Индивидуальный проект.

Этап 4. Использование nikto

Стариков Данила НПИбд-02-22

Содержание

# 1 Цель работы

Познакомиться с утилитой nikto для поиска уязвимостей веб-серверов, проверить ее работу на ранее установленном сервере DVWA.

# 2 Теоретическое введение

nikto — базовый сканер безопасности веб-сервера. Он сканирует и обнаруживает уязвимости в веб-приложениях, обычно вызванные неправильной конфигурацией на самом сервере, файлами, установленными по умолчанию, и небезопасными файлами, а также устаревшими серверными приложениями.

Для запуска nukto в командной строке необходимо указать несколько параметров (Табл. 1):

Таблица 1: Основные параметры запуска nikto

| Ключ | Описание ключа |
| --- | --- |
| -host | Указание IP-адрес сервера, для которого необходимо выполнить проверку на уязвимости. Передав текстовый файл можно указать список адресов для проверки. |
| -Display | Определение сообщения nikto будет выводить в консоль. Возможные значения: |
|  | 1 – показывать перенаправления |
|  | 2 – показывать полученные файлы cookie |
|  | 3 – показывать все ответы 200/OK |
|  | 4 – показывать URL-адреса, для которых требуется аутентификация |
|  | D – вывод для отладки |
|  | V – подробный вывод |
|  | E – показывать все HTTP ошибки |
|  | P – выводить прогресс в стандартный вывод (STDOUT) |
| -Tuning | Контроль над тестами, которые nikto будет проводить. Возможные значения: |
|  | 1 –Интересный файл / Замеченный в логах |
|  | 2 –Неправильная настройка / Файл по умолчанию |
|  | 3 –Раскрытие информации |
|  | 4 –Внедрение (XSS/Script/HTML) |
|  | 5 –Удаленный поиск файлов - Внутри корневого веб-каталога |
|  | 6– Отказ в обслуживании |
|  | 7– Удаленный поиск файлов - на сервере |
|  | 8 –Выполнение команд / Удаленная оболочка |
|  | 9 –SQL-инъекция |
|  | 0 –Загрузка файла |
|  | a –Обход проверки подлинности |
|  | b– Идентификация программного обеспечения |
|  | c – Включение удаленного источника |
|  | d –Веб-сервис |
|  | e – Административная консоль |
|  | x – Параметры обратной настройки (т.е. включить все, кроме указанных) |
| -o | Указание, в какой файл записать результаты проверки. |
| -Format | Указание формата файла результатов (htm, csv, json, nbe, sql, txt, xml) |

Более подробная информация может быть найдена на man-pages [1].

# 3 Выполнение лабораторной работы

1. Запустили веб-сервер DVWA, который будем тестировать (Рис. 1):

sudo service mysql start  
sudo service apache2 start

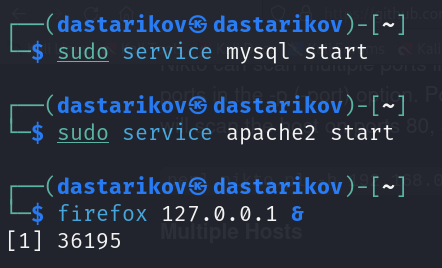


Рис. 1: Запус веб-сервера DVWA.

1. Открыли веб-страницу запущенного сервера, чтобы убедиться, что он работает (Рис. 3):

firefox 127.0.0.1/DVWA

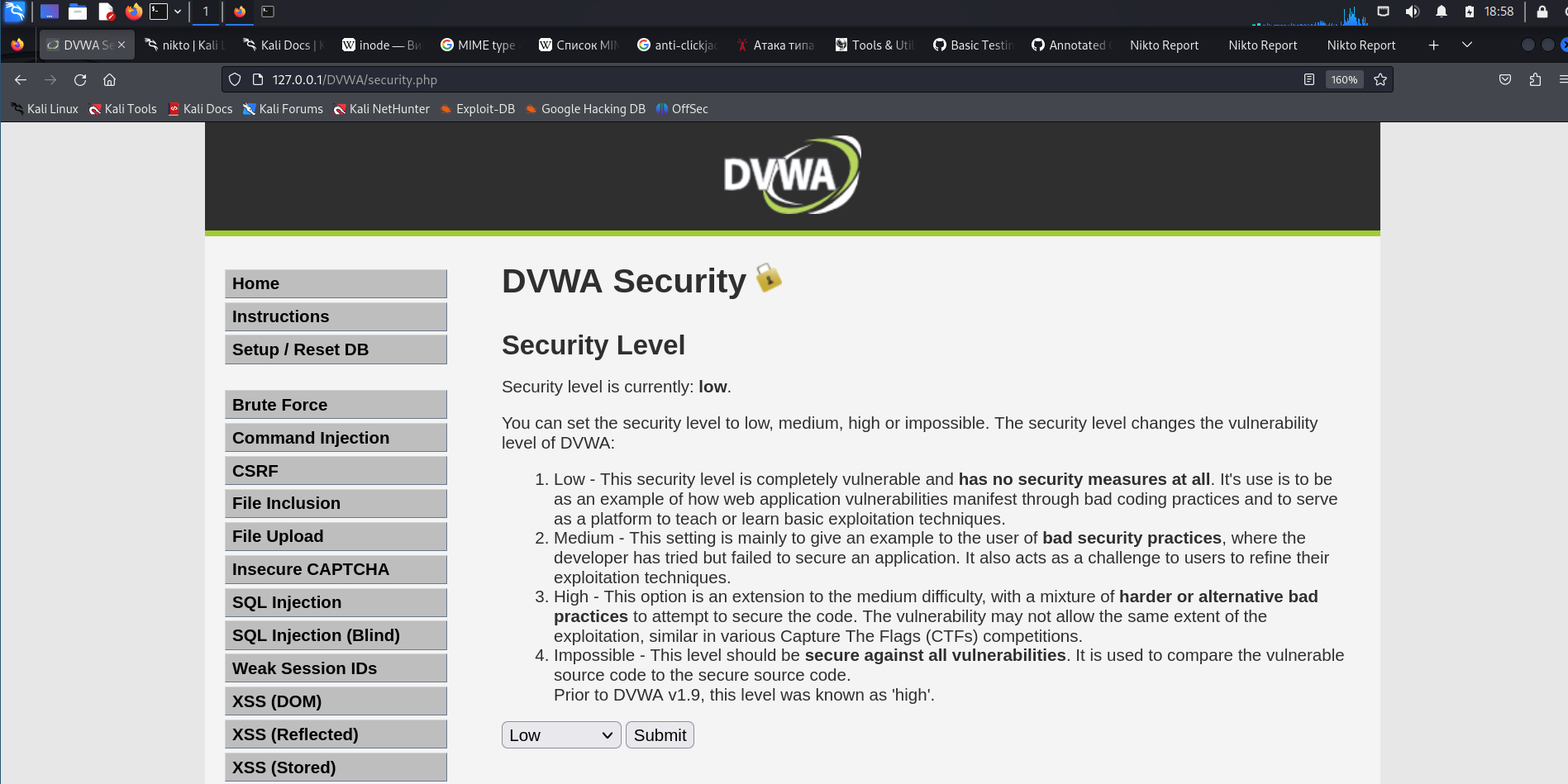


Рис. 2: Проверка работы сервера.

1. Открыли описание утилиты nikto для составления команды (Рис. 1):

man nikto

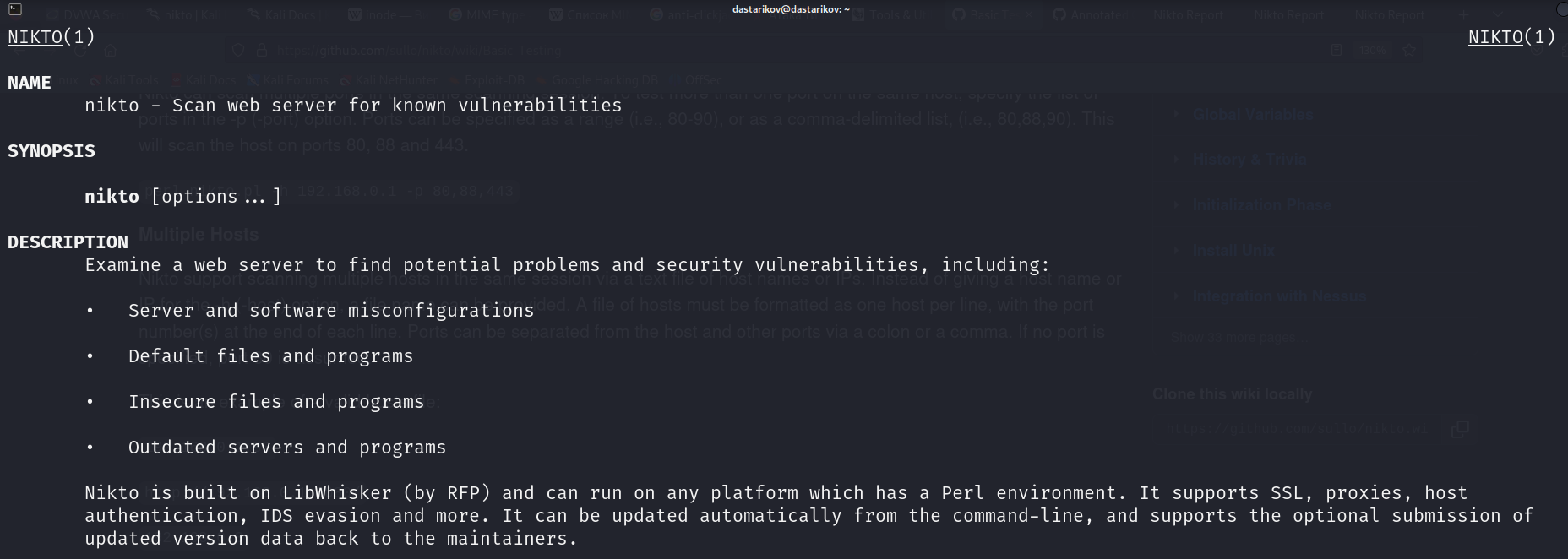


Рис. 3: Man-page nikto.

1. Запустили утилиту nikto со следующими параметрами (Рис. 4):

nikto -Display 1234EP -o report.html -Format htm -Tuning 123bde -host 127.0.0.1

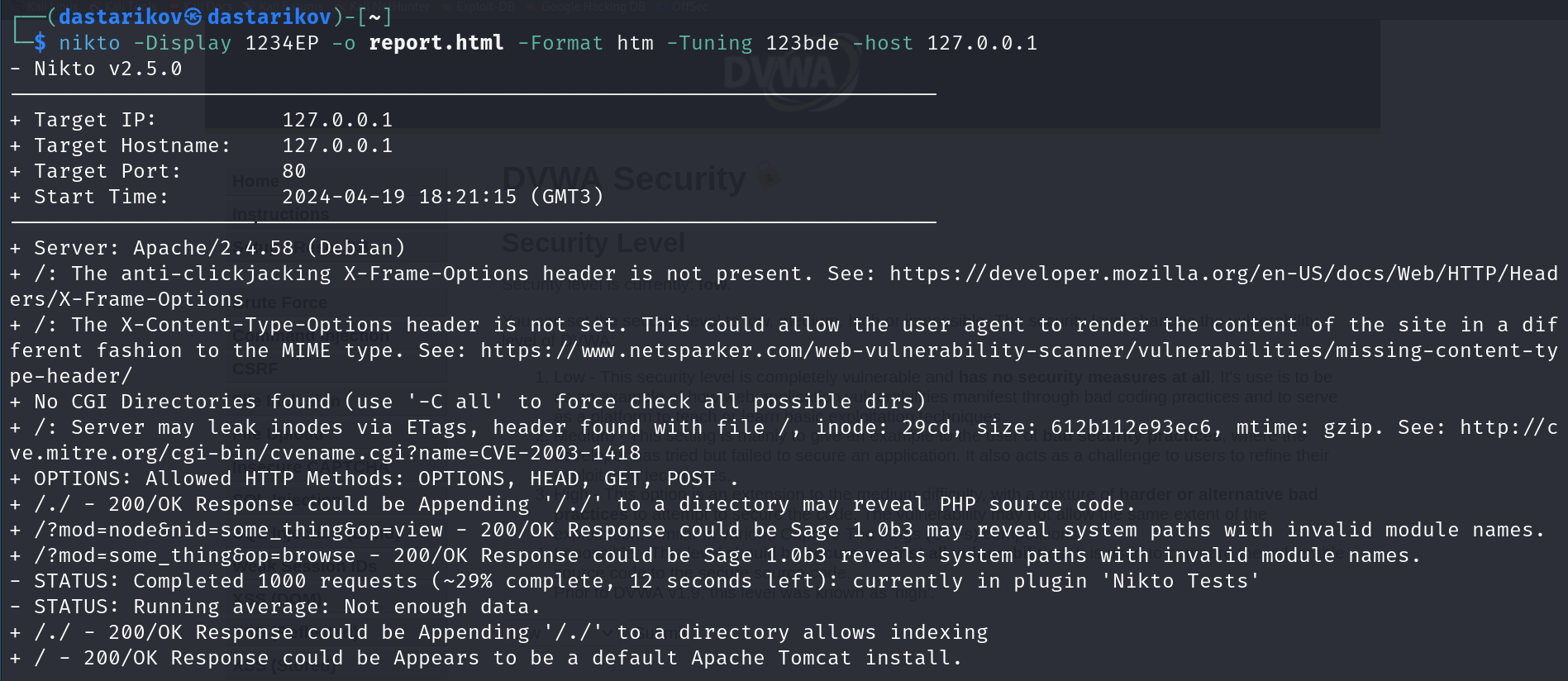


Рис. 4: Консольный вывод программы во время работы.

Помимо вывода в консоль указали утилите сохранить отчет в файле output.html, приведен пример найденной уязвимости (Рис. 5) и итог по тестированию (Рис. 6).

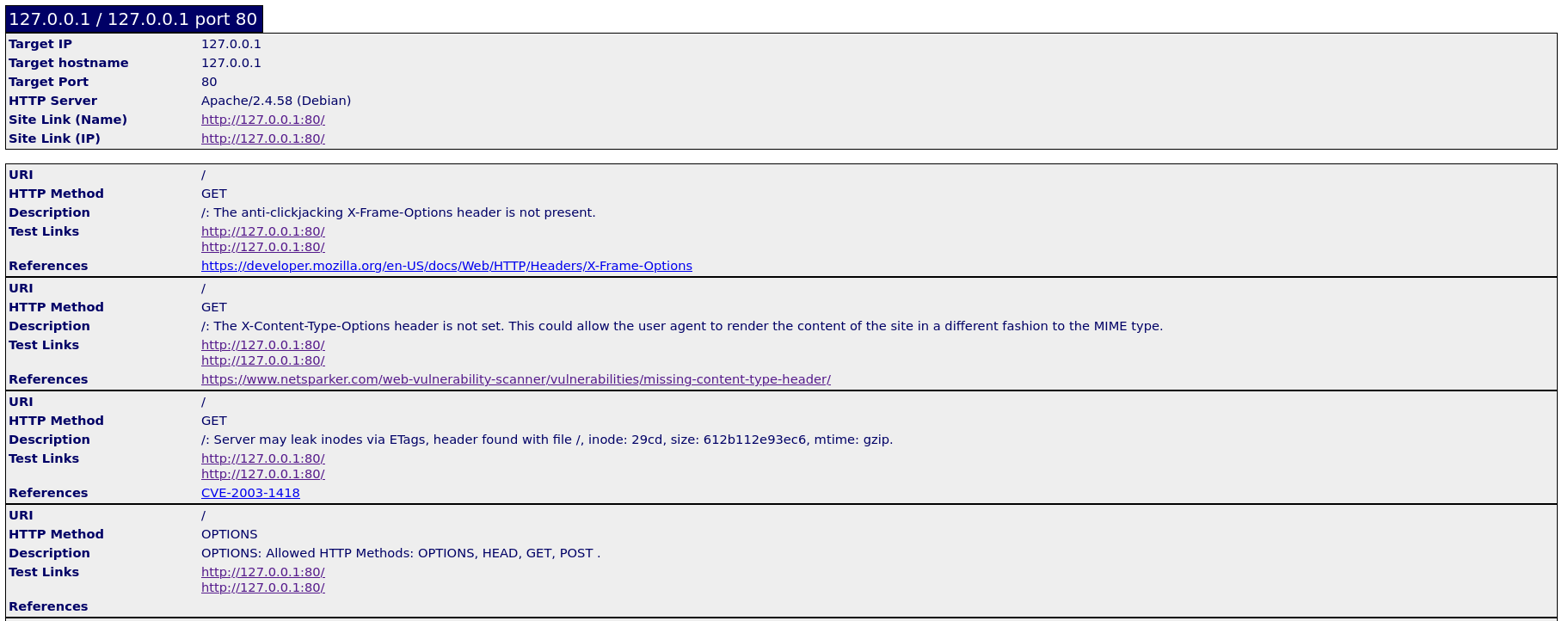


Рис. 5: Пример найденной уязвимости веб-сервера.

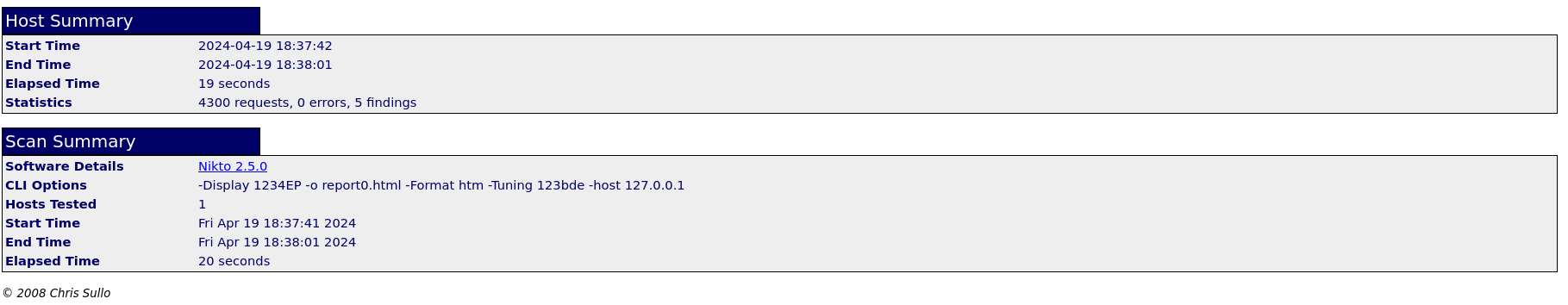


Рис. 6: Итоговый отчет по тестированию веб-сервера.

1. Также запустили другой тест, на этот раз не уточняя параметр Tuning, поэтому были проведены все варианты тестов. Обратим внимание, что в этот тест занял больше времени, и нашел больше уязвимостей (Рис. 7)

nikto -Display 1234EP -o report.html -Format htm -host 127.0.0.1

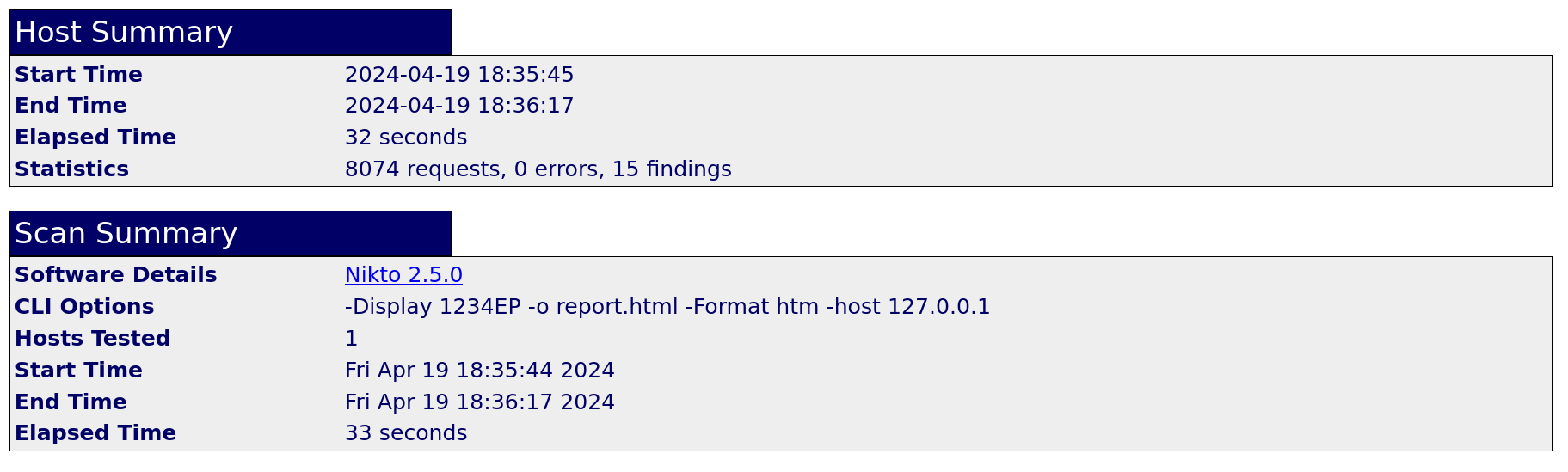


Рис. 7: Итоговый отчет по тестированию веб-сервера с большим числом тестов.

# 4 Выводы

В результате работы познакомились с утилитой nikto и проверили уязвимости веб-сервера DVWA с разными параметрами теста.

# Список литературы

1. Sullo C., Lodge D. nikto(1) Linux User’s Manual. 2010.