Индивидуальный проект - этап 4

Использование nikto

Содержание

# 1 Цель работы

Целью данной работы является изучение сканера уязвимостей nikto.

# 2 Введение

## 2.1 Nikto: Описание

**Nikto** — это популярный сканер веб-серверов с открытым исходным кодом, который проверяет веб-серверы на наличие уязвимостей, неправильных настроек, устаревших версий ПО и прочих проблем безопасности.

Основные задачи Nikto:

* Поиск общих уязвимостей веб-серверов.
* Проверка наличия опасных файлов и конфигураций.
* Выявление устаревших версий веб-серверов и их компонентов.
* Определение серверных технологий и модулей.

Особенности:

* Поддержка множества серверов и протоколов (HTTP, HTTPS, HTTP/2 и другие).
* Возможность добавления собственных правил для обнаружения уязвимостей.
* Регулярные обновления базы данных уязвимостей.

Nikto — это пассивный сканер, и он не пытается активно взламывать систему, а только собирает информацию о потенциальных уязвимостях.

Рекомендуется использовать Nikto в сочетании с другими инструментами безопасности, такими как Nmap и OpenVAS, для более полного анализа безопасности веб-сервера.

## 2.2 Полезные параметры и примеры

Nikto написан на Perl, и для его работы необходимо наличие Perl на системе.

Сканирование веб-сервера

perl nikto.pl -h <URL>

Сканирование определенного порта

perl nikto.pl -h <URL> -p <port>

Вывод результатов в файл

perl nikto.pl -h <URL> -o output.txt

Дополнительные аргументы:

* -ssl — принудительное использование SSL (HTTPS).
* -no\_ssl — игнорирование SSL-сертификатов.
* -Tuning — настройка интенсивности сканирования (например, отключение проверки директорий).
* -Plugins — выбор определенных плагинов для сканирования.
* -timeout — установка таймаута для запросов.

# 3 Выполнение лабораторной работы

Nikto может использоваться для пассивного сканирования DVWA, выявления базовых уязвимостей и проверок на неправильную конфигурацию.

Когда DVWA запущено, мы можем использовать Nikto для сканирования. Основной командой для сканирования будет:

perl nikto.pl -h http://localhost/dvwa/

## 3.1 Сканирование localhost

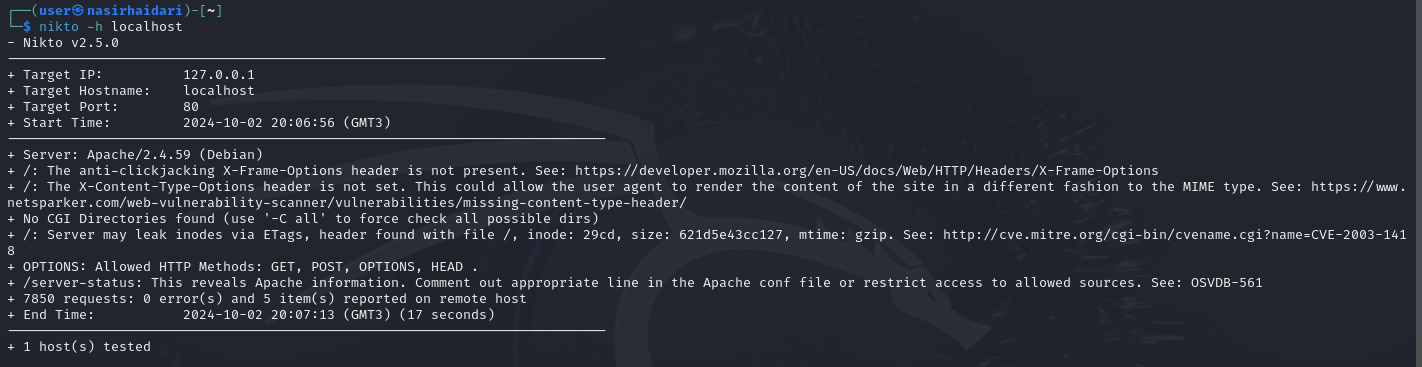


Figure 1: Тестирование localhost

Отчет сканирования:

* IP-адрес цели: 127.0.0.1
* Имя хоста: localhost
* Порт: 80
* Путь: /dvwa
* Сервер: Apache/2.4.59 (Debian)
* Количество запросов: 7849
* Ошибок: 0
* Количество обнаруженных проблем: 3

Детализация обнаруженных проблем:

* Отсутствие заголовка X-Frame-Options: На странице / отсутствует заголовок X-Frame-Options. Это делает сайт уязвимым к атакам Clickjacking, позволяющим злоумышленникам внедрять сайт в iframe на другой странице.
* Отсутствие заголовка X-Content-Type-Options: На странице / отсутствует заголовок X-Content-Type-Options. Это может позволить браузеру отображать содержимое не в соответствии с его MIME-типом, что повышает риск XSS-атак.
* Утечка inode через заголовок ETag: Сервер может утекать идентификаторы inode через заголовки ETag. Это может быть использовано для определения версий файлов и атак типа кэш-поиска.
* Разрешенные HTTP-методы: Разрешены HTTP-методы OPTIONS, HEAD, GET, POST. Хотя эти методы могут быть необходимы для работы сайта, их наличие открывает возможность для злоумышленников собирать информацию о поддерживаемых сервером методах.
* Открытый доступ к /server-status: Путь /server-status открыт для всех, что раскрывает информацию о сервере Apache (включая информацию о процессах и клиентах).

## 3.2 Сканирование localhost/dvwa/



Figure 2: Тестирование localhost/dvwa/

Отчет сканирования:

* IP-адрес цели: 127.0.0.1
* Имя хоста: localhost
* Порт: 80
* Путь: /dvwa
* Сервер: Apache/2.4.59 (Debian)
* Количество запросов: 7849
* Ошибок: 0
* Количество обнаруженных проблем: 3

Детализация обнаруженных проблем:

* Отсутствие заголовка X-Frame-Options: На странице /dvwa/ отсутствует заголовок X-Frame-Options. Это позволяет злоумышленникам внедрять сайт в iframe на других сайтах, что может привести к атакам Clickjacking.
* Отсутствие заголовка X-Content-Type-Options: На странице /dvwa/ отсутствует заголовок X-Content-Type-Options. Это может позволить браузеру обработать файл не в соответствии с его MIME-типом, что может привести к неправильной интерпретации содержимого.
* Разрешенные HTTP-методы: Разрешены методы OPTIONS, HEAD, GET, POST. Хотя сами по себе эти методы не уязвимы, наличие метода OPTIONS может предоставить злоумышленникам дополнительную информацию о поддерживаемых сервером HTTP-методах.

# 4 Вывод

Мы изучили возможности сканера nikto.