Внутрішній тест на проникнення

Доповідь по результатам

«Juice Shop»

3-16 липня 2023р.

V.1.0

ЗМІСТ

ПІДХІД

Тест на проникнення був проведений за підходом «чорної скриньки» з 3 по 16 липня 2023 року без облікових даних або будь-яких попередніх знань про внутрішнє середовище Juice Shop з метою виявлення невідомих слабких місць. Тестування проводилося з метою виявлення якомога більшої кількості неправильних конфігурацій і вразливостей. Тестування проводилося локально, на робочій станції де й запущений сайт онлайн-магазину. Кожна виявлена слабкість була задокументована та вручну досліджена для визначення можливостей використання та потенціалу ескалації.

Інформація про осіб хто проводив тестування

Ім’я	E-mail
Роман Сидоренко	roma18sidor@gmail.com

Об'єкт досліджень

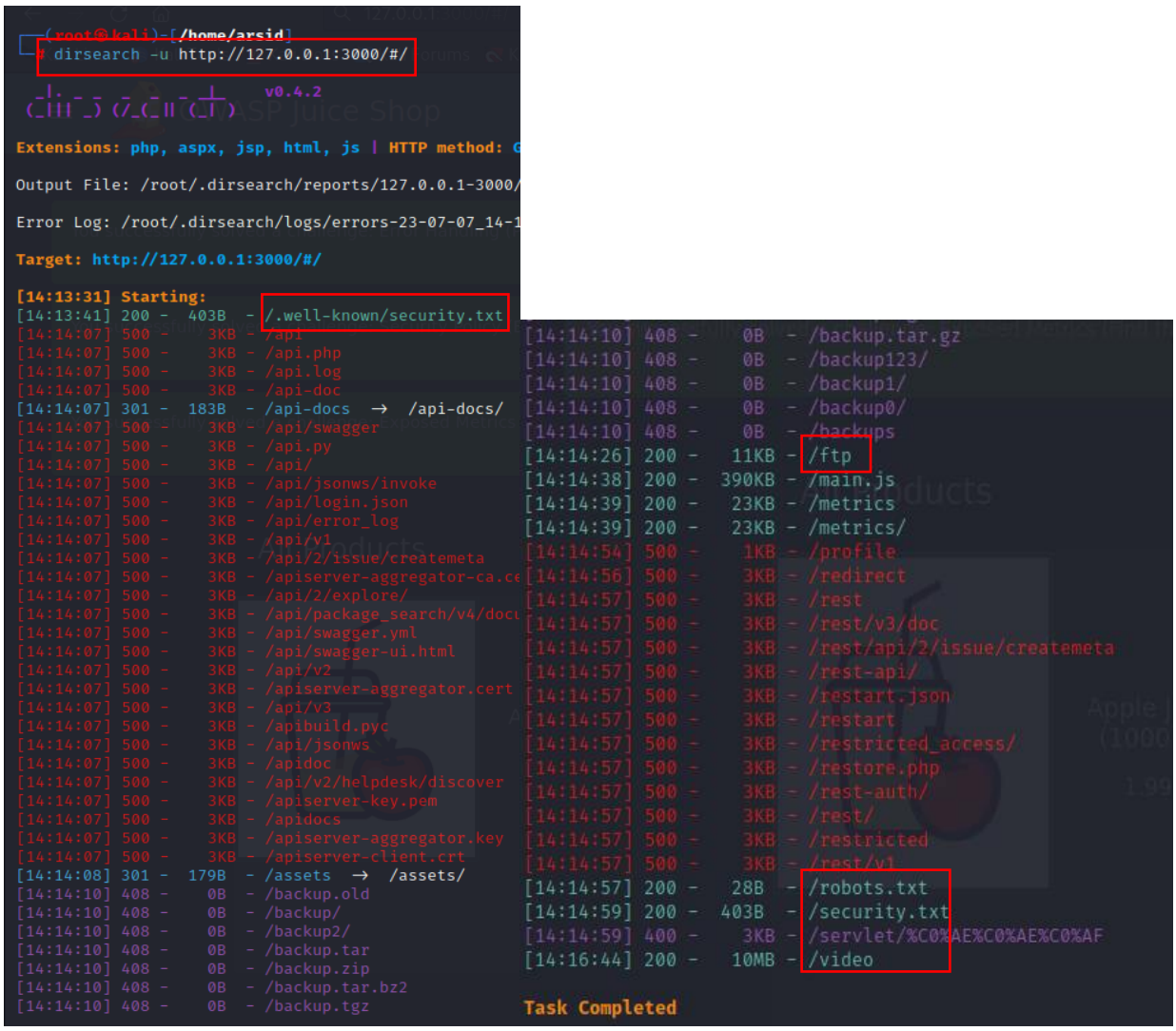
Host	Опис
http://127.0.0.1:3000/#/	Сайт онлайн-магазину Juice Shop

РЕЗУЛЬТАТИ РОБОТИ З ОБ’ЄКТОМ ДОСЛІДЖЕНЬ

Під час внутрішнього тесту на проникнення Juice Shop, командою було виявлено 6 вразливостей, які загрожують конфіденційності, цілісності та доступності інформаційної системи онлайн-магазину Juice Shop. Результати були класифіковані за ступенем важкості, а саме 5 вразливостей критичного рівню та 1 середнього.

2.1 Розкриття конфіденційної інформації

За допомогою програмних засобів сканування операційної системи Kali Linux – dirsearch, під час сканування ip-адреси сайту онлайн-магазину Juice Shop, без використання будь яких специфічних методів, було виявлено існування каталогів та файлів з інформацією, яка можливо не є для загального користування, а саме: .well-known/security.txt, robots.txt, security.txt, ftp.



Файл **.well-known/security.txt**: Цей файл зазвичай використовується для надання інформації про політику безпеки веб-сайту. Однак, якщо вміст файлу не призначений для спільного доступу, його наявність може розкрити конфіденційну інформацію, таку як версії програмного забезпечення, адреси електронної пошти або іншу чутливу інформацію.

Файл **robots.txt**: Файл robots.txt надає інструкції пошуковим системам про те, як індексувати та обробляти веб-сторінки. Якщо небажані файли або каталоги вказані у файлі robots.txt, але вони все ще доступні, це може призвести до розкриття інформації, яка повинна бути прихована від публічного доступу.

Файл **security.txt**: Цей файл може містити інформацію про політику безпеки, контактні дані відповідальних осіб або інструкції щодо повідомлення про вразливість. Якщо цей файл містить конфіденційну інформацію або інформацію, яка має бути доступна лише обмеженому колу осіб, його виявлення може призвести до витоку даних або порушення безпеки.

Каталог **ftp**: Якщо каталог FTP містить файли або ресурси, які не призначені для спільного доступу, і його налаштування дозволяють публічний доступ, це може призвести до несанкціонованого отримання конфіденційних файлів або витоку інформації.

Рівень вразливості: Критичний

Виправлення вразливості:

Short-term план:

- Негайно видалити або перемістити небажані файли та каталоги, виявлені в результаті сканування за допомогою dirsearch.
- Оновити файл "robots.txt" із зазначенням заборони на індексацію та доступ до конфіденційних файлів та каталогів.

Mid-term план:

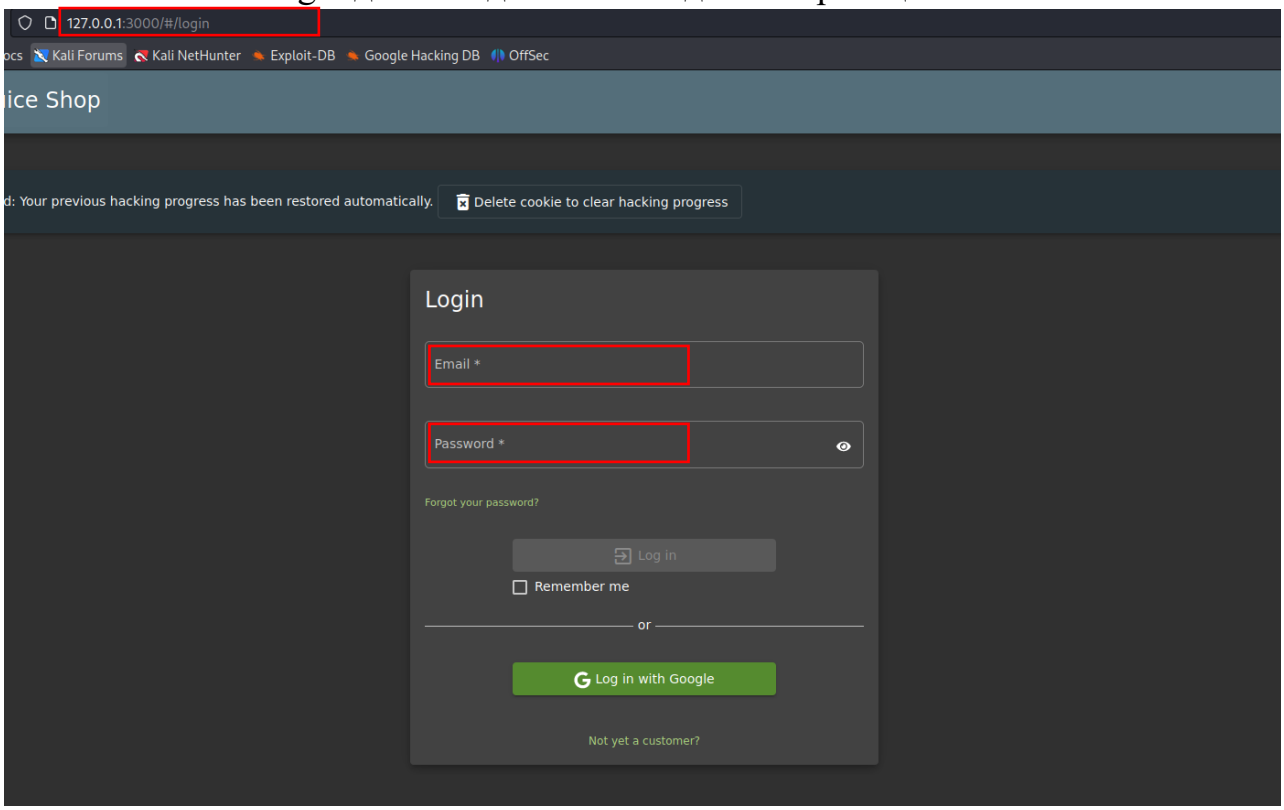
- Впровадити систему автентифікації та авторизації для контролю доступу до конфіденційних файлів та каталогів.
- Перевірити та налаштувати дозволи файлової системи та сервера для забезпечення обмеженого доступу до конфіденційних ресурсів.

Long-term план:

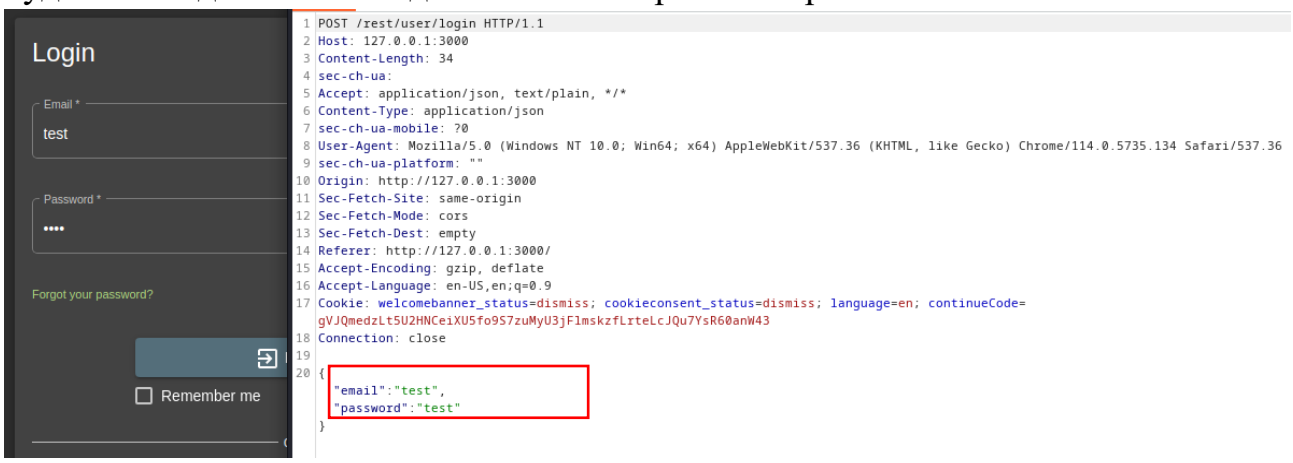
- Провести повний аудит системи для виявлення інших потенційних вразливостей, пов'язаних із доступом до конфіденційної інформації.
- Розробити та впровадити стратегію захисту даних, включаючи шифрування конфіденційної інформації у спокої та під час передачі.
- Провести навчання працівників безпеки, щоб вони розуміли важливість забезпечення безпечного доступу до конфіденційних ресурсів.

2.2 Вразливість до SQL-Injection

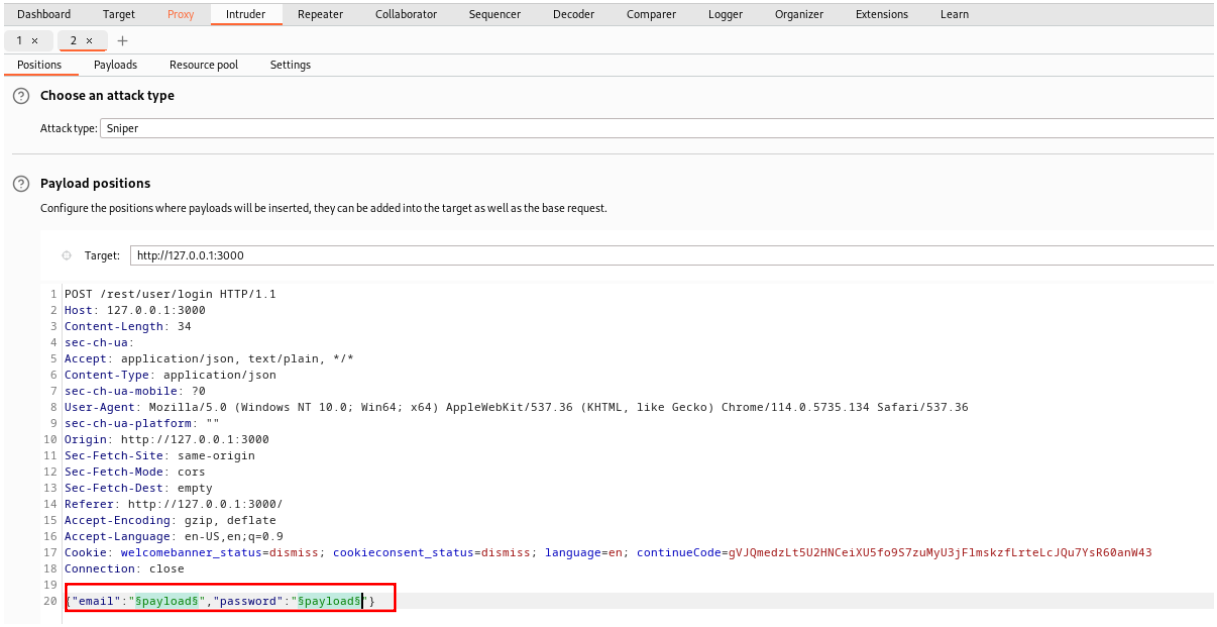
Під час звернення за адресою 127.0.0.1:3000 відбувається редірект за на ресурс 127.0.0.1:3000/#/login де знаходяться поля для авторизації Email та Password.



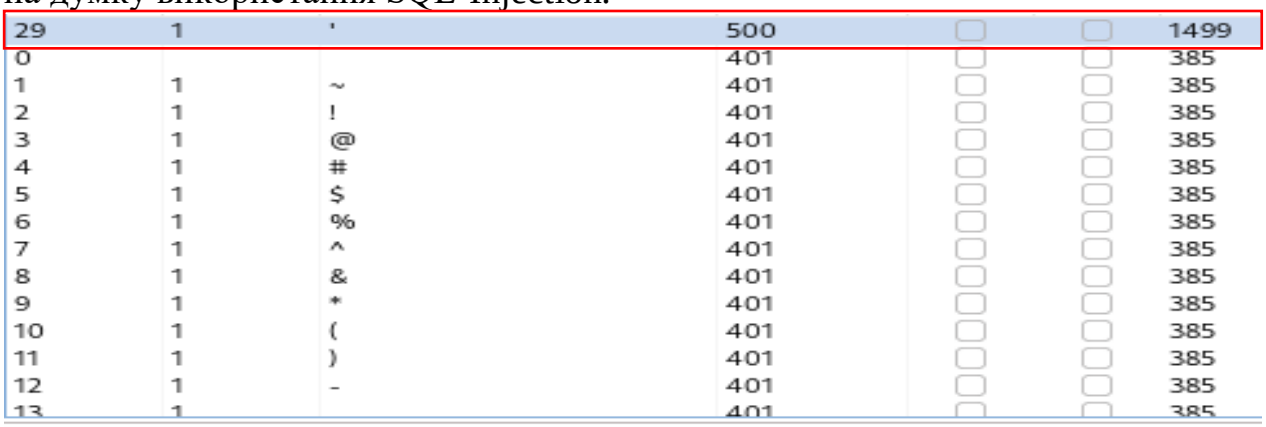
Перевіряємо чи вразливі поля до передачі до них спецсимволів, які в свою чергу можна використовувати для різних атак. В даному випадку заповнюємо поля будь-якими даними та за допомогою Burp Suite перехоплюємо запит.



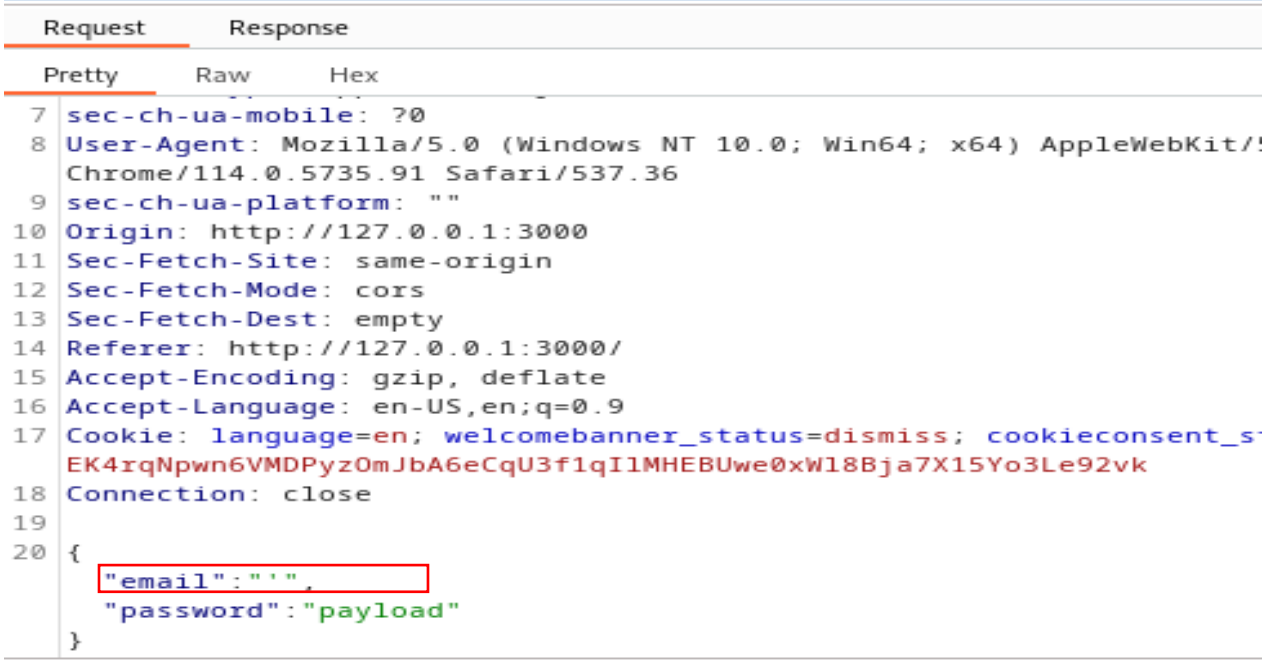
В Burp Suite, перехоплений запит відправляємо до режиму «Intruder» в якому замінюємо значення «test» на «payload» та в самому payload як параметри вказуємо усі спецсимволи(!,»,’,” та інші)



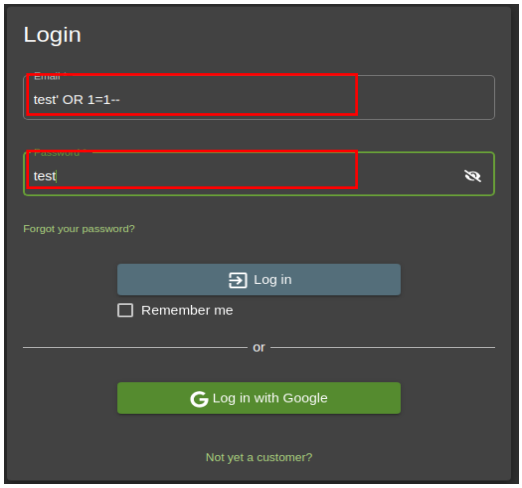
Після проведення атаки, в результаті визначено, що поле Email на передачу до нього спецсимвол «’» відповідає кодом 500, це означає що сервер отримав його, але він не має додаткових умов, що йому робити далі. В свою чергу наштовхує на думку використання SQL-Injection.

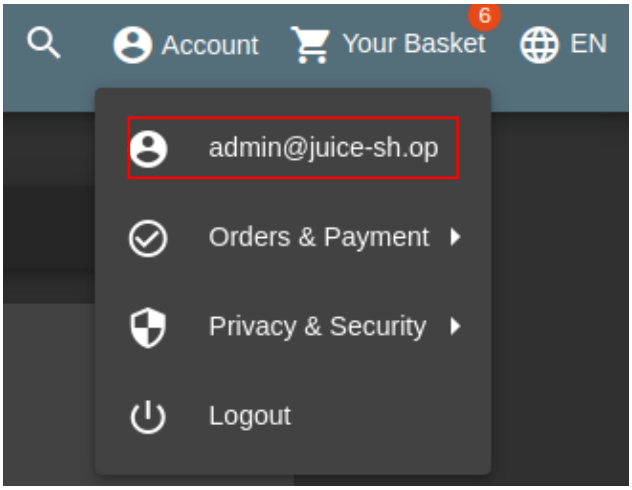


29	1	'	500			1499
0			401			385
1	1	~	401			385
2	1	!	401			385
3	1	@	401			385
4	1	#	401			385
5	1	$	401			385
6	1	%	401			385
7	1	^	401			385
8	1	&	401			385
9	1	*	401			385
10	1	(	401			385
11	1	)	401			385
12	1	-	401			385
13	1		401			385

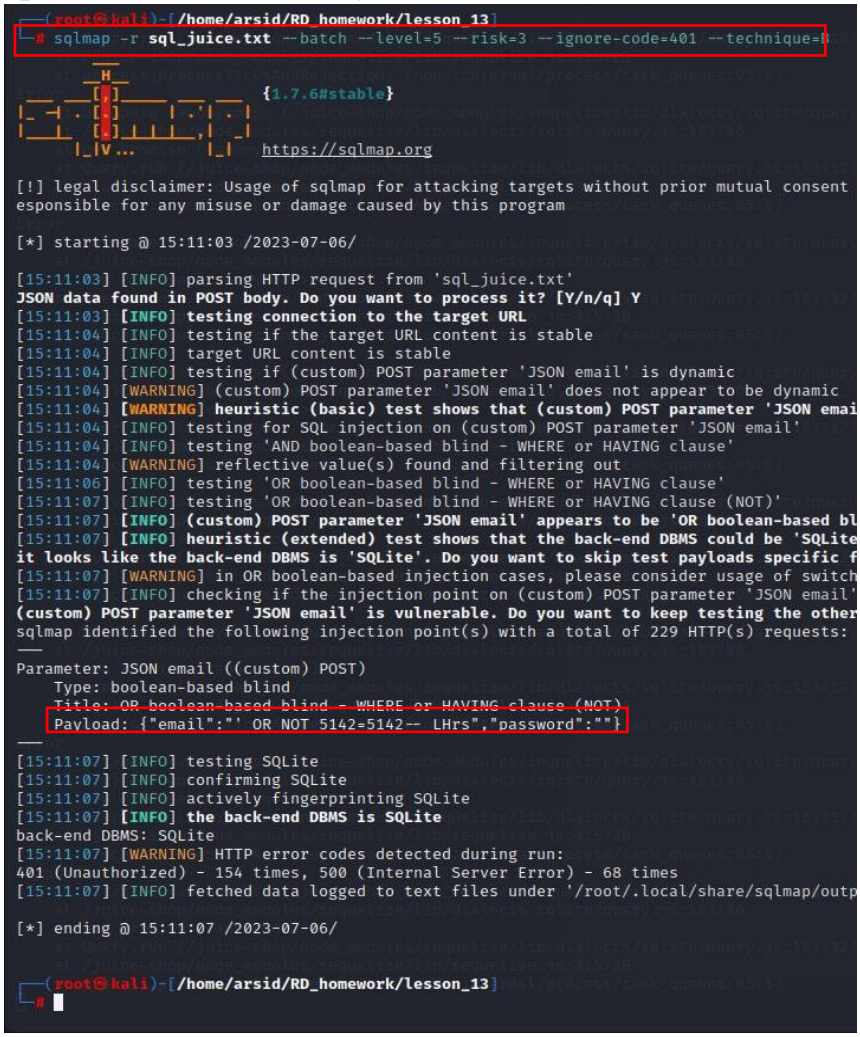


Для тесту використовуємо простий payload «’ OR 1=1--» разом з будь якими даними та отримуємо доступ до облікового запису адміністратора.

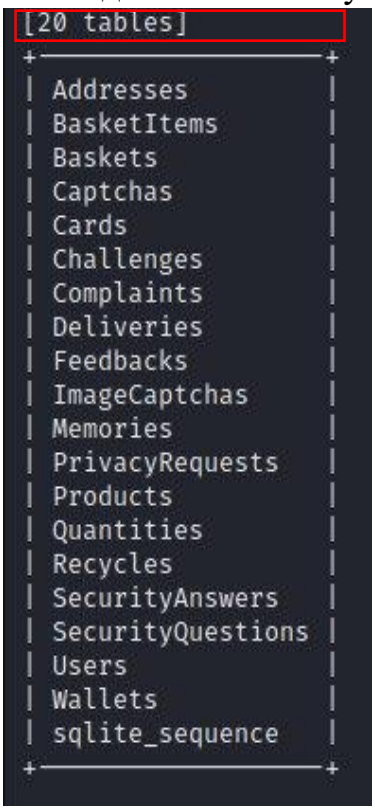




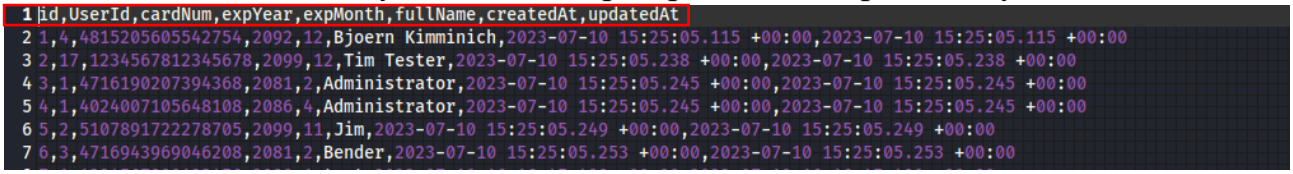
Окрім цього, якщо запустити sqlmap, з перехопленим раніше зазначеним запитом(п.2.3), використовуючи додаткові параметри можна підтвердити вразливість до SQL-Injection Blind-Boolean-based.



А якщо додати параметр –tables можна переглянути список таблиць що знаходяться в базі даних сайту.



Використовуючи отриманий доступ, можна як приклад, отримати доступ до таблиці Cards та витягнути з неї номери кредитних карт та дату їх закінчення.



Рівень вразливості: Критичний

Виправлення вразливості:

Short-term план:

- Негайно застосувати патчі та оновлення системи управління базами даних та використовуваних фреймворків для закриття відомих вразливостей SQL-ін'єкції.
- Обмежити доступ до адміністративних функцій та інструментів, щоб знизити ризик несанкціонованого доступу.

Mid-term план:

- Провести аудит кодової бази та ідентифікувати вразливі точки введення, особливо у зв'язку з обробкою SQL-запитів. Виправити виявлені проблеми, включаючи впровадження параметризованих запитів та використання підготовлених операторів.
- Реалізувати фільтрацію та валідацію введення користувача на стороні сервера, щоб запобігти можливості застосування SQL-Injection.
- Застосовувати принцип найменших привілеїв та встановити обмежені права доступу до бази даних для кожного користувача чи ролі.

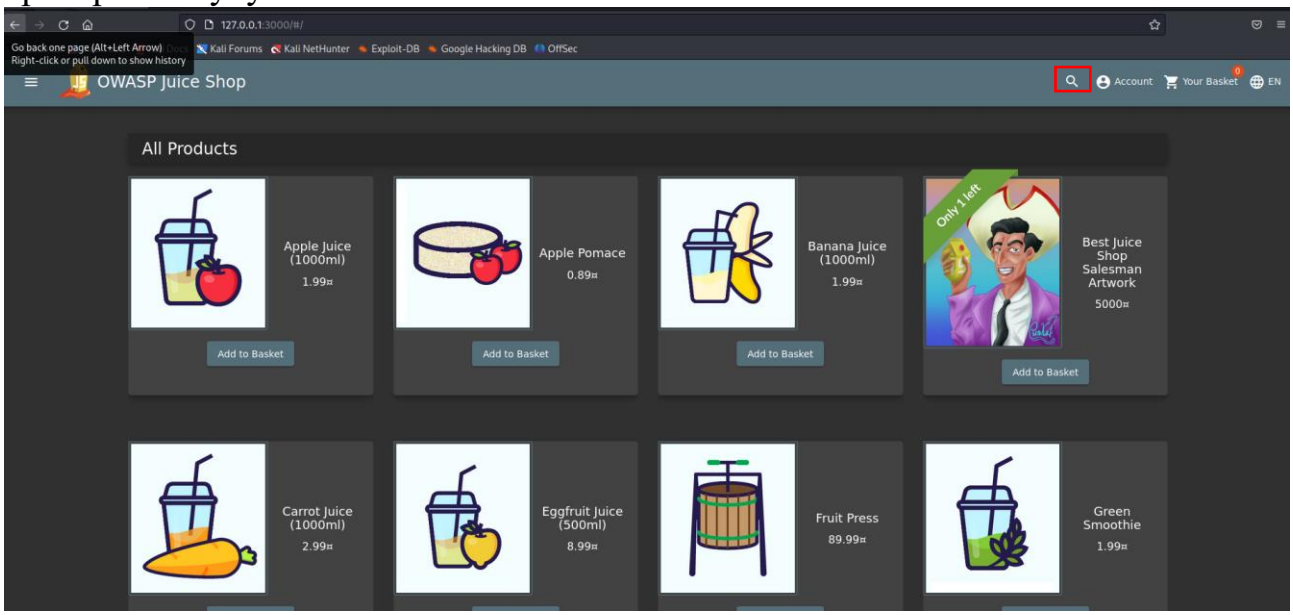
Long-term план:

Включити у процес розробки безпечні практики, у тому числі тестування на проникнення (penetration testing) та регулярний аудит безпеки коду для виявлення та виправлення нових вразливостей.

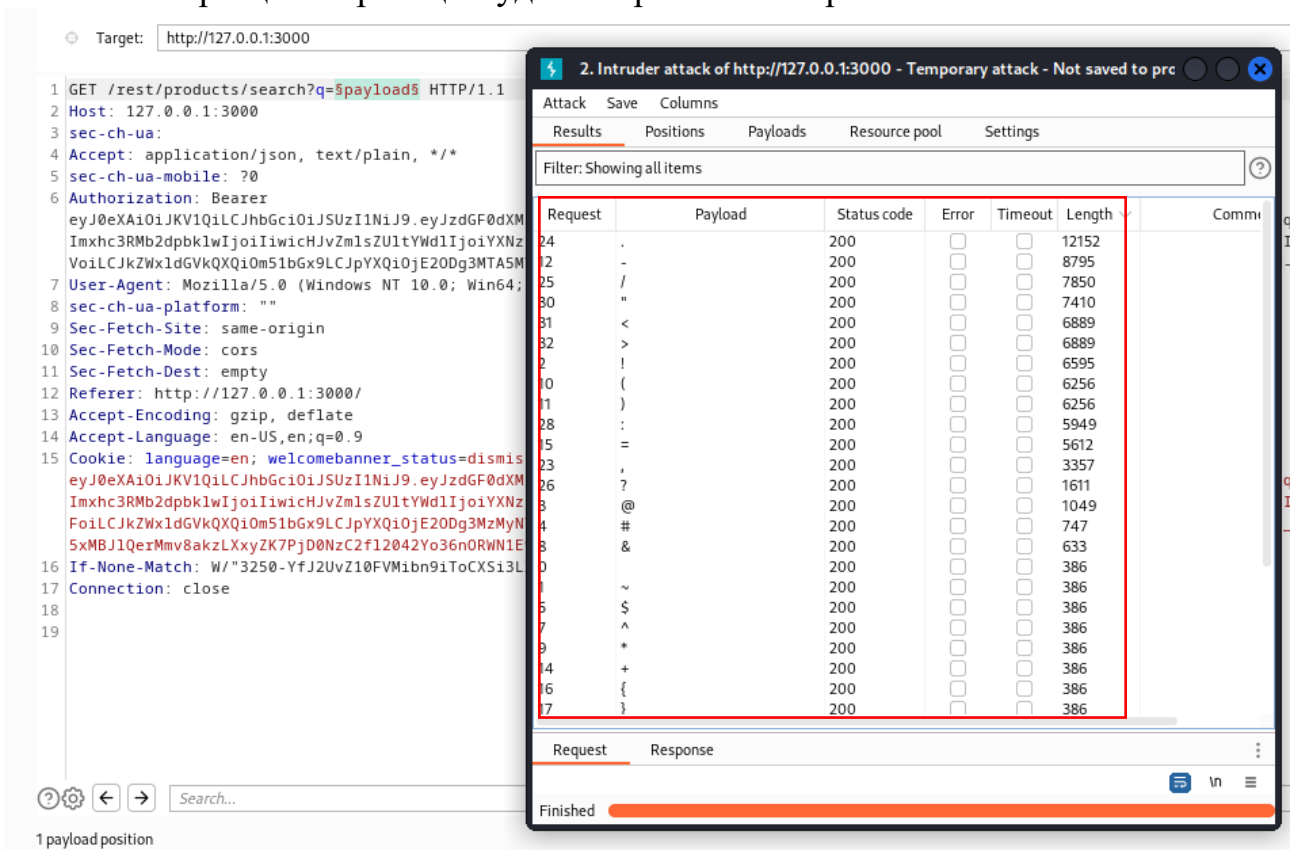
- Навчити розробників та адміністраторів системи про методи запобігання SQL-ін'єкцій та безпечні практики при роботі з базами даних.
- Реалізувати механізми моніторингу та реєстрації, щоб виявляти та реагувати на спроби SQL-ін'єкцій та інші атаки на систему.

2.3 XSS-Injection

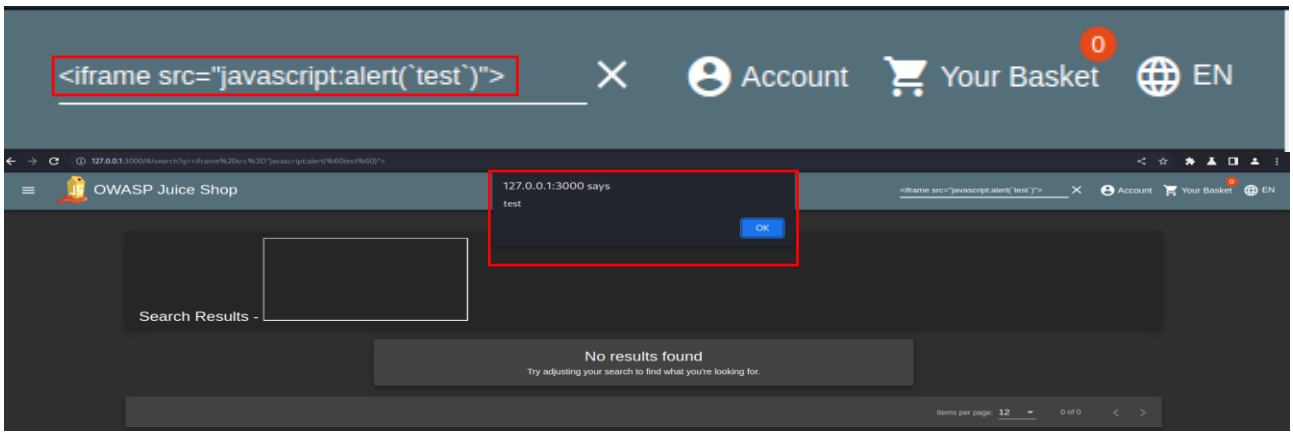
Після авторизації на сайті, у користувача відкривається сторінка за адресою 127.0.0.1/#/search.
На сторінці є поле для швидкого пошуку товарів, в якому користувач вводить критерії пошуку.



Для перевірки можливості використання вразливостей, як і у випадку з полем Email на сторінці авторизації буде використано Burp Suite.



Сервер, через поле пошуку, відповідає кодом 200 на всі спецсимволи які він отримує. В зв'язку з чим існує припущення, що можна використовувати XSS.



Якщо передати JS скрипт до поля пошуку, то на сторінці відбудеться результат роботи скрипту, що можна характеризувати як XSS Reflected Atack.

Рівень вразливості: Критичний

Виправлення вразливості:

Short-term план:

- Використання функцій фільтрації даних перед їх відображенням на веб-сторінці. Наприклад, екранування символів, заміна небезпечних символів на їх відповідники тощо.
- Встановлення правил безпеки на сервері та веб-фреймворці для блокування потенційно шкідливих запитів або введення даних.

Mid-term план:

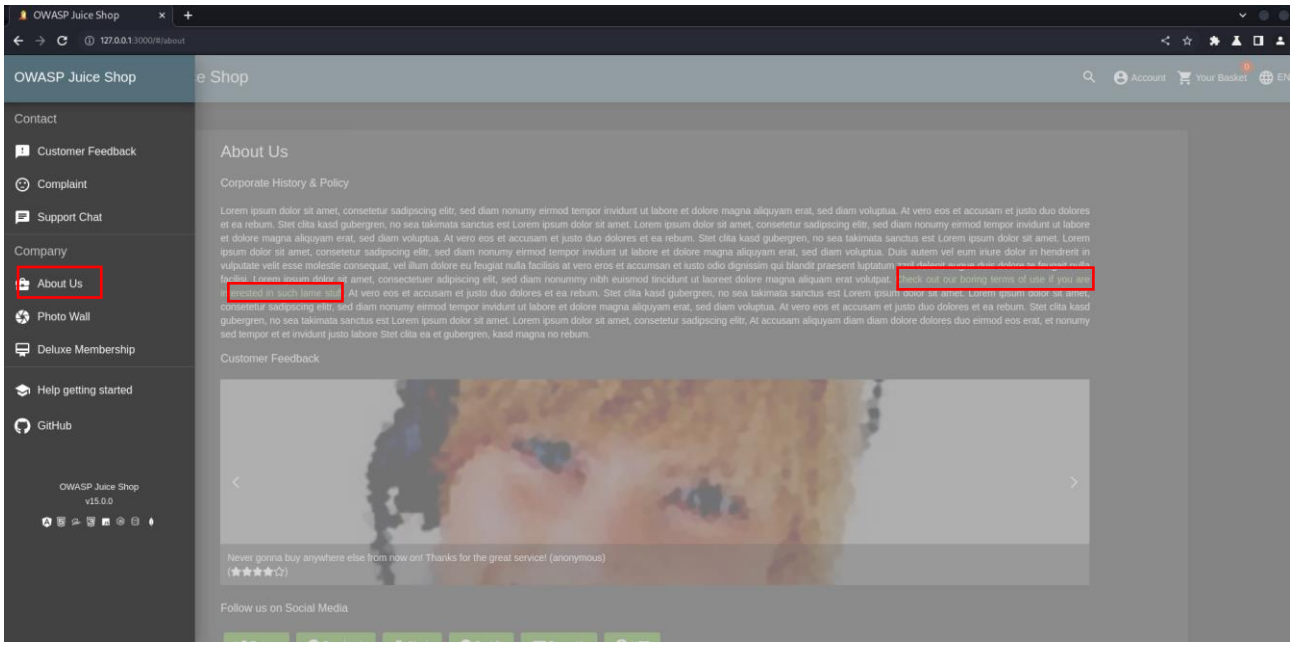
- Забезпечення валідності даних на сервері. Перевірка введених користувачів даних на наявність небезпечних символів або шаблонів перед збереженням бази даних.
- Використання контекстної екранізації при вставці даних у різні контексти (HTML, CSS, JavaScript тощо).
- Регулярні оновлення та патчі для серверного ПЗ та веб-фреймворків, щоб відправити відомі вразливості.

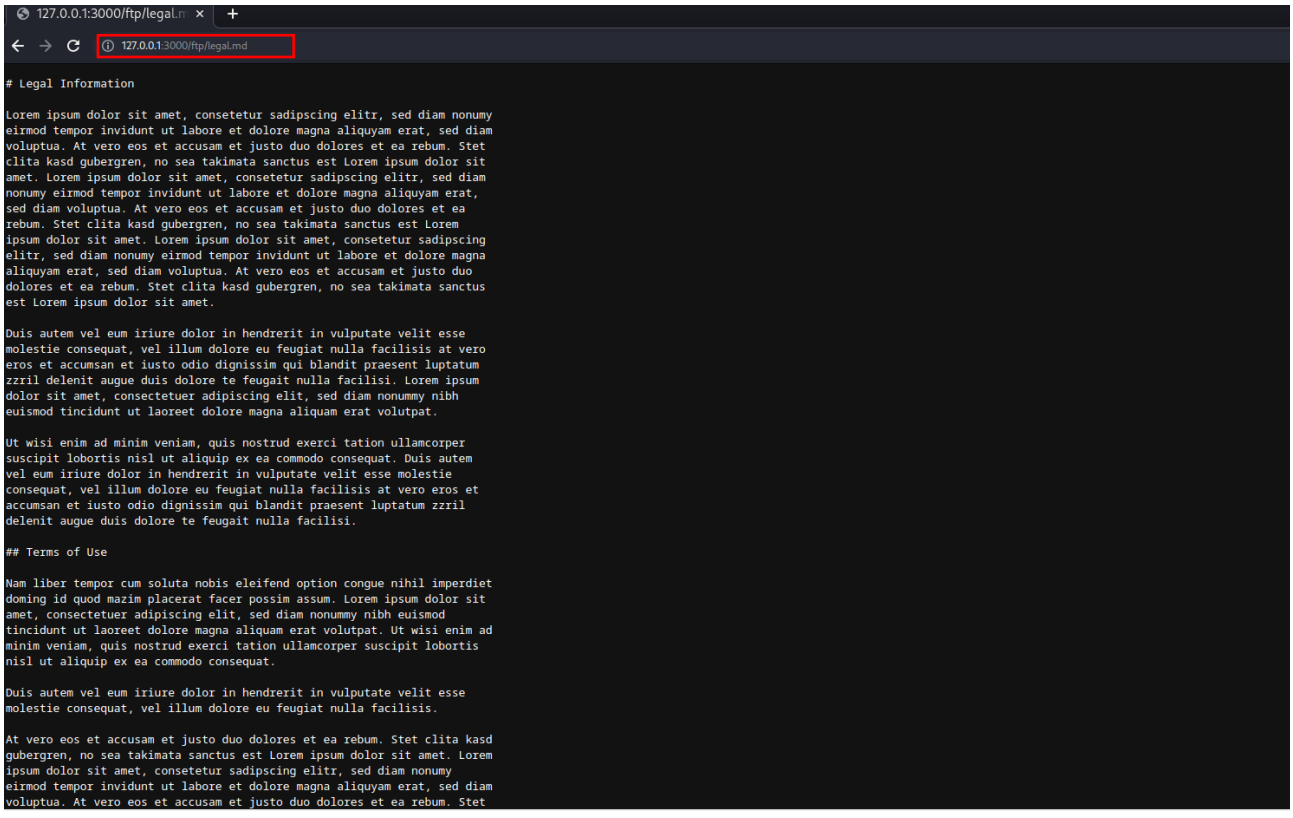
Long-term план:

- Навчання розробників безпечної розробки та освідченості про потенційні вразливості XSS.
- Використання політики безпеки вмісту (CSP) для обмеження та контролю інших джерел завантаження скриптів та вмісту на веб-сторінці.
- Усунення вразливостей XSS вимагає комплексного підходу та комбінації різних методів на різних часових відрізках. Важливо вжити заходів як на рівні додатків коду, так і на рівні серверної інфраструктури для забезпечення безпеки.

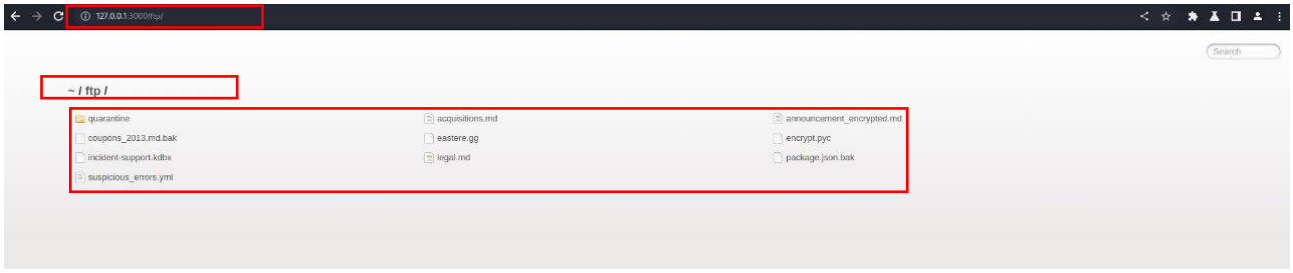
2.4 Неправильне налаштування переадресації та розкриття інформації

На сторінці «About Us» знаходиться посилання на умови користування сайтом, під час переходу за посиланням відбувається редірект за адресою 127.0.0.1/#/ftp/legal.md.

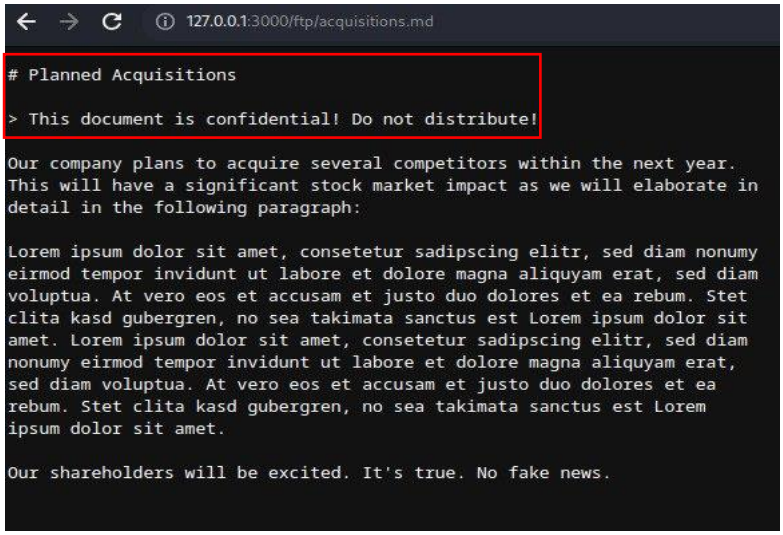




В пошуковому рядку видно, що доступ до фалу legal.md знаходиться у каталозі …/ftp/. Якщо видалити в пошуковому рядку legal.md то користувач потряпляє до каталога /ftp/ де знаходиться інформація яка ймовірно не повинна бути у відкритому доступі.



Як приклад, з’являється доступ до файлу asqisitions.md, який згідно інформації в ньому є конфіденційним.



Рівень вразливості: Критичний

Виправлення вразливості:

Short-term план:

- Введіть перевірку та фільтрацію введення користувача перед використанням його в процесі переадресації. Переконайтеся, що адреса переадресації відповідає очікуваним значенням або шаблонам URL.
- Використовуйте політики переадресації, які обмежують перенаправлення лише на довірені веб-сайти та запобігають перенаправленню на локальні чи внутрішні ресурси.

Mid-term план:

- Розгляньте можливість зміни дизайну та функціональності, щоб умови користування могли бути надані безпосередньо на сторінці "About Us" без необхідності переадресації користувача на зовнішній ресурс.
- Оновіть документацію та навчіть розробників про безпечні методи переадресації та обробки посилань.

Long-term план:

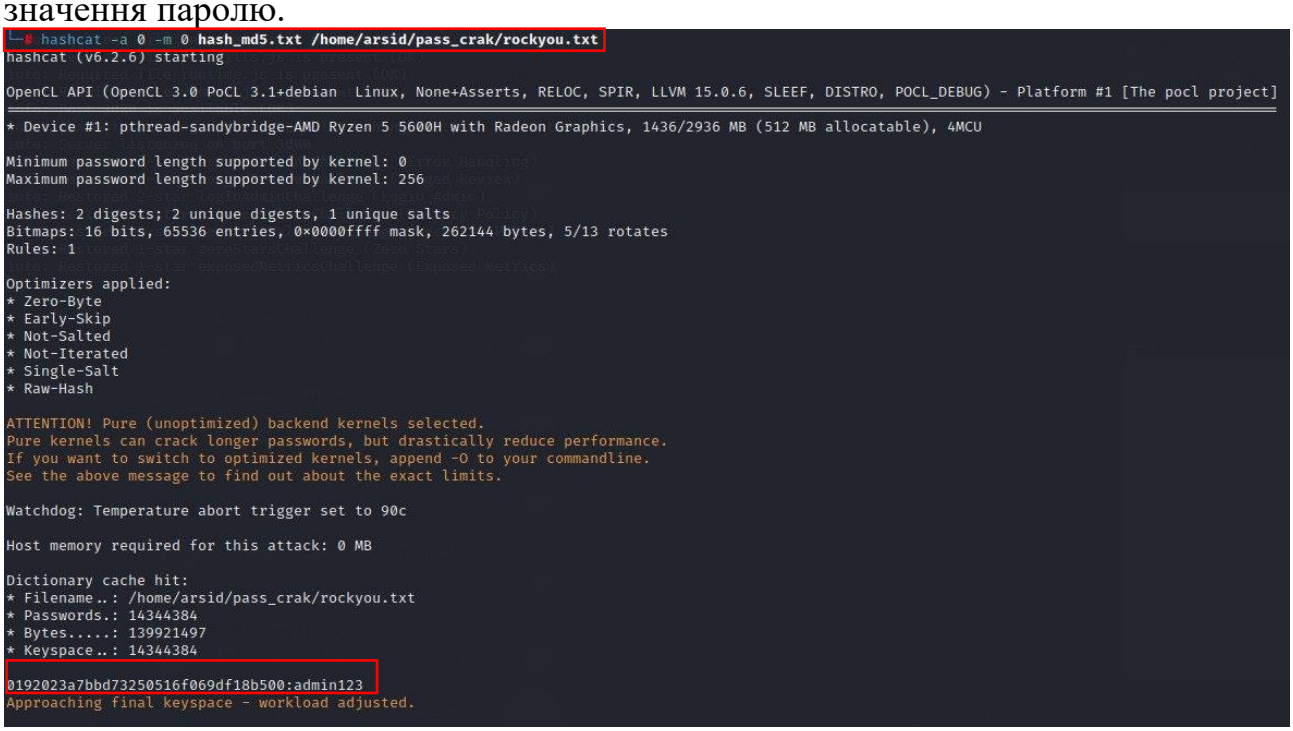
- Проведіть ретельний аудит безпеки програми, щоб виявити інші можливі вразливості та проблеми безпеки. Використовуйте автоматизовані інструменти та проведіть ручний аналіз коду.

- Проведіть навчання розробників про безпечну розробку та поінформованість про потенційні вразливості. Забезпечте регулярні оновлення та навчальні матеріали для підвищення поінформованості про безпеку.
- Застосовуйте рекомендації з безпеки, специфічні для вашої платформи та фреймворку. Оновлюйте використовуване ПЗ, щоб усунути відомі вразливості.

2.5 Використання слабких алгоритмів шифрування та паролів.

Використовуючи SQL ін'єкцію для авторизації під обліковим записом адміністратора, під час перегляду перехоплених запитів було виявлено в JSON Web Token параметр «password» формат якого нагадує алгоритм шифрування MD5(порівняння здійснювалось за допомогою ресурсу - https://hashcat.net/wiki/doku.php?id=example_hashes)

Застосувавши hashcat, використовуючи метод 0(для алгоритму MD5) та словник з переліком найпоширеніших паролів(rockyou.txt), було визначено оригінальне значення паролю.



Рівень вразливості: Критичний

Виправлення вразливості:

Short-term план:

- Замініть слабкі алгоритми шифрування на сучасні та надійні алгоритми, такі як AES (Advanced Encryption Standard) для шифрування даних. Уникайте використання застарілих алгоритмів, таких як DES або MD5.
    - Замість зберігання паролів у вигляді тексту, хешуйте їх за допомогою сильних хеш-функцій, таких як bcrypt або Argon2. Це дозволить зберігати паролі в зашифрованому вигляді, запобігаючи їхньому прямому розкриттю.

Mid-term план:

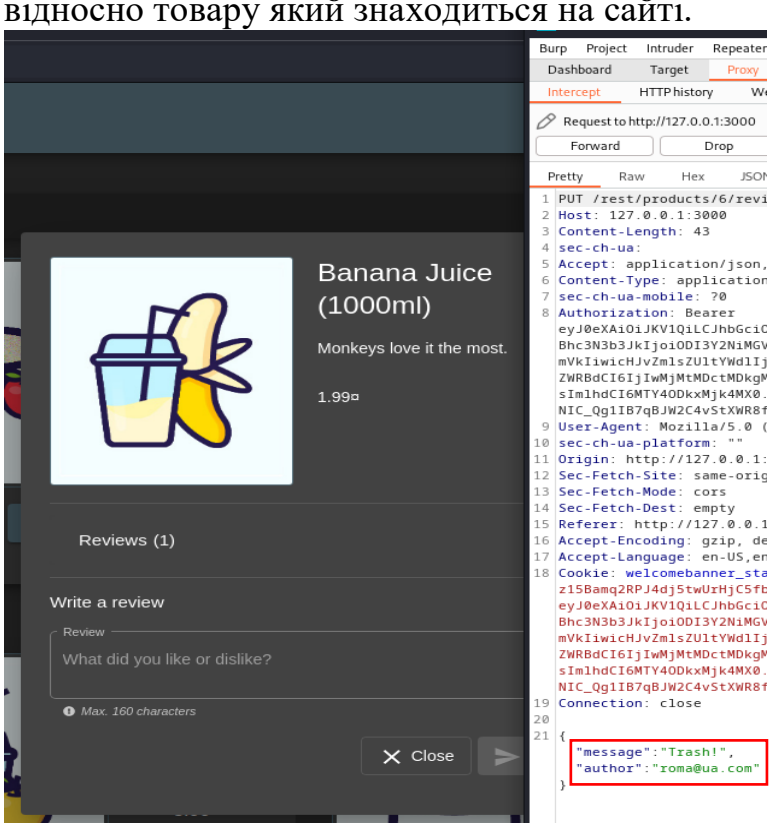
- Якщо веб-додаток вже використовує слабкі алгоритми шифрування або зберігає паролі в незахищеному вигляді, оновіть існуючі дані, перешифруючи або хешуючи їх за допомогою сильних алгоритмів та методів зберігання.
- Введіть вимоги до складності паролів, щоб користувачі створювали паролі, які містять комбінацію символів верхнього та нижнього регістру, цифр та спеціальних символів. Це допоможе посилити захист паролів від злому шляхом перебору.

Long-term план:

- Забезпечте навчання користувачам щодо безпечного створення паролів та їх важливості. Рекомендуйте використовувати унікальні паролі для кожного веб-сервісу та не ділитися ними з іншими людьми.
- Слідкуйте за оновленнями для використаних алгоритмів шифрування та методів хешування. Проводьте регулярні аудити безпеки, щоб виявити та виправити потенційні вразливості.
- Розгляньте впровадження двофакторної аутентифікації, щоб посилити безпеку входу до системи. Це додає додатковий рівень захисту, вимагаючи від користувачів надати додаткову форму ідентифікації, окрім пароля.

2.6 Підробка ідентифікації під час залишення коментарів від імені іншого користувача.

Будучи авторизованим під тестовим обліковим записом roma@ua.com, успішно було здійснено підробку інформації, щодо особи, яка залишила коментар відносно товару який знаходиться на сайті.

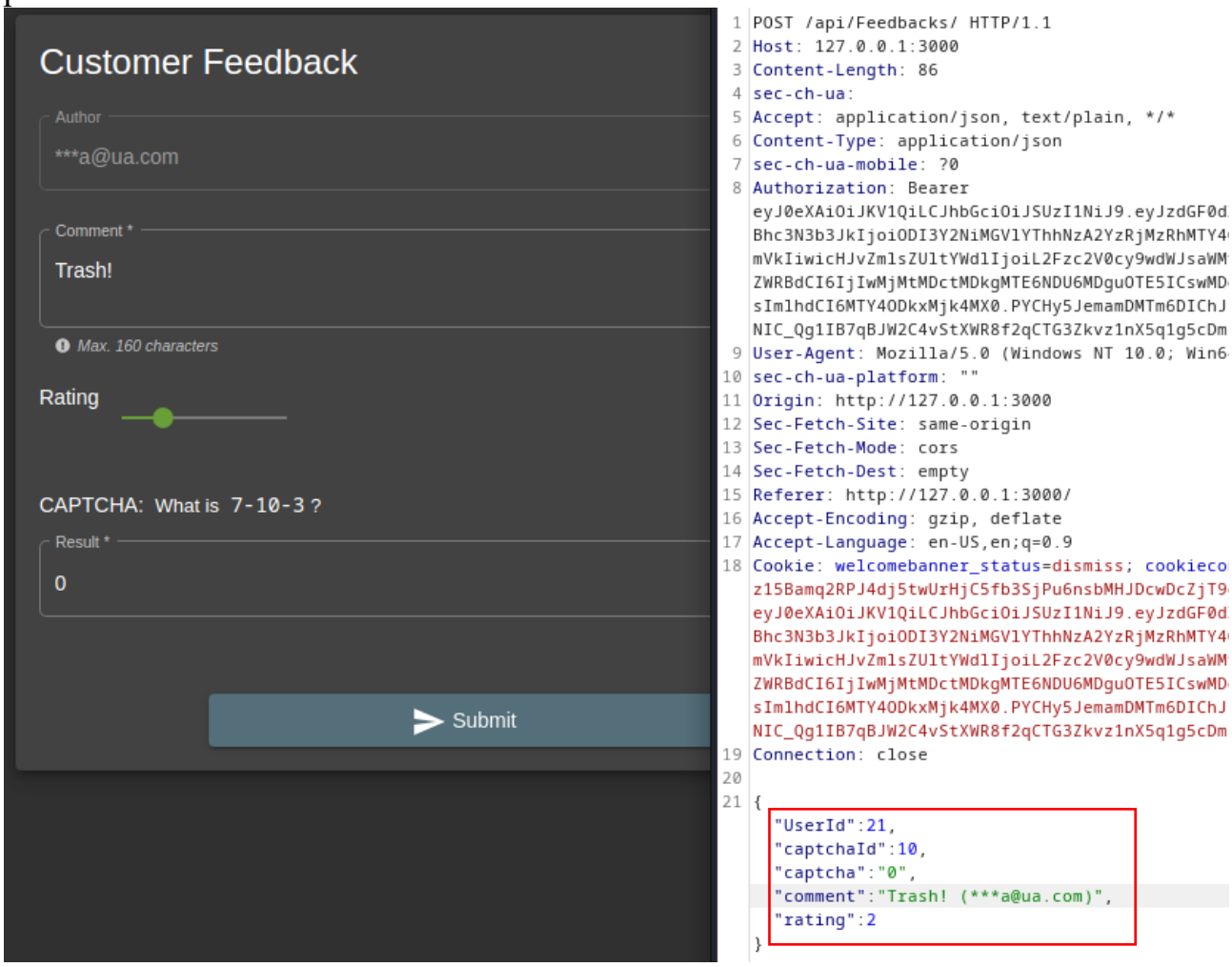


За допомогою Burp Suite було перехоплено запит який відповідає за передачу на сервер інформацію про відгук та електронну пошту користувача.

Для тесту, параметр author був замінений на неіснуючий Bob@pl.com.

В результаті було залишено коментар від імені іншого користувача.

За таким же принципом можна залишити відгук про магазин з низьким рейтингом.



Замінюємо оригінальні данні на фальшиві та ставимо низький рейтинг(Bob@pl.com та 0 відповідно).

Рівень вразливості: Середній

Виправлення вразливості:

Short-term план:

- Перевірте, що система коментування вимагає автентифікації користувача, перш ніж залишати коментар. Переконайтеся, що процес автентифікації надійний і не може бути обхідним шляхом використання вразливостей, таких як підробка сеансу (session hijacking) або слабкі паролі.
- Перед публікацією коментаря переконайтеся, що користувач має відповідні права доступу для залишення коментаря. Перевіряйте ідентифікацію користувача та рівень його авторизації перед дозволом публікації коментаря.

Mid-term план:

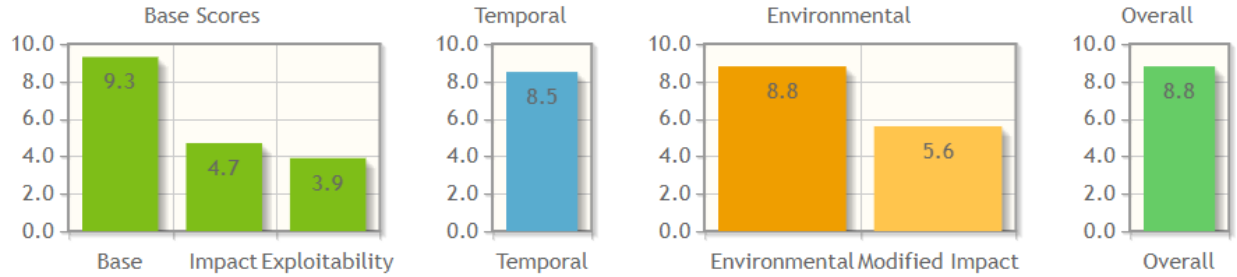
- Використовуйте механізми захисту від підробки міжсайтових запитів (CSRF) для перевірки, що запити на відправлення коментаря походять від дійсного та автентифікованого користувача. Генеруйте та перевіряйте токени CSRF у всіх формах відправлення коментарів.
- Введіть журнал усіх дій, пов'язаних із коментуванням, включаючи інформацію про авторизацію та відправника коментаря. Спостерігайте за активністю в системі, щоб виявити підозрілі чи незвичайні спроби підробки ідентифікації.

Long-term план:

- Оновіть систему авторизації та аутентифікації за допомогою надійних методів і протоколів, таких як двофакторна аутентифікація (2FA) або аутентифікація на основі токенів.
- Навчання користувачів: Проведіть навчання користувачів про безпеку та проблеми підробки ідентифікації. Поясніть їм, як розпізнавати легітимні коментарі та бути уважними до можливих шахрайських дій.

3. Загальні відомості про систему

Використовуючи ресурс nvd.nist.gov, було пораховано та визначено загальний рівень системи в цифровому еквіваленті в розрахунках від 0 до 10, де 0 низький рівень вразливості а 10 найвищий.



Base Scores – параметр відображає оцінку вразливості в ізоляції, незалежно від контексту та оточення. Він вказує на серйозність вразливості на основі кількох метрик, таких як вплив на конфіденційність, цілісність та доступність системи.

Impact – параметр відображає оцінку впливу вразливості на конфіденційність, цілісність та доступність системи. Значення 4.7 вказує на помірний вплив вразливості на систему.

Exploitability - параметр оцінює ймовірність успішної експлуатації вразливості. Значення 3.9 вказує на помірну ймовірність експлуатації вразливості.

Temporal - параметр враховує додаткові фактори, що можуть змінюватися з часом, такі як наявність патчів або поширеність активних зловживань, спрямованих на вразливість. Він відображає поточний стан можливості експлуатації вразливості.

Environmental - параметр враховує контекстні фактори, такі як типи активів, ландшафт загрози, вплив на бізнес та інші фактори. Він відображає оцінку критичності вразливостей в конкретному середовищі. Значення 8.8 вказує на високу критичність вразливостей.

Modified Impact – параметр впливу враховує налаштування та контрольні заходи, що можуть зменшити або збільшити вплив вразливості на систему. Значення 5.6 вказує на середній рівень впливу вразливості після застосування контрольних заходів. Це може бути в результаті налаштування обмежень доступу, впровадження систем контролю безпеки або інших заходів, що зменшують вплив вразливостей на систему.