Отчёт

об уязвимостях WEB-приложения

Juice Shop

1. OWASP Juice Shop — это веб-приложение с открытым исходным кодом для демонстрации самых распространенных уязвимостей. Приложение написано на Node.js, Express and Angular.
2. Сканирование TOP 10 OWASP командой

semgrep scan –config “p/owasp-top-ten” juice-shop/

показало наличие 41 вероятной уязвимости по 1106 правилам проверки. Полный отчёт представлен в файле report10.txt.

Результат сканирования стандартным набором правил Semgrep (--config auto) представлен в файле semgrep.txt и содержит 84 обнаружения.

1. Уязвимости
   1. A03:2021 - Injection

CWE-89: SQL Injection

juice-shop/routes/login.ts line 36

Для предотвращения использовать подготовленные запросы и проверку содержимого запроса при вводе до отправки в базу.

* 1. A07:2021 – Identification and Authentication Failures

CWE-384: Session Fixation:

juice-shop/lib/insecurity.ts line 201

Для предотвращения закрывать сессию до входа нового пользователя.

CWE-798: Use of Hard-coded Credentials

juice-shop/lib/insecurity.ts line 23

Для предотвращения хранить ключи в отдельных зашифрованных базах данных, для первого входа использовать режим «First login» с требование установки сложного пароля.

* 1. A01:2021 - Broken Access Control

CWE-22: Improper Limitation of a Pathname to a Restricted Directory

juice-shop/routes/dataErasure.ts line 69

Для предотвращения производить проверку ввода согласно спецификации поля на стороне клиента и сервера.

* 1. A10:2021 - Server-Side Request Forgery (SSRF)

CWE-918: Server-Side Request Forgery (SSRF)

juice-shop/routes/profileImageUrlUpload.ts line 23

Для предотвращения проверка и очистка пользовательских запросов, использование списка белых адресов для подключения, ограничение допустимых запросов к сторонним ресурсам.

* 1. 07:2017 - Cross-Site Scripting (XSS) / A03:2021 – Injection

CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

juice-shop/frontend/src/app/search-result/search-result.component.ts line 151

Для предотвращения использовать архитектуру и библиотеки устойчивые к уязвимости. Дополнительно разделять код и данные.

* 1. A04:2021 - Insecure Design

CWE-73: External Control of File Name or Path

juice-shop/routes/fileServer.ts line 33

Для предотвращения вместо имён файлов использовать кодовые обозначения, запускать в «песочнице».

1. Эксплуатация
   1. A03:2021- Injection

Login admin: Пробуем войти с данными admin/123 -> Перехватываем запрос -> Подставляем в поле логина возможные SQL инъекции перебором из файла auth\_bypass.txt -> Выбираем понравившуюся «admin’ or ‘1’=’1’» -> входим

(A03-admin-inj-0.png & A03-admin-inj-1.png)

* 1. A01:2021 - Broken Access Control

Admin section: Открываем инспектор страницы в браузере -> В Sources в файле main.js находим “path:” -> переходим в Admin Section. (A01-admin-sect.png)

View basket: Открываем корзину и смотрим запрос в BurpSuite -> запрос GET /rest/basket/1 HTTP/1.1 меняем на GET /rest/basket/2 HTTP/1.1 -> попадаем в чужую корзину

* 1. A07:2017 - Cross-Site Scripting (XSS) / A03:2021 – Injection

Bonus payload: В строку поиска вводим

<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto\_play=true&hide\_related=false&show\_comments=true&show\_user=true&show\_reposts=false&show\_teaser=true"></iframe>

И получаем выполнение скрипта со стороннего сайта.

(A03-XSS.png)