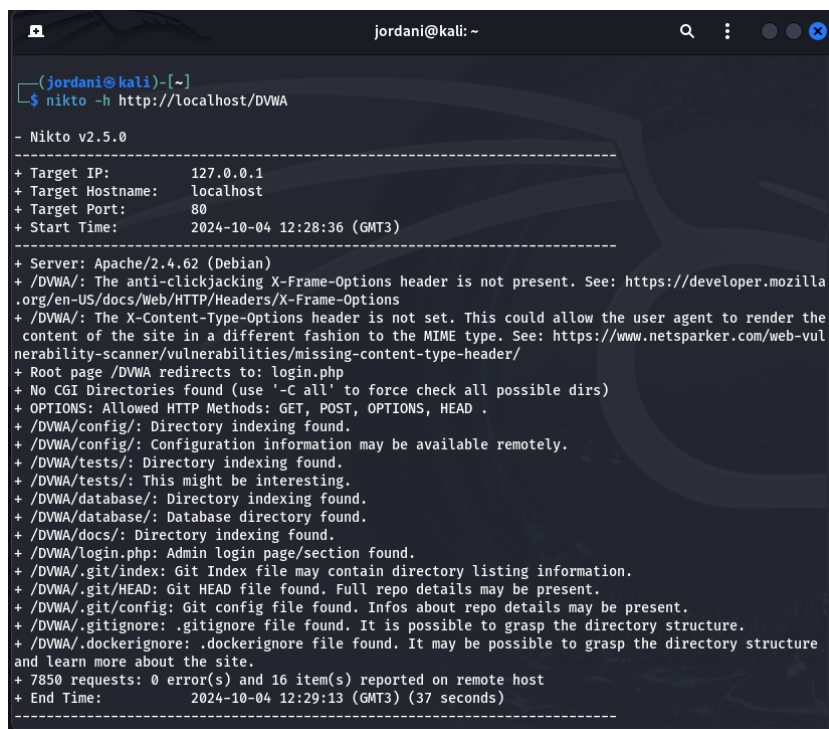
### 2.1 Введение в Nikto

- Nikto используется для проверки безопасности веб-серверов и приложений, сканируя заголовки HTTP, файлы конфигурации и версии программного обеспечения для обнаружения известных уязвимостей.
- Инструмент особенно полезен для тестирования безопасности на начальном этапе, так как он предоставляет исчерпывающий отчет о всех найденных проблемах.

## 3 Выполнение лабораторной работы

### 3.1 Использование Nikto

- Для выполнения сканирования я использовал следующую команду: (рис. 3.1)



```
jordani@kali: ~  
$ nikto -h http://localhost/DVWA  
  
- Nikto v2.5.0  
-----  
+ Target IP: 127.0.0.1  
+ Target Hostname: localhost  
+ Target Port: 80  
+ Start Time: 2024-10-04 12:28:36 (GMT3)  
-----  
+ Server: Apache/2.4.62 (Debian)  
+ /DVWA/: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options  
+ /DVWA/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/  
+ Root page /DVWA redirects to: login.php  
+ No CGI Directories found (use '-C all' to force check all possible dirs)  
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .  
+ /DVWA/config/: Directory indexing found.  
+ /DVWA/config/: Configuration information may be available remotely.  
+ /DVWA/tests/: Directory indexing found.  
+ /DVWA/tests/: This might be interesting.  
+ /DVWA/database/: Directory indexing found.  
+ /DVWA/database/: Database directory found.  
+ /DVWA/docs/: Directory indexing found.  
+ /DVWA/login.php: Admin login page/section found.  
+ /DVWA/.git/index: Git Index file may contain directory listing information.  
+ /DVWA/.git/HEAD: Git HEAD file found. Full repo details may be present.  
+ /DVWA/.git/config: Git config file found. Infos about repo details may be present.  
+ /DVWA/.gitignore: .gitignore file found. It is possible to grasp the directory structure.  
+ /DVWA/.dockerignore: .dockerignore file found. It may be possible to grasp the directory structure and learn more about the site.  
+ 7850 requests: 0 error(s) and 16 item(s) reported on remote host  
+ End Time: 2024-10-04 12:29:13 (GMT3) (37 seconds)  
-----
```

Рис. 3.1: Использование Nikto

- Эта команда запускает полное сканирование на адресе localhost и проверяет уязвимости, характерные для веб-приложений.

## 3.2 Дополнительные Опции Сканирования

- Сохранение результатов сканирования в файл: (рис. 3.2)

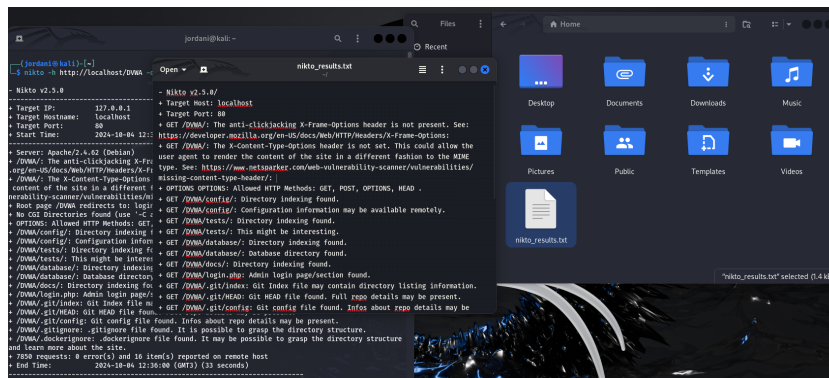


Рис. 3.2: Сохранение результатов сканирования в файл

- Я сохранил результаты в файл для более детального анализа и для документирования проделанной работы.

## 3.3 Результаты Сканирования

- Согласно результатам сканирования Nikto, были обнаружены несколько уязвимостей:

### 1. Отсутствие заголовков безопасности:

- **X-Frame-Options** отсутствует, что делает приложение уязвимым для атак **Clickjacking**. Рекомендовано добавить этот заголовок для предотвращения загрузки страницы в **iframe** третьими лицами.
- **X-Content-Type-Options** также не установлен, что потенциально позволяет браузеру интерпретировать содержимое неправильно.

### 2. Индексирование каталогов:



- Были найдены каталоги **/config/**, **/tests/**, **/database/** с включенной функцией индексирования, что позволяет пользователю видеть содержимое каталогов и, возможно, получить доступ к конфиденциальной информации. Для повышения безопасности я рекомендую отключить индексирование этих каталогов с помощью **.htaccess**.

### 3. Файлы конфигурации Git и Docker:

- Были обнаружены файлы **.git/config**, **.gitignore**, и **.dockerignore**, которые могут содержать критически важную информацию о структуре приложения. Я заблокировал доступ к этим файлам, чтобы предотвратить возможные атаки.

## 4 Выводы

Использование **Nikto** позволило мне не только выявить текущие проблемы безопасности, но и лучше понять, как неправильные конфигурации могут сделать приложение уязвимым для атак. Я получил ценный опыт в анализе безопасности веб-приложений и в настройке веб-сервера для обеспечения защиты от известных угроз.