Четвертый этап индивидуального проекта

Использование nikto

Беличева Дарья Михайловна

Содержание

# 1 Цель работы

Целью данной работы является использование Nikto для сканирования уязвимостей веб-приложения.

# 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.

В начале сканирования всегда отображается следующий блок с информацией[1]:

* Target IP: IP адрес сканируемого домена;
* Target Hostname: имя хоста (доменное имя) сканируемого сайта;
* Target Port: порт, на котором находится сайт;
* Start Time: дата и время начала сканирования в формате год-месяц-день час:минута:секунда.

Вывод результатов сканирования имеет несколько форматов:

1. Формат: Тип компонента сайта: Наименование компонента. Пример: Server: nginx.
2. Описание: Nikto умеет определять, какие компоненты использует сайт. Сюда относят наименование веб-сервера, используемой СУБД, фреймворков, языков программирования, а также их версии. Формат: путь до файла/директории, где найдена уязвимость: описание уязвимости. Пример: /phpinfo.php: Output from the phpinfo() function was found.

# 3 Выполнение лабораторной работы

Будем проверять работу nikto на веб-приложении DVWA. Для этого запустим mysql и apache2 (рис. 1).

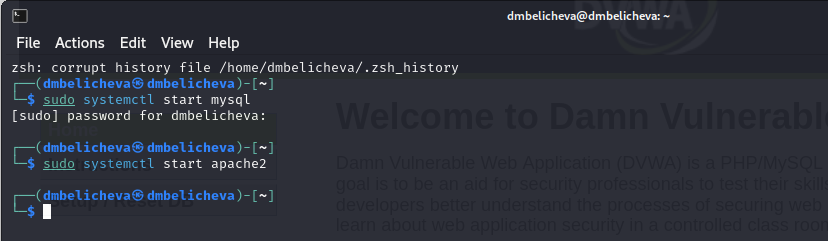


Рис. 1: Запуск mysql и apache2

Теперь можем в адресной строке открыть DVWA (рис. 2).

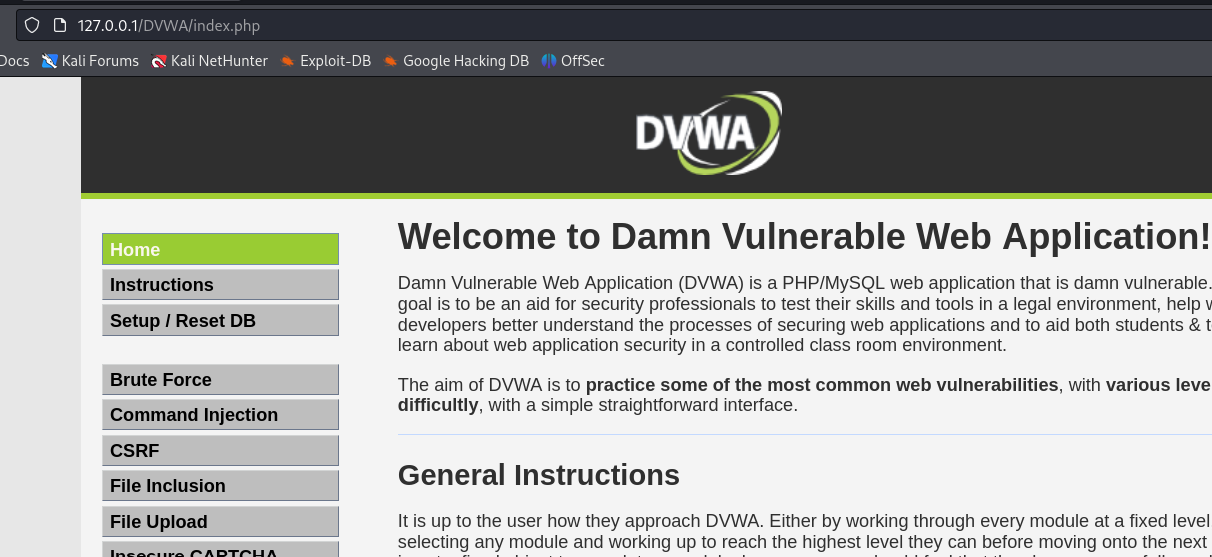


Рис. 2: Страница DVWA

Теперь проверим, что nikto установлен (рис. 3).

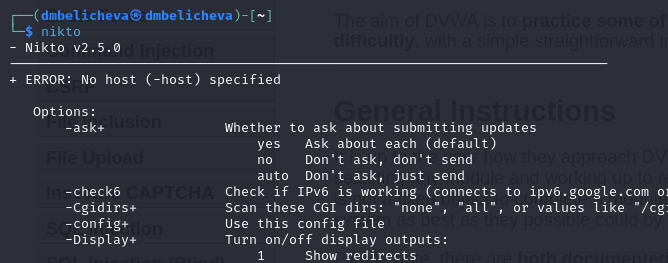


Рис. 3: Проверка наличия nikto

Теперь можем проверить сайт DVWA. Минимальный синтаксис для запуска сканирования выглядит следующим образом:

nikto -h доменное\_имя или IP\_адрес

Параметр -h обязателен к использованию, иначе программа не сможет запустить сканирование (рис. 4). Чтобы получить отчет в удобном формате, можно использовать опцию Format. Я указала формат html.

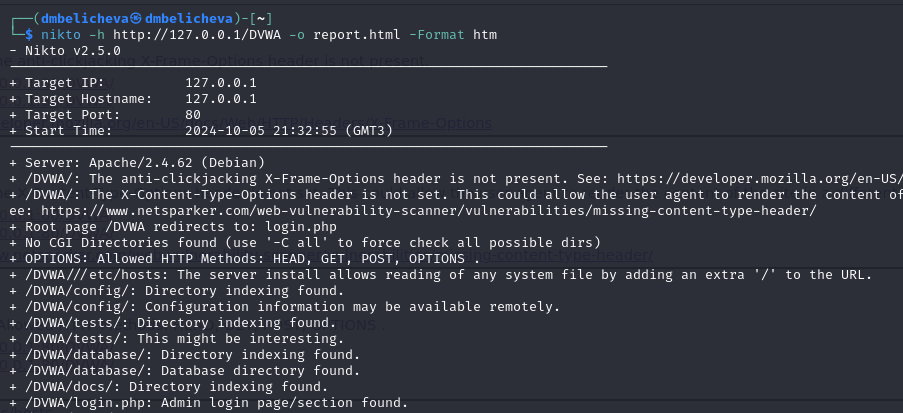


Рис. 4: Проверка уязвимостей по доменному имени

Получаем следующий отчет (рис. 5).

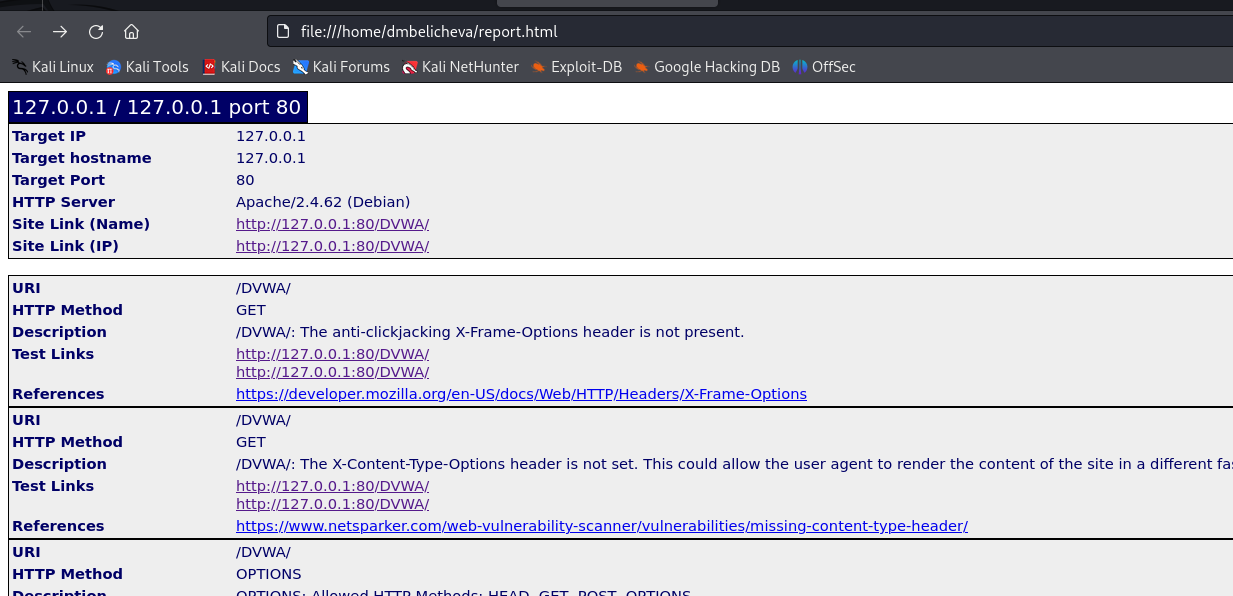


Рис. 5: Отчет об уязвимостях в формате htm

Можем увидеть, что найдены такие уязвимости как отсутствие защиты от кликджекинга, не установлен заголовок X-Content-Type-Options(в связи с чем пользователь может выполнить вредоносный контент не того типа, который предполагает администратор), возможность удаленного доступа к файлам конфигураций, также найдена скрытая папка git, в которой хранятся данные о структуре сайта. Уязвимость типа This might be interesting... означает, что необходимо дополнительная ручная проверка(скорей всего это незначительная уязвимость раскрытия информации – доступен просмотр файлов каталога). В конце отчета указано, что найдено 26 уязвимостей.

Также можно посмотреть информацию об уязвимостях по конкретному порту (в нашем случае порт 80 для локального хоста) (рис. 6).

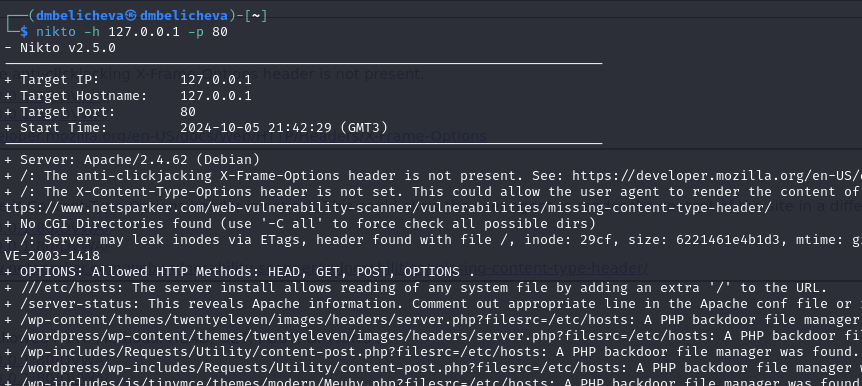


Рис. 6: Проверка уязвимостей с указанием порта

# 4 Выводы

В результате выполнения работы был использован сканер Nikto для сканирования уязвимостей веб-приложения.

# Список литературы

1. Обзор сканера Nikto для поиска уязвимостей в веб-серверах [Электронный ресурс]. 2023. URL: <https://habr.com/ru/companies/first/articles/731696/>.