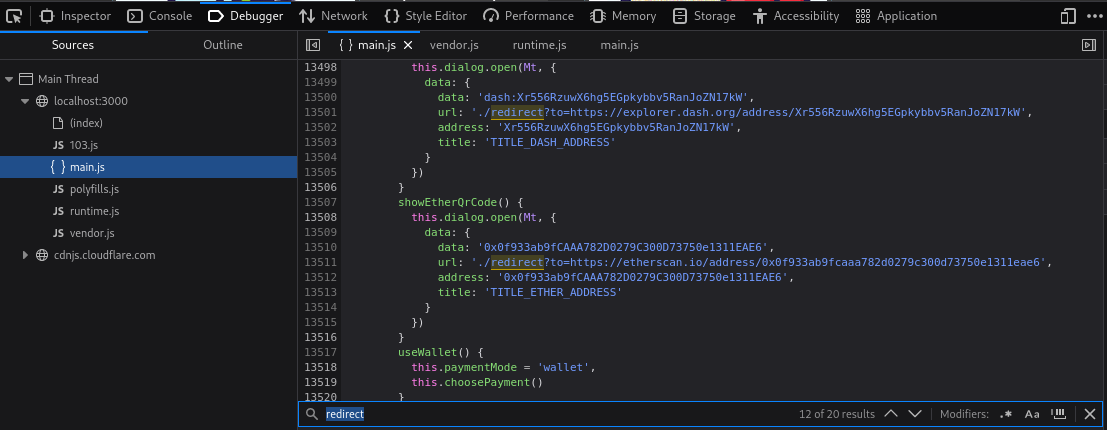
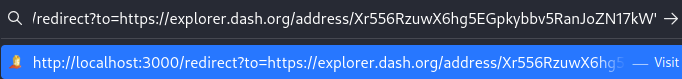
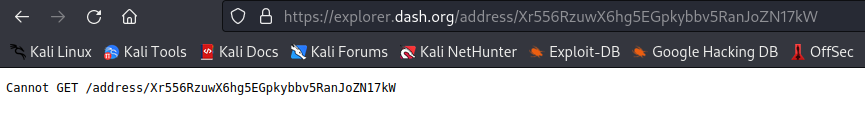
## Allowlist Bypass

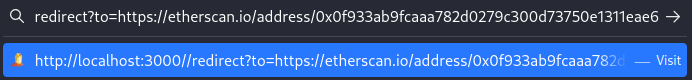
Zadanie jest podobne do wcześniejszego jednogwiazdkowego wyzwania Unvalidated Redirects. Dlatego do jego wykonania przystąpiono w podobny sposób – wyszukując redirect w Debuggerze w DevTools:

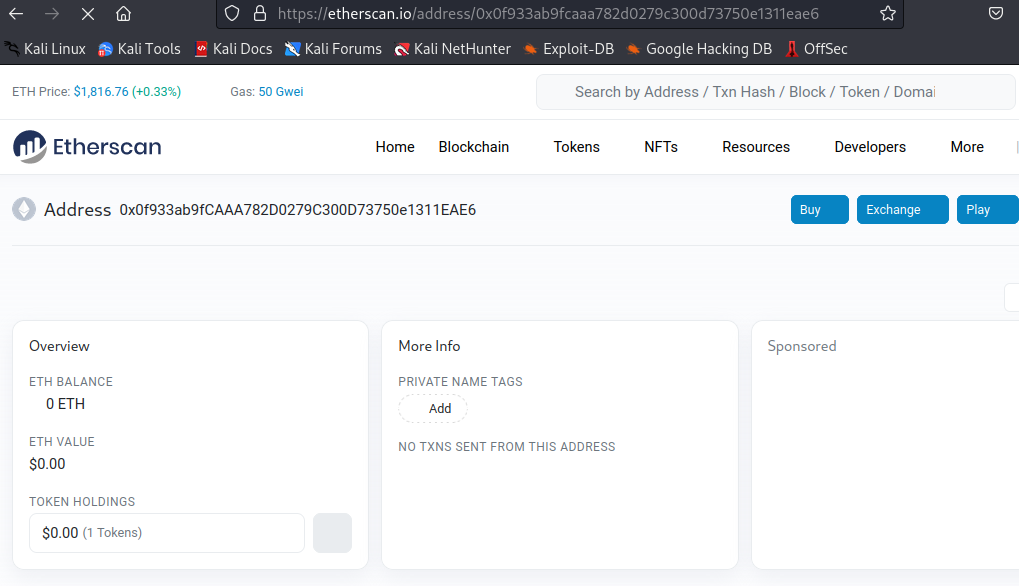


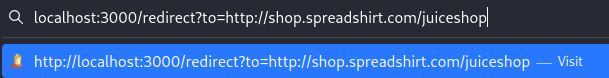
Jak widać, przekierowania można dokonać za pomocą ścieżki /redirect?to=<adres>. Rozpoczęto testowanie wszystkich linków po kolei:

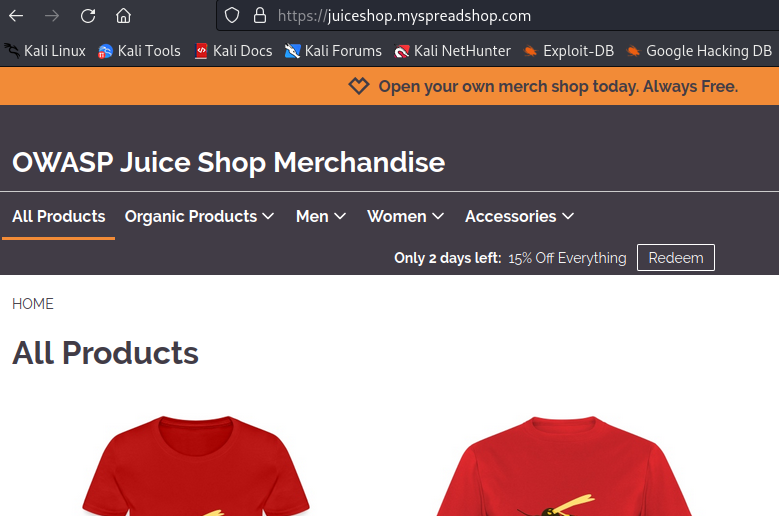


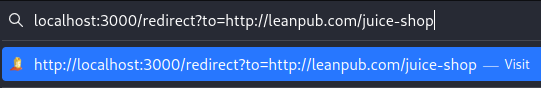


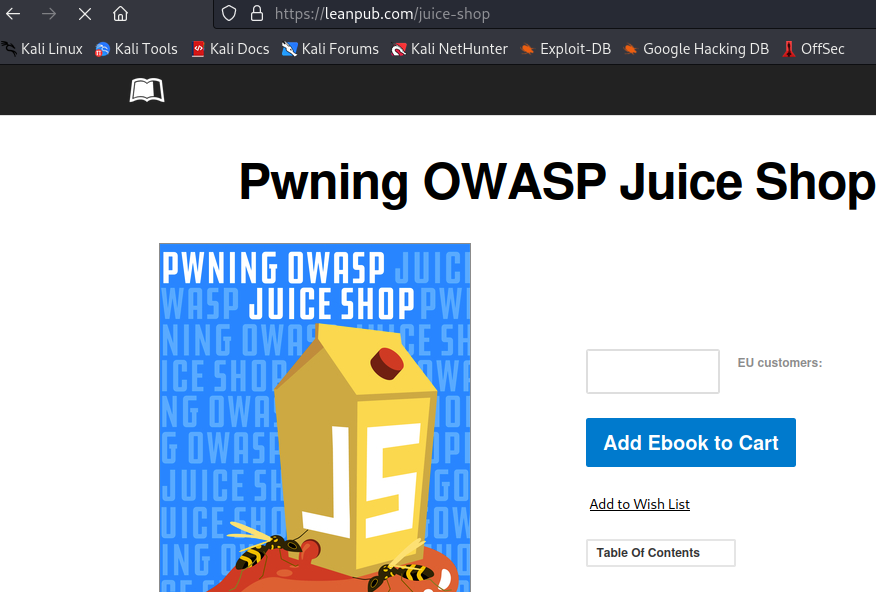


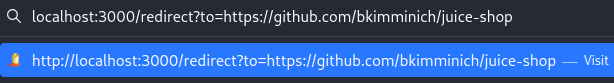


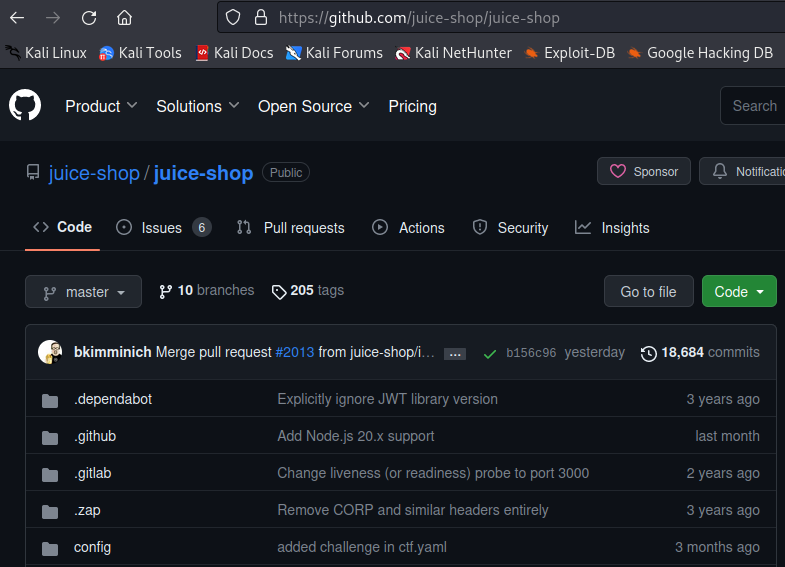




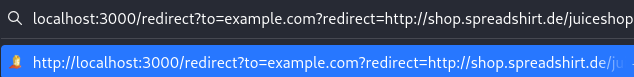




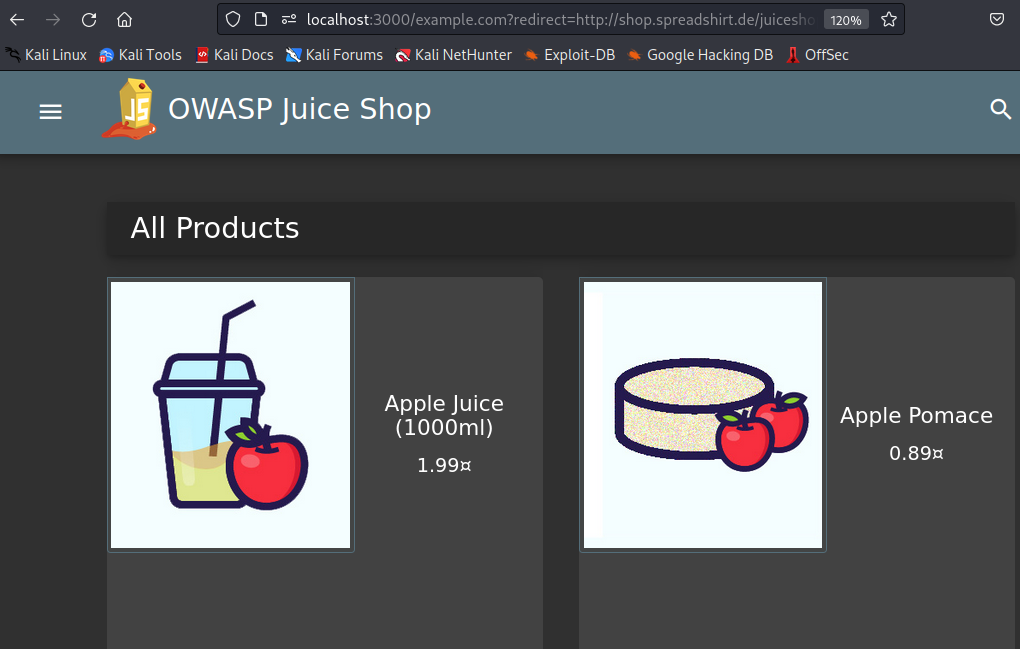




Intuicja podpowiadała, że nie jest to odpowiednia droga, dlatego poszukano kolejnych wskazówek. Okazuje się, że chodzi o podwójne przekierowanie – przez niedozwoloną stronę na dozwoloną:

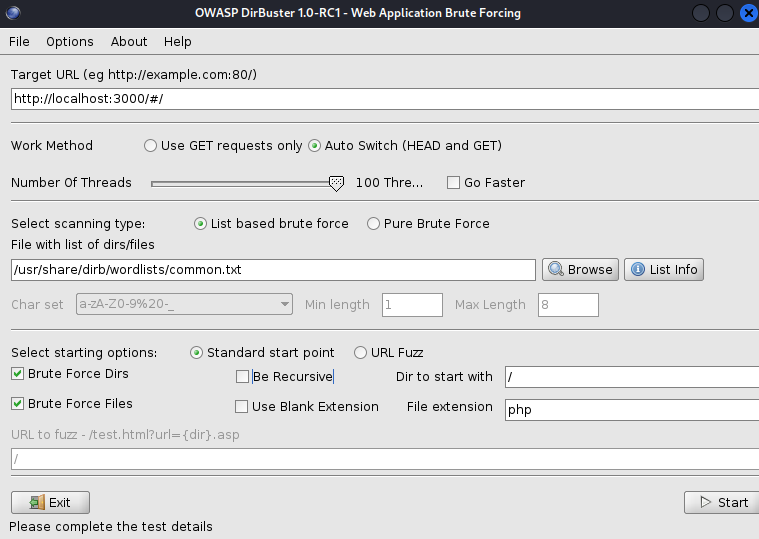


Tym razem udało się ukończyć zadanie:

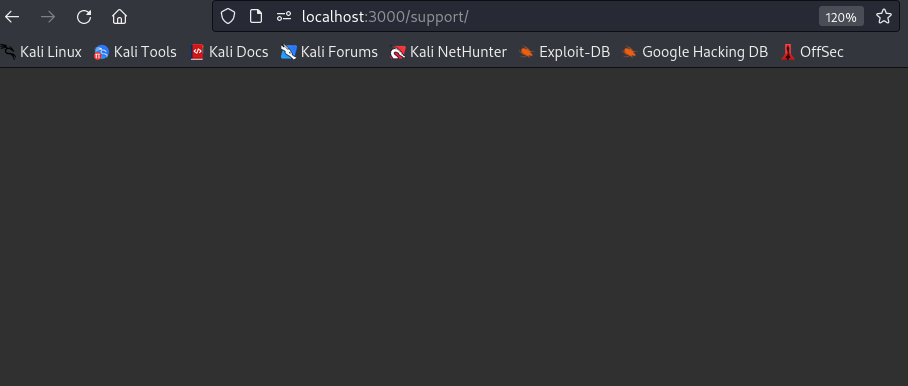


## Access Log

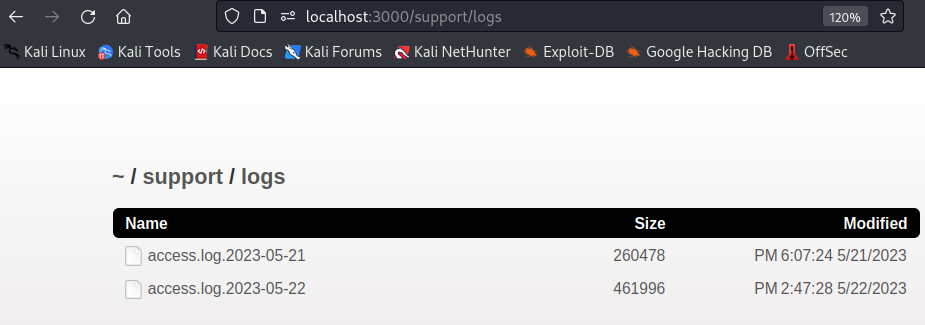
Wyzwanie polega na znalezieniu ścieżki na serwerze, która zawiera logi. W tym celu użyto narzędzia DirBuster, używając gotowej listy z popularnymi słowami, która już znajduje się w Kali Linux:



Niestety DirBuster wszystkie zapytania oznaczał jako 200, dlatego ta droga jest nieprawidłowa. Spróbowano także dokonać enumeracji plików/katalogów za pomocą Intrudera (Burp Suite), lecz skutek był taki sam. Z tego powodu postanowiono poszukać wskazówek. Znaleziono podpowiedź, że warto poszukać w ścieżce /support. Jednak to nie dało żadnego wyniku:



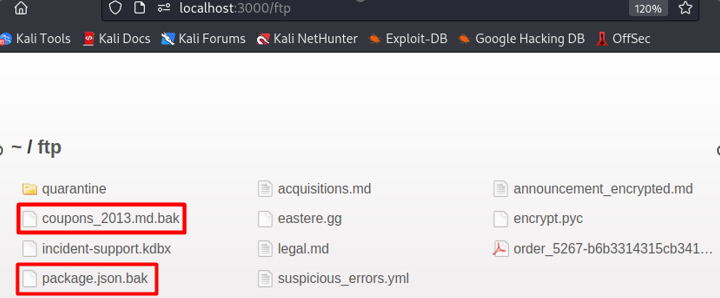
Spróbowano do ścieżki dopisać /logs:



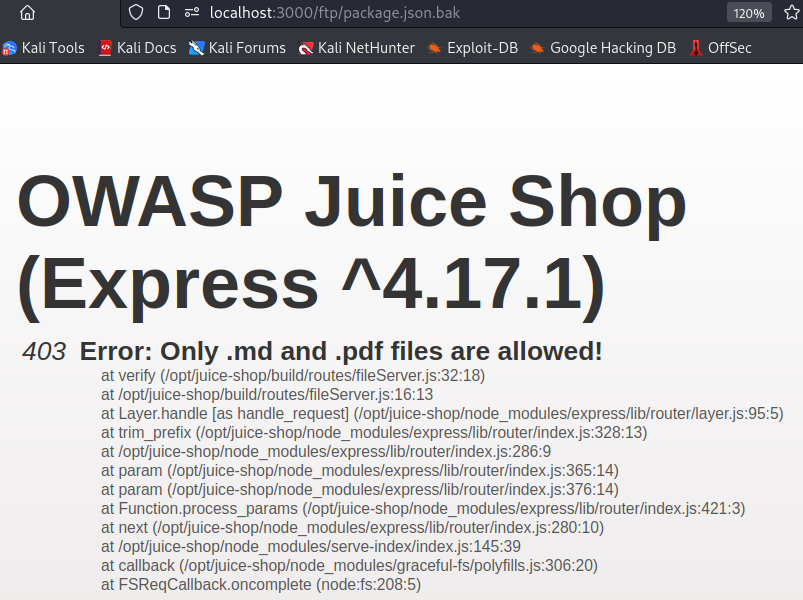
Udało się znaleźć 2 pliki access log, jednak zadanie nie zostało odznaczone jako ukończone w Score Board, dlatego postanowiono pobrać jeden z plików. Tym samym udało się ukończyć wyzwanie.

## Forgotten Developer Backup & Poison Null Byte

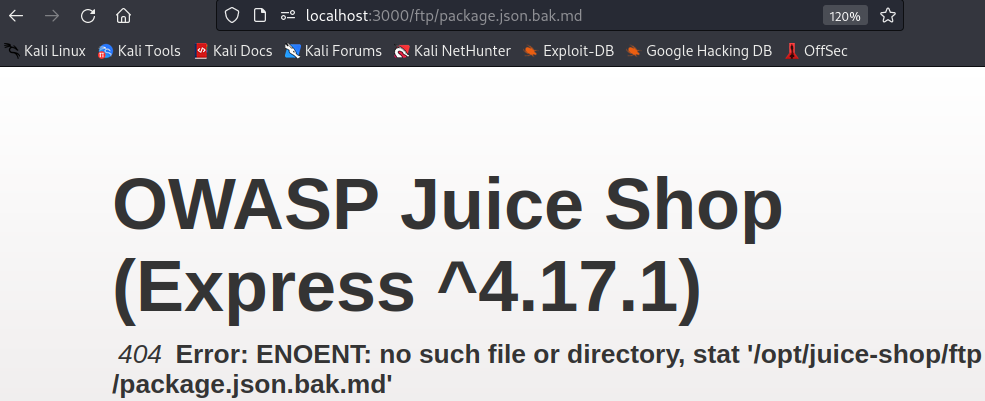
W tym zadaniu chodzi o znalezienie pliku „backup”. W pierwszej kolejności przeszukano miejsce, gdzie wcześniej udało się znaleźć wiele plików, czyli FTP:



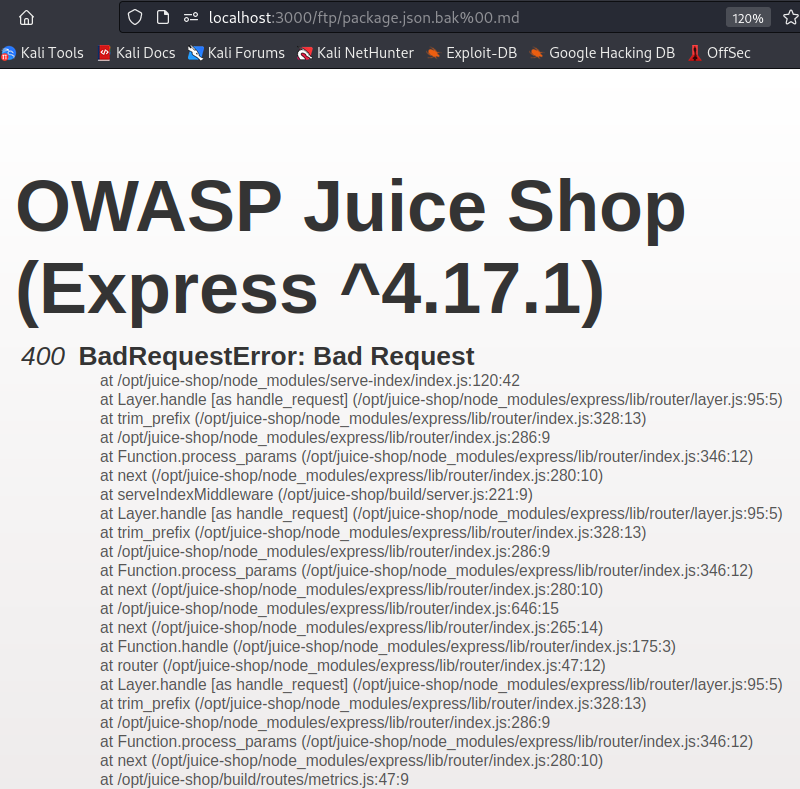
Od razu można zauważyć pliki z rozszerzeniem .bak, który wskazuje na backup. Spróbowano pobrać jeden z nich:



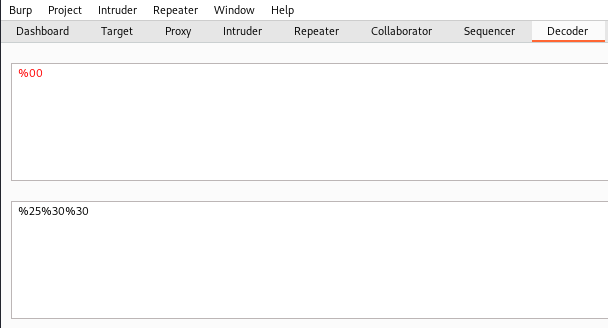
Jak widać, wyłącznie pliku o rozszerzeniami .md oraz .pdf są dozwolone. Postanowiono dopisać rozszerzenie, jednak nie przyniosło do oczekiwanego skutku:



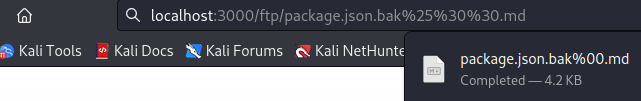
Spróbowano pobrać inne z tych plików, aby poszukać jakichś wskazówek. Niestety nie udało się znaleźć nic interesującego. Poszukano podpowiedzi – okazuje się, że należy zastosować atak Null byte injection. Chodzi o dopisanie %00 przed rozszerzeniem:



Tym razem strona odpowiedziała kodem 400. Prawdopodobnie URL interpretuje %00 jako inną wartość. Postanowiono zaenkodować w URL tę część:



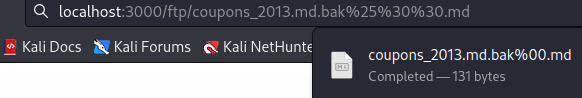
Wpisano to w adres URL:



Udało się rozwiązać przez przypadek 2 wyzwania jednocześnie.

## Forgotten Sales Backup

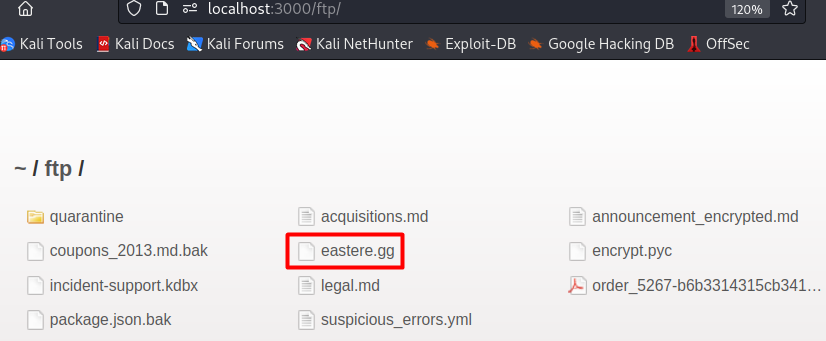
W wyzwaniu chodzi o pobranie drugiego pliku, tym razem prawdopodobnie coupons\_2013.md.bak. Spróbowano pobrać plik tym samym sposobem:



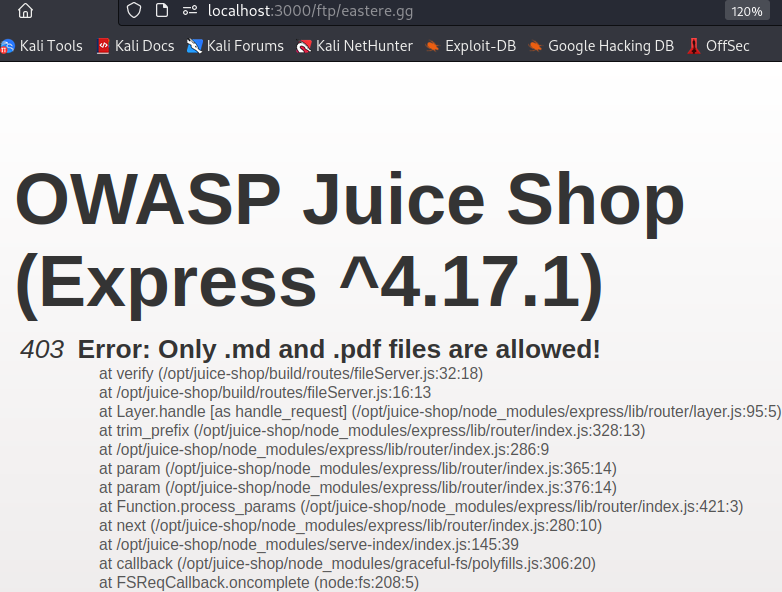
Udało się pobrać w ten sam sposób i ukończyć wyzwanie.

## Easter Egg

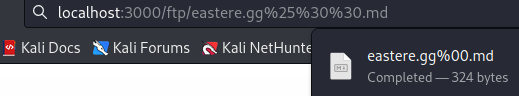
Wyzwanie jest związane z pobraniem Easter Egga. Ponownie w katalogu /ftp można zauważyć docelowy plik – eastere.gg:



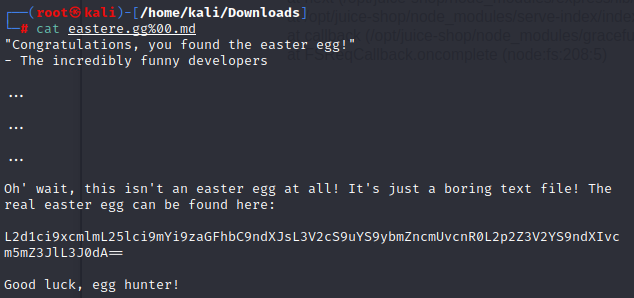
Spróbowano go pobrać:



Kolejny raz postanowiono zastosować ten sam sposób do pobrania pliku:



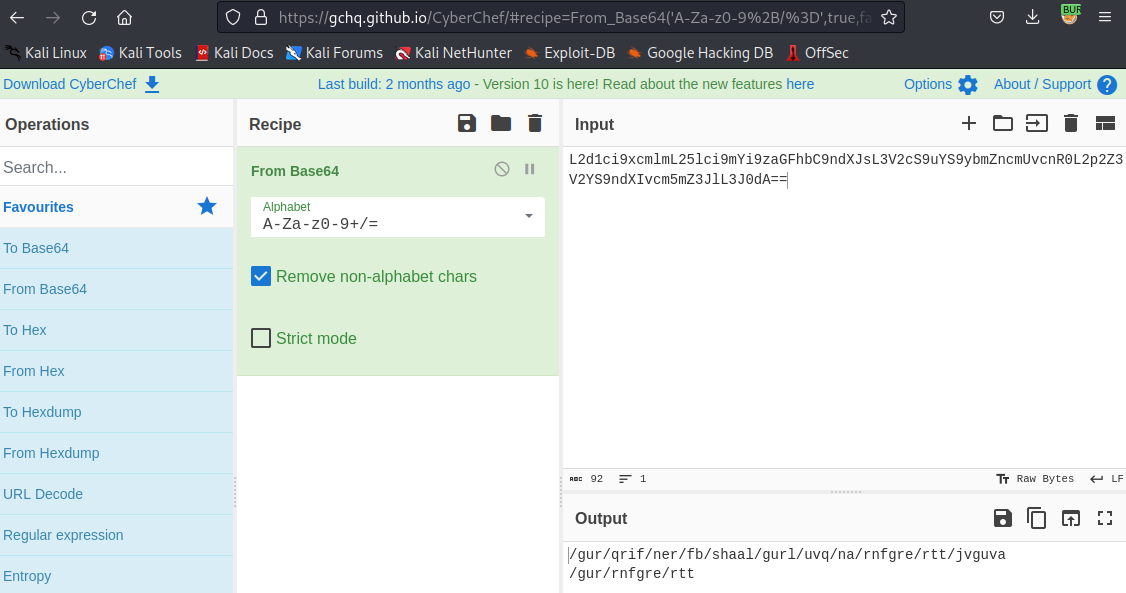
Po pobraniu sprawdzono narzędziem cat, co znajduje się wewnątrz pliku:



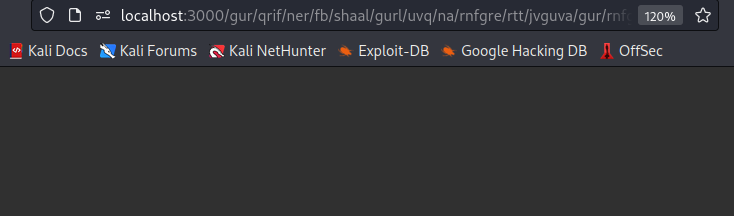
Na tym etapie wyzwanie zostało już ukończone, jednak postanowiono sprawdzić, czym jest ciąg znaków zawarty w pliku.

## Nested Easter Egg

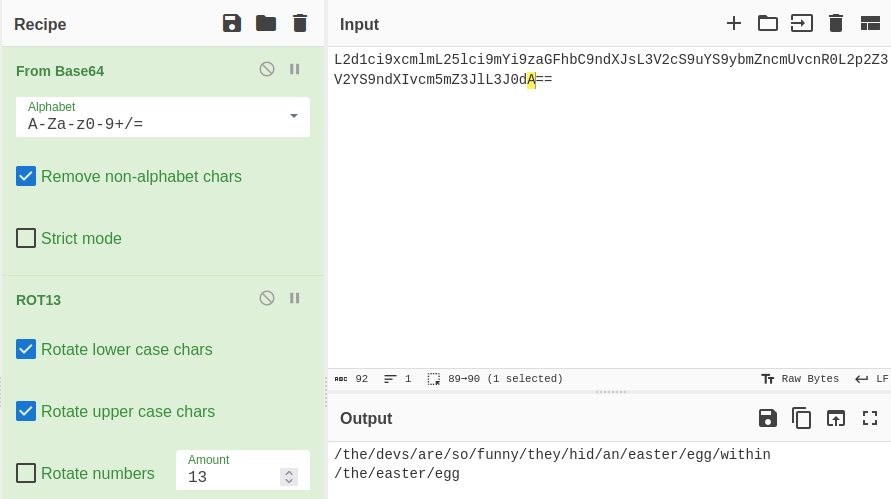
Wyzwanie polega na kryptoanalizie pliku eastere.gg. Na pierwszy rzut oka można zauważyć == na końcu, co może wskazywać na Base64. Wrzucono ten ciąg do CyberChefa:



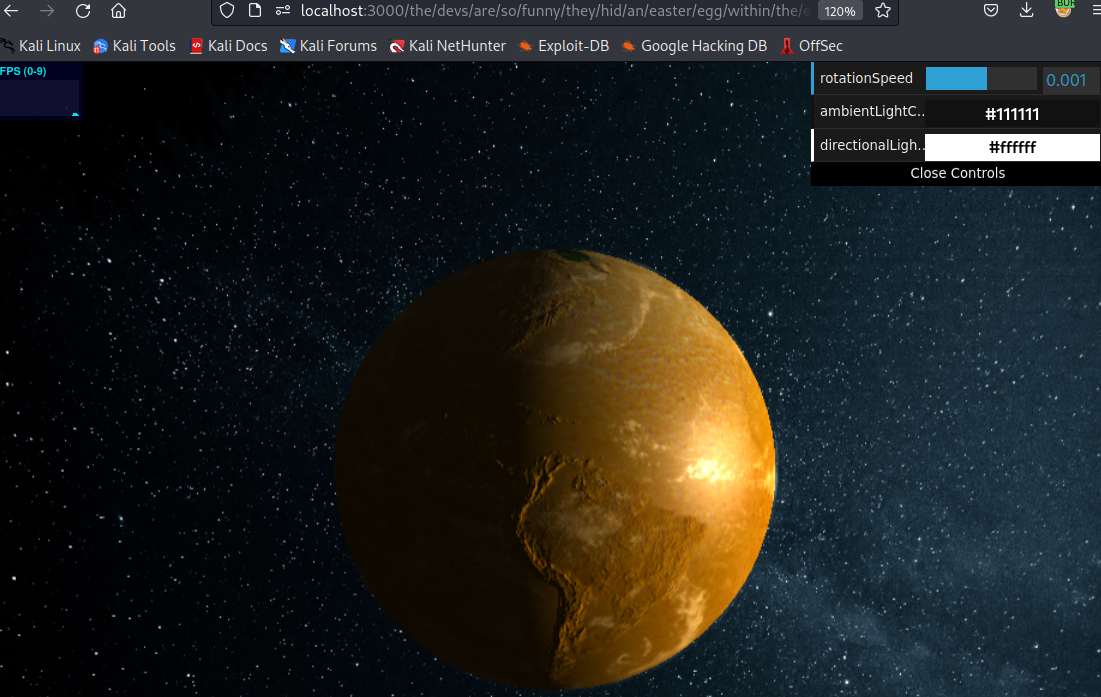
Odnaleziono ścieżkę, na pierwszy rzut oka wygląda na ponownie zaszyfrowaną jakimś szyfrem przestawnym, np. ROT13. Mimo wszystko sprawdzono najpierw, czy ta ścieżka do czegoś prowadzi:



Następnie spróbowano wykorzystać szyfr ROT13:



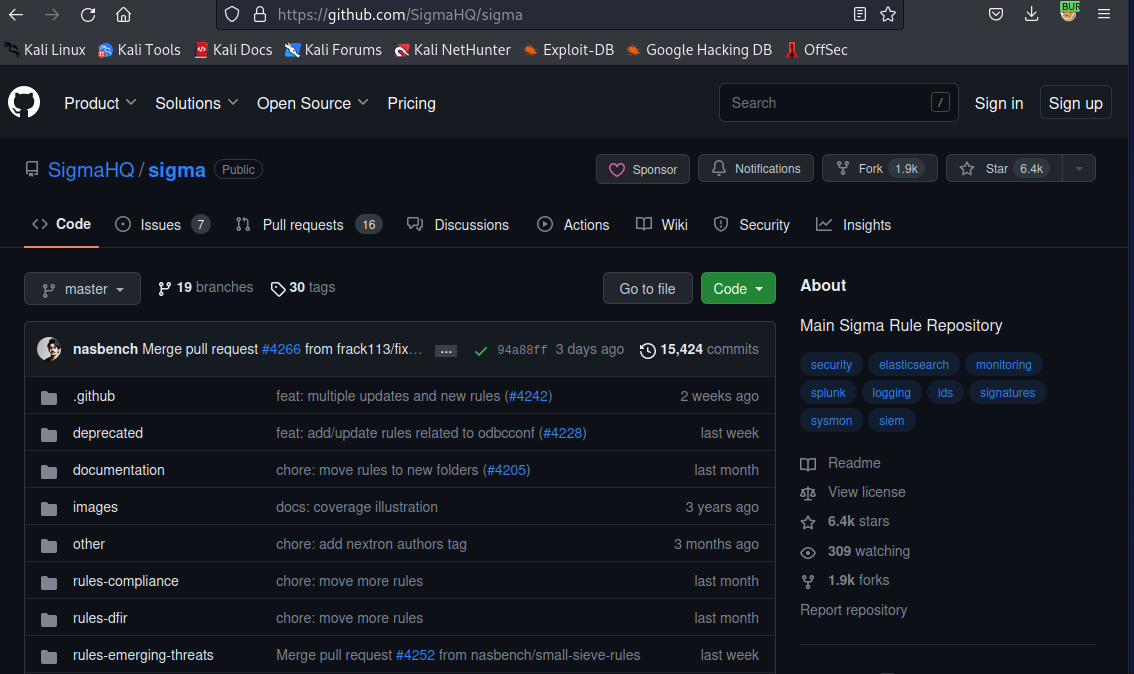
Udało się znaleźć ścieżkę, która niestety nie wygląda na ścieżkę. Mimo tego sprawdzono, czy prowadzi do czegoś:



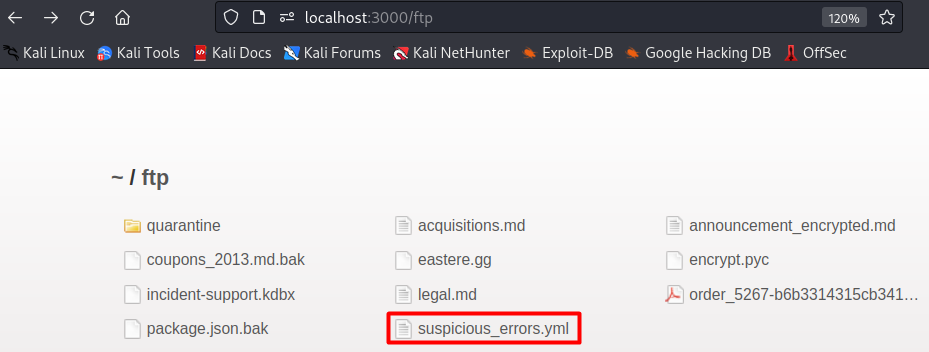
W ten sposób udało się ukończyć wyzwanie.

## Misplaced Signature File

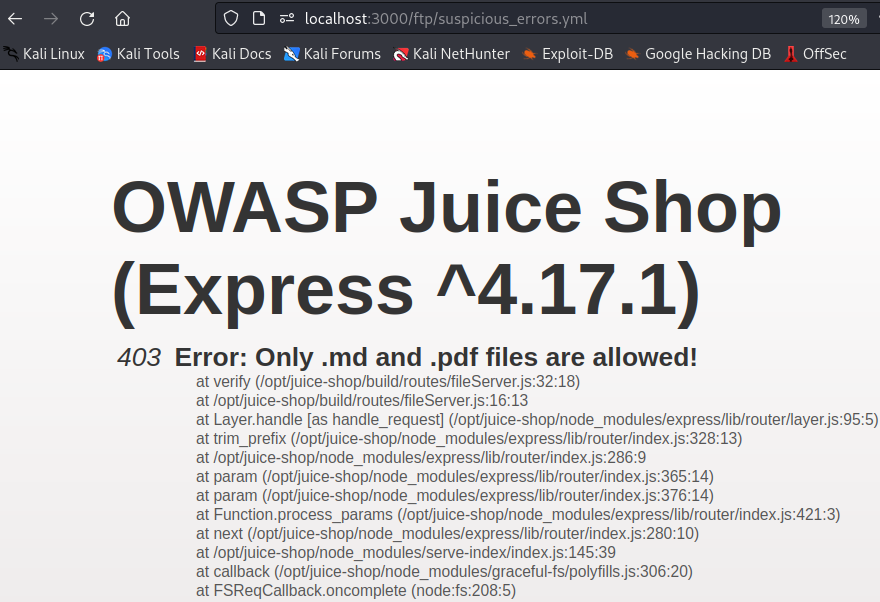
Wyzwanie polega na znalezieniu pliku z podpisem SIEM. Link przekierowuje na githuba:



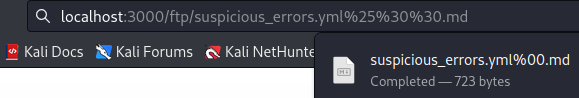
Przykładowe pliki znajdujące się tam, są o rozszerzeniu .yml. Ponownie sprawdzano katalog ftp, gdzie jeden plik jest właśnie w tym formacie: suspicious\_errors.yml.



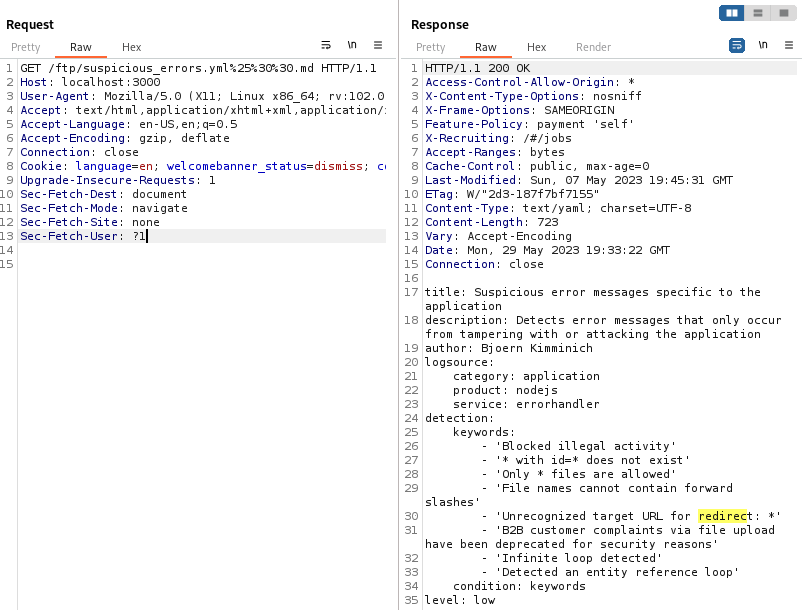
Jak w poprzednich przypadkach – pliku nie da się pobrać:



Spróbowano ponownie użyć Poison Null Byte:

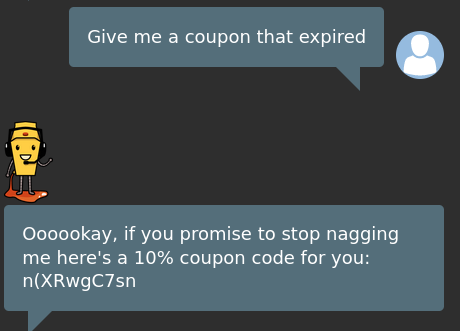


Plik udało się pobrać:

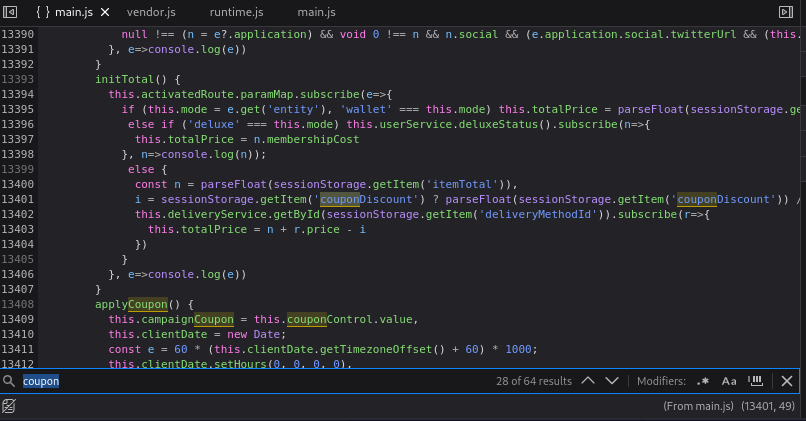


## Expired Coupon

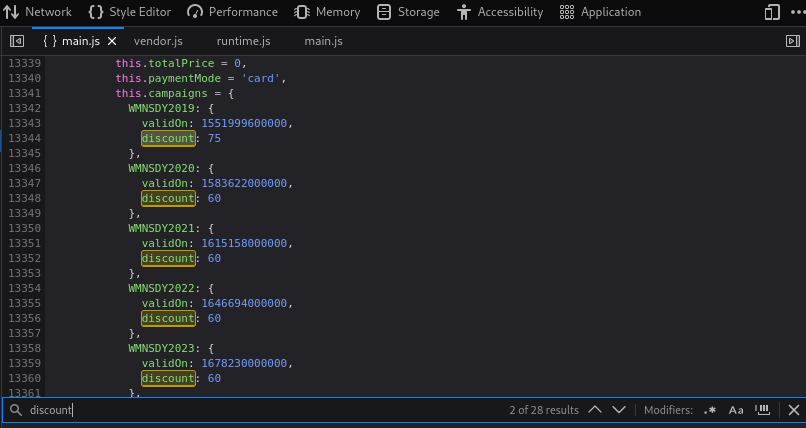
Zadanie polega na zastosowanie kuponu, który już wygasł. Z uwagi na fakt, że w 1-gwiazdkowym challenge’u pojawił się bot, który przekazał kupon, w pierwszej kolejności spytano go o przestarzałe kupony:



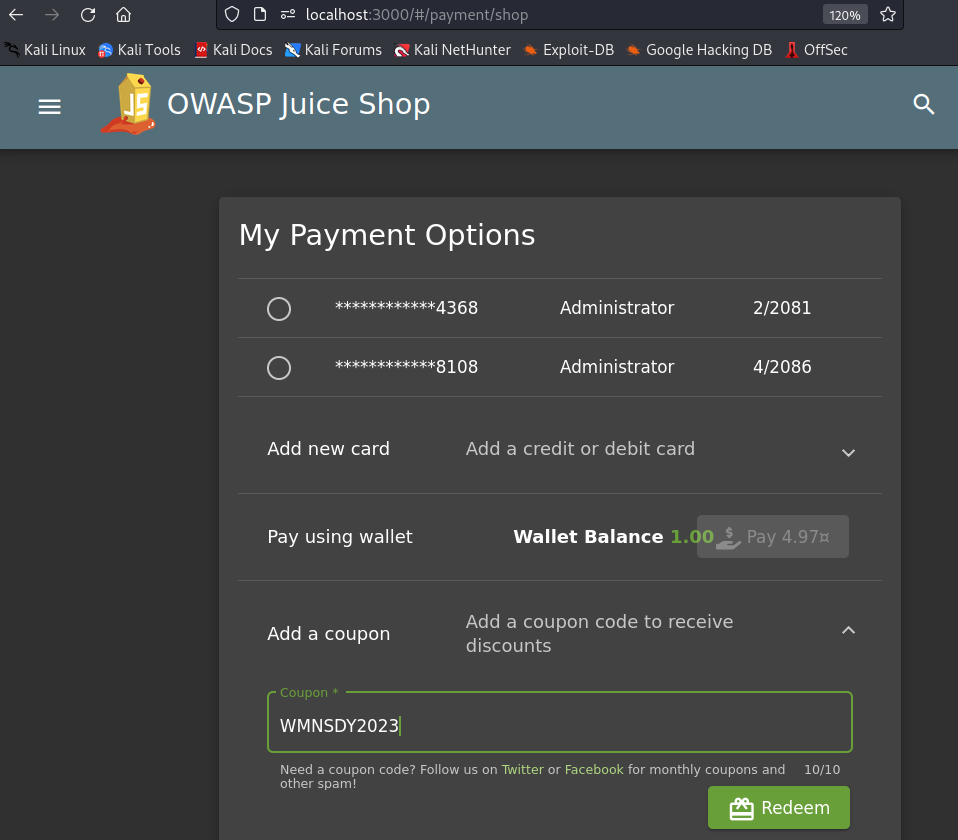
Niestety bot podał tylko aktualny kod rabatowy. W związku z tym postanowiono ponownie przeszukać kod strony:



Spróbowano także odszukać słowo „discount”:



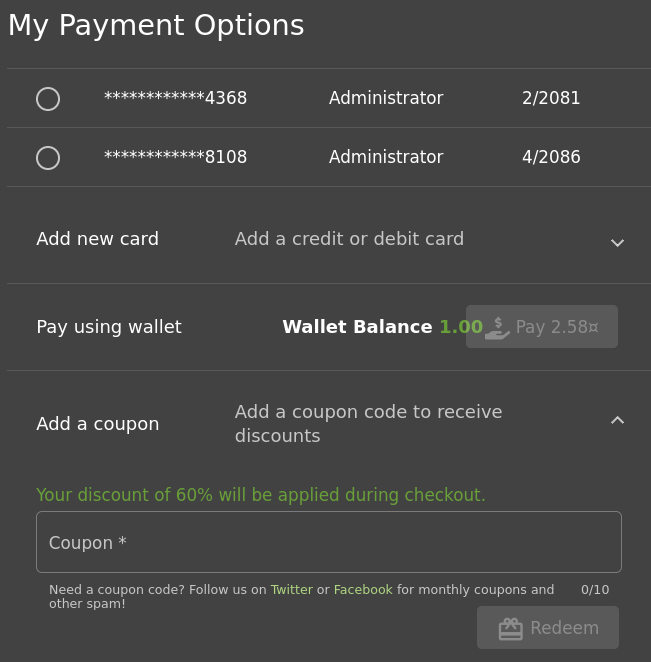
Znaleziono kilka kodów rabatowych. Spróbowano użyć tego ostatniego:



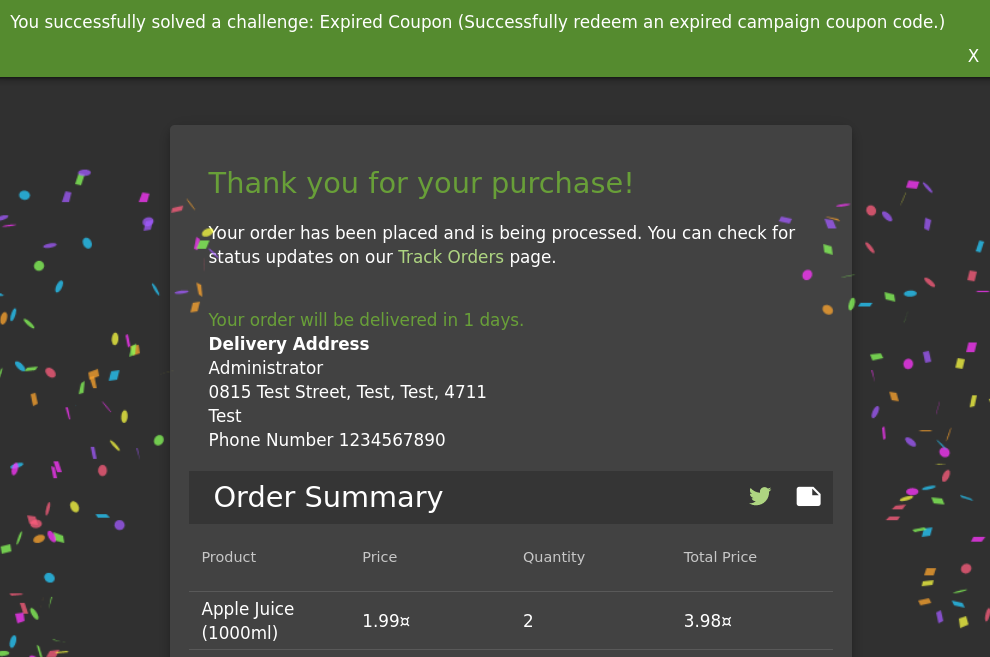
Wyświetliła się informacja, że kupon jest nieważny. Kupon wskazuje na dzień kobiet (women’s day), dlatego postanowiono zmienić czas wirtualnej maszynie za pomocą timedatectl:



W ten sposób udało się zastosować kod:



Następnie dokonano zakupu:

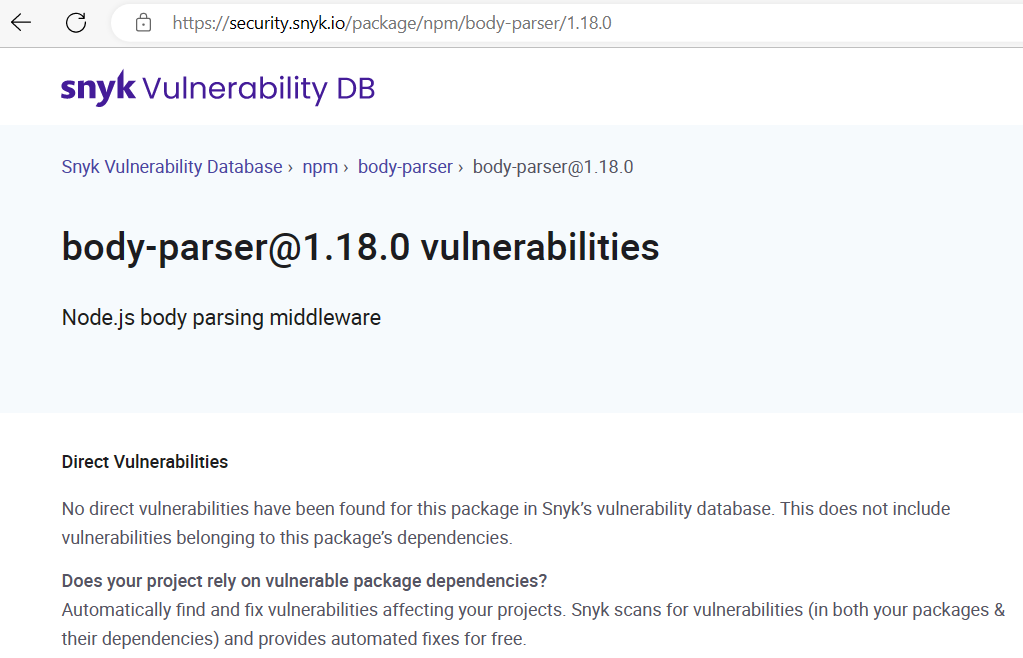


## Vulnerable Library

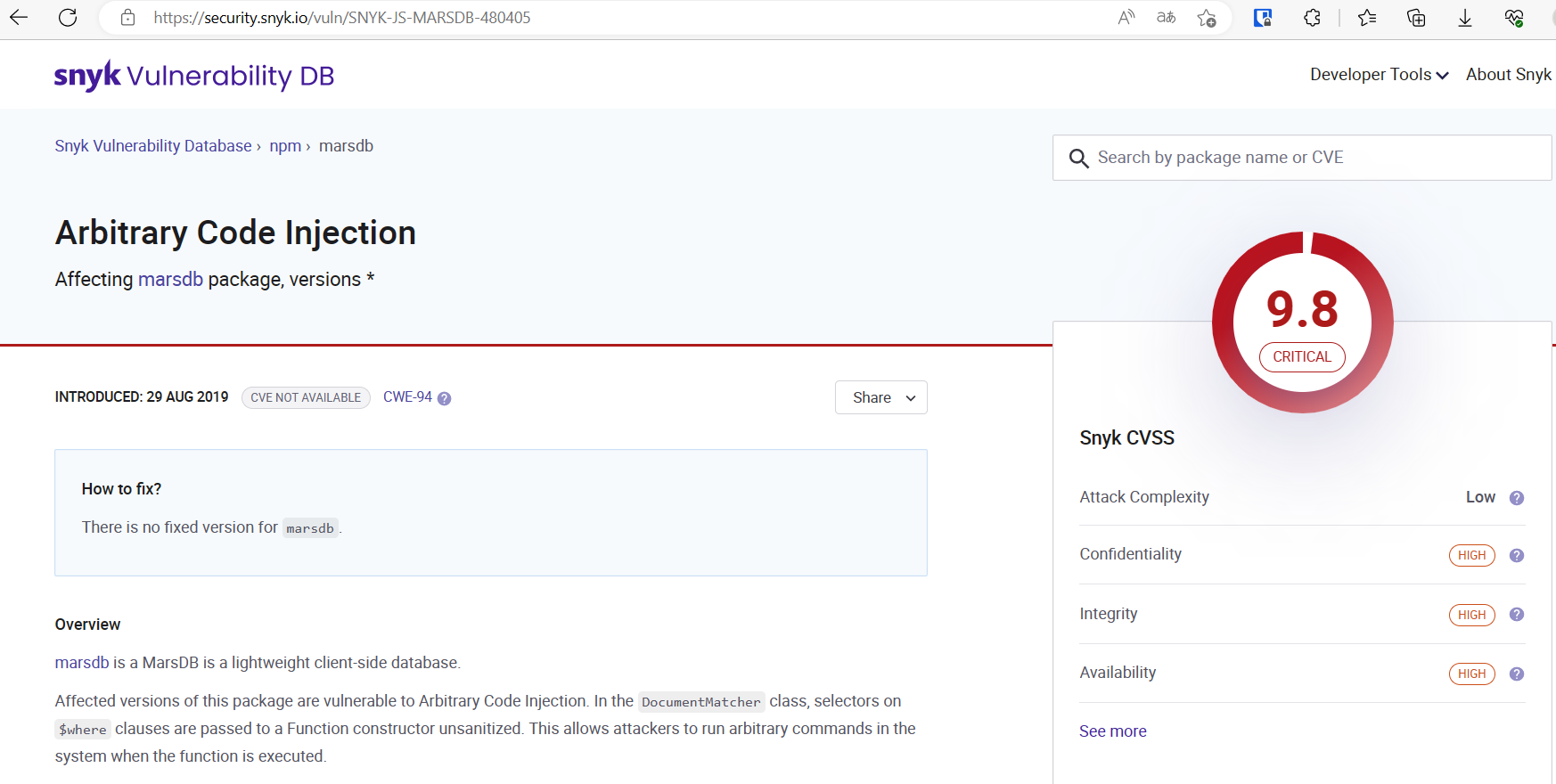
Zadanie jest podobne do wcześniejszego wyzwania, w którym należało wysłać feedback do sklepu, wówczas odnośnie słabego algorytmu kryptograficznego. Tym razem należy poinformować o podatnej bibliotece. Wszystkie biblioteki udało się znaleźć w pliku package.json.bak:



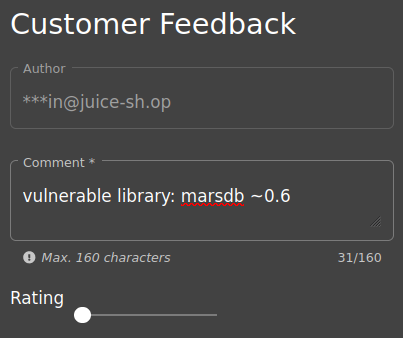
Po kolei sprawdzono w Internecie, które biblioteki są podatne w następujący sposób:



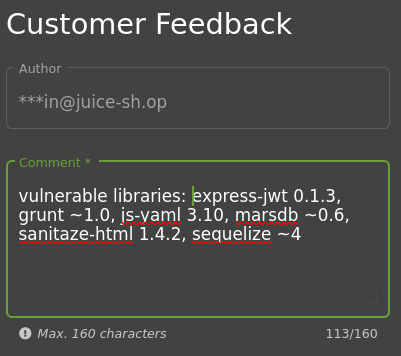
Znaleziono podatności w bibliotece marsdb 0.6:



Postanowiono sprawdzić, czy o tę bibliotekę chodzi:



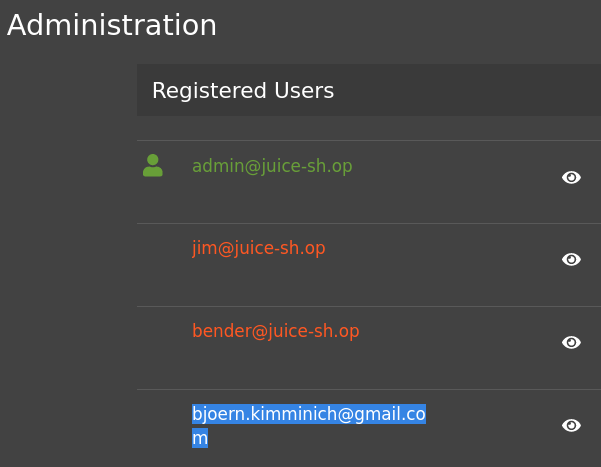
Niestety to nie wystarczyło, poszukano kolejnych bibliotek. Znaleziono jeszcze kilka podatnych i poinformowano sklep o nich:



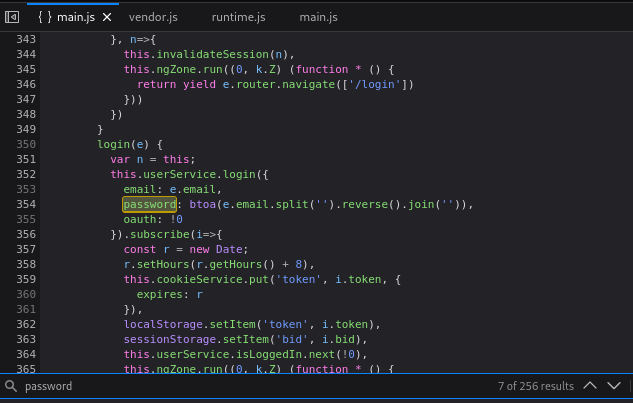
Dopiero w ten sposób udało się rozwiązać zadanie.

## Login Bjoern

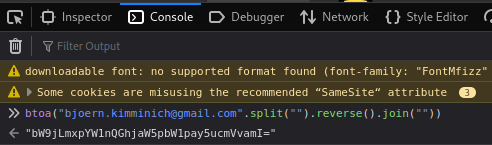
Wyzwanie polega na zalogowaniu się jako Bjoern (na konto gmail) bez zmieniania hasła, SQL injection, czy też hackowania konta. Jako pierwszy krok wyświetlono listę użytkowników na koncie admina:



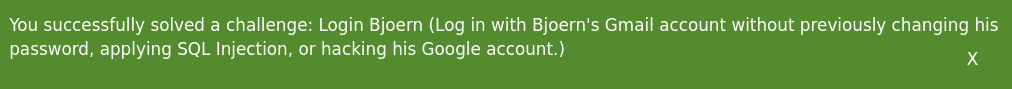
Adres Bjoerna: bjoern.kimminich@gmail.com. Z braku pomysłów postanowiono przeszukać ponownie kod strony:



Znaleziono fragment, gdzie hasło zostaje pozyskane ze zmodyfikowanego adresu e-mail - btoa(e.email.split('').reverse().join('')). Zgodnie z dokumentacją, funkcja btoa() odpowiada za przekształcenie na base64:

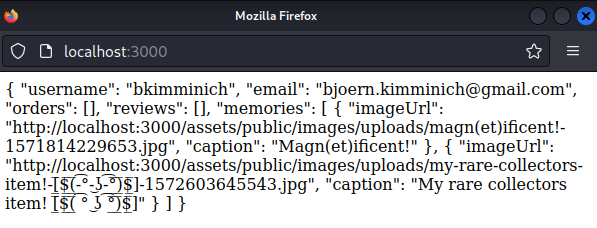


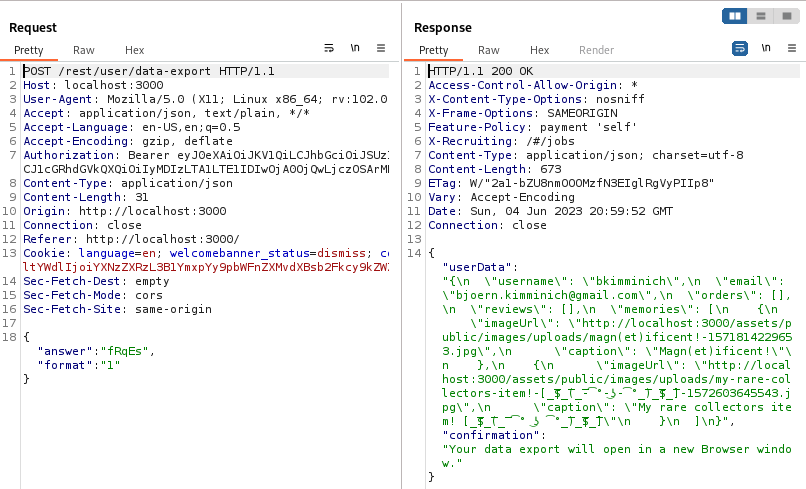
Spróbowano wpisać ciąg znaków (bW9jLmxpYW1nQGhjaW5pbW1pay5ucmVvamI=) jako hasło:



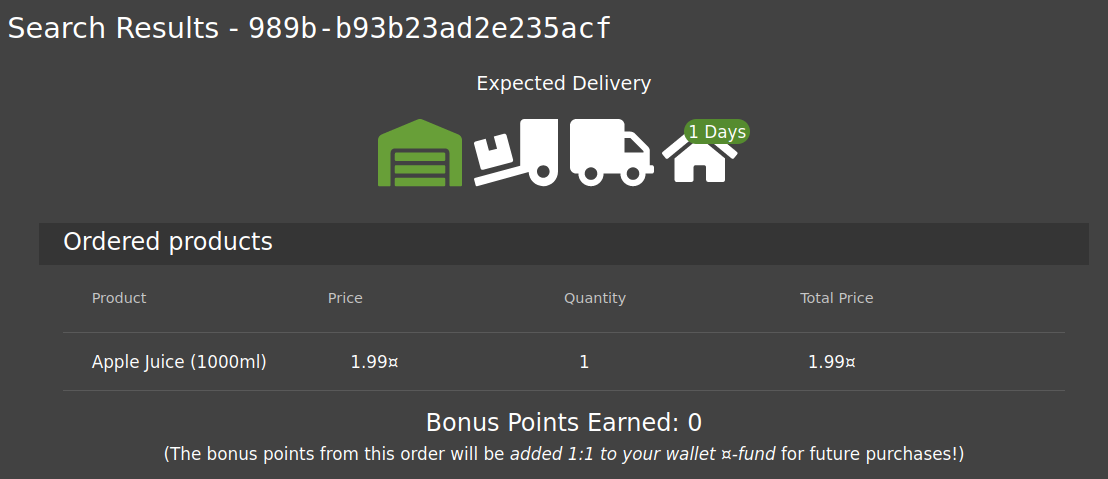
## GDPR Data Theft

Wyzwanie polega na zdobyciu danych osobowych bez wstrzyknięć. Spróbowano zdobyć dane Bjoerna, eksportując plik JSON:





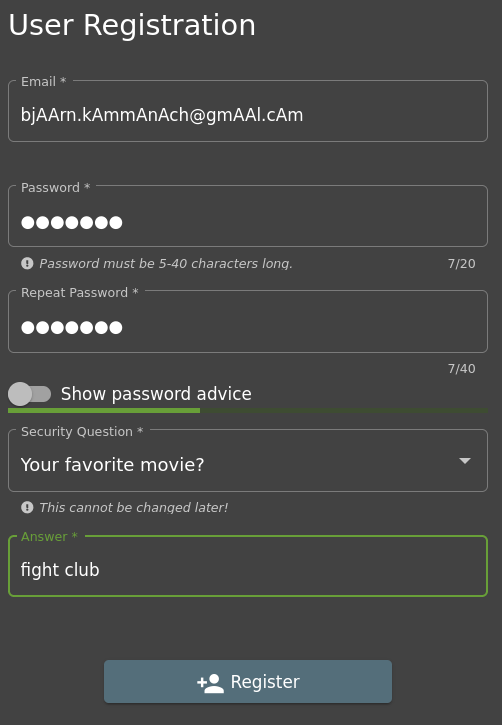
Zgodnie ze wskazówkami, warto przyjrzeć się procesowi zakupu.



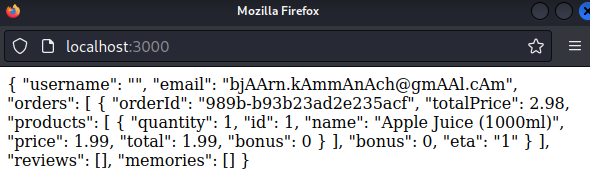
Znaleziono interesującą rzecz, email został wygwiazdkowany:



Wylogowano się z aktualnego konta i utworzono nowe, z innymi literami w miejscach gwiazdek:

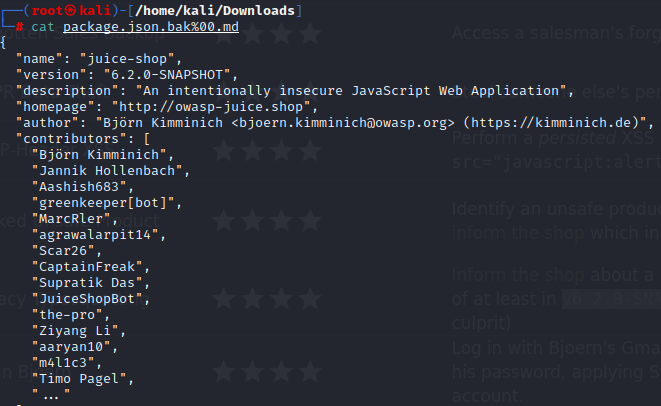


Zalogowano się na utworzone konto i pobrano dane, w ten sposób udało się rozwiązać zadanie:

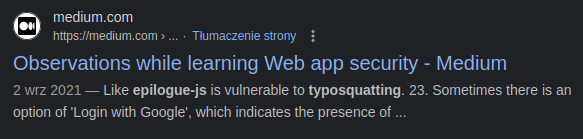


## Legacy Typosquatting

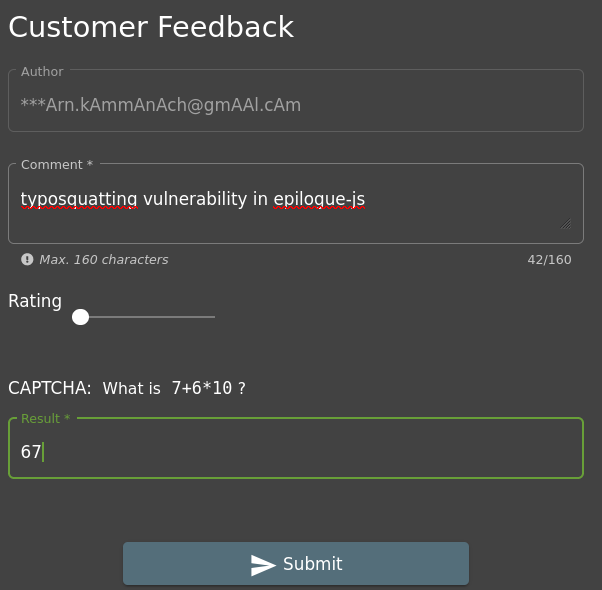
Zadanie polega na poinformowaniu sklepu o typosquattingu powiązanym ze snapshotem v6.2.0-SNAPSHOT. Typosquatting jest to utworzeniem strony o podobnej nazwie, np. ze zmienioną literą. Zajrzano ponownie do pliku backup, ponieważ pojawił się tam właśnie ten snapshot:



Postanowiono wpisać nazwy bibliotek z dopiskiem ‘typosquatting’.



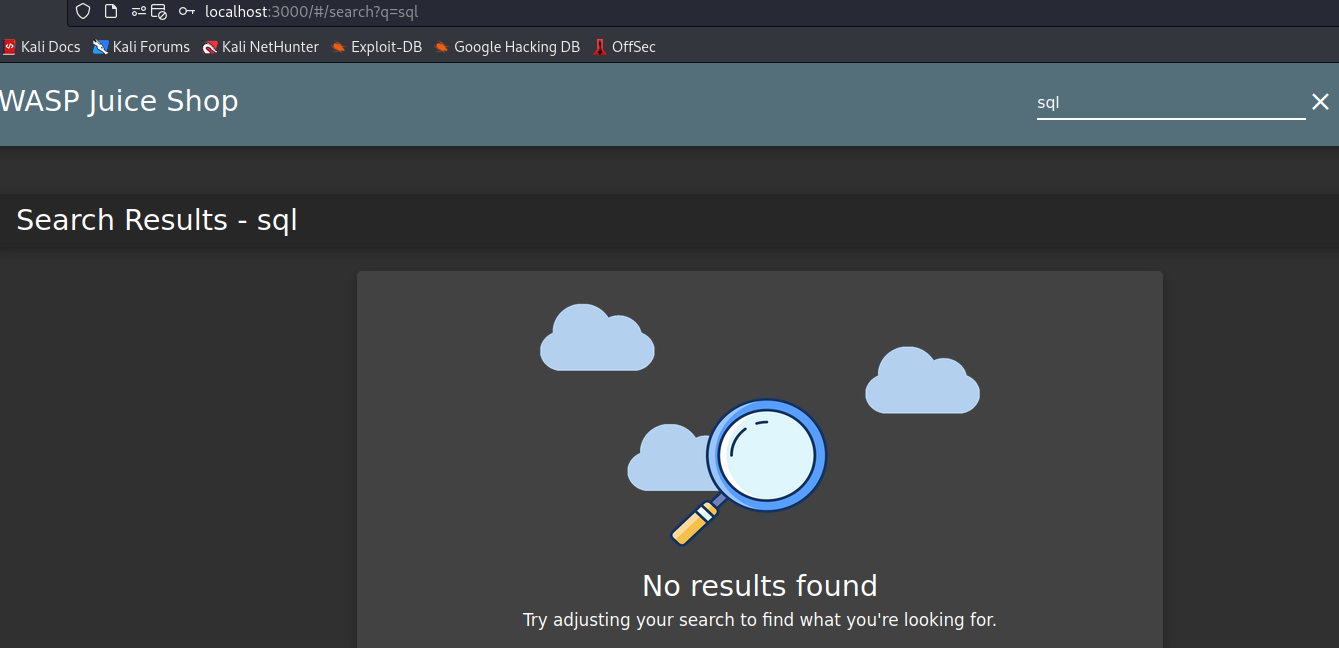
Wpisano epilogue-js jako feedback:



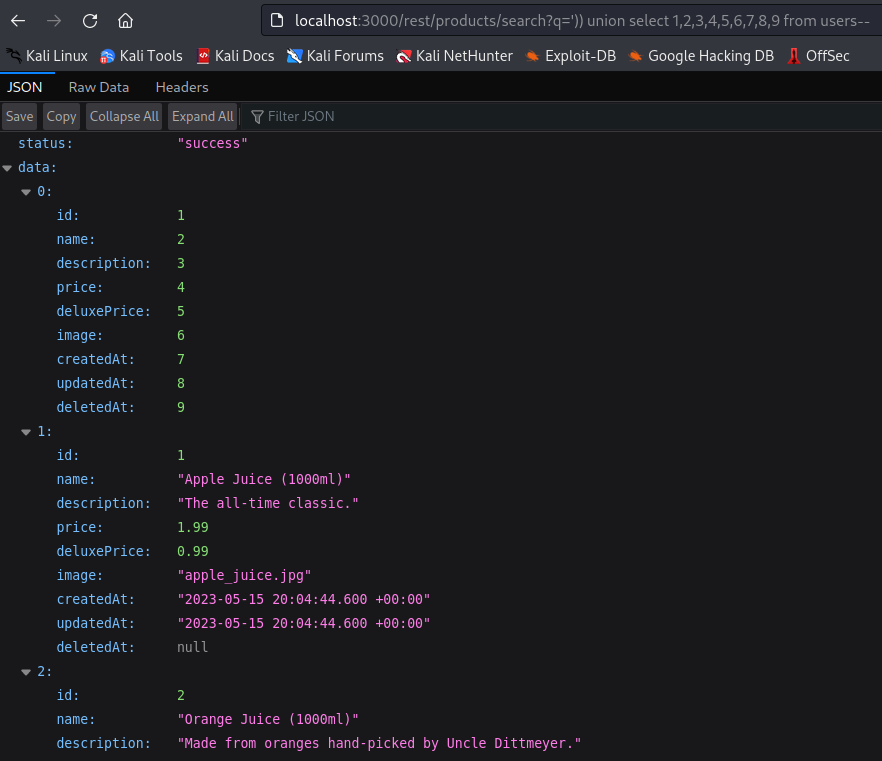
W ten sposób ukończono wyzwanie.

## User credentials

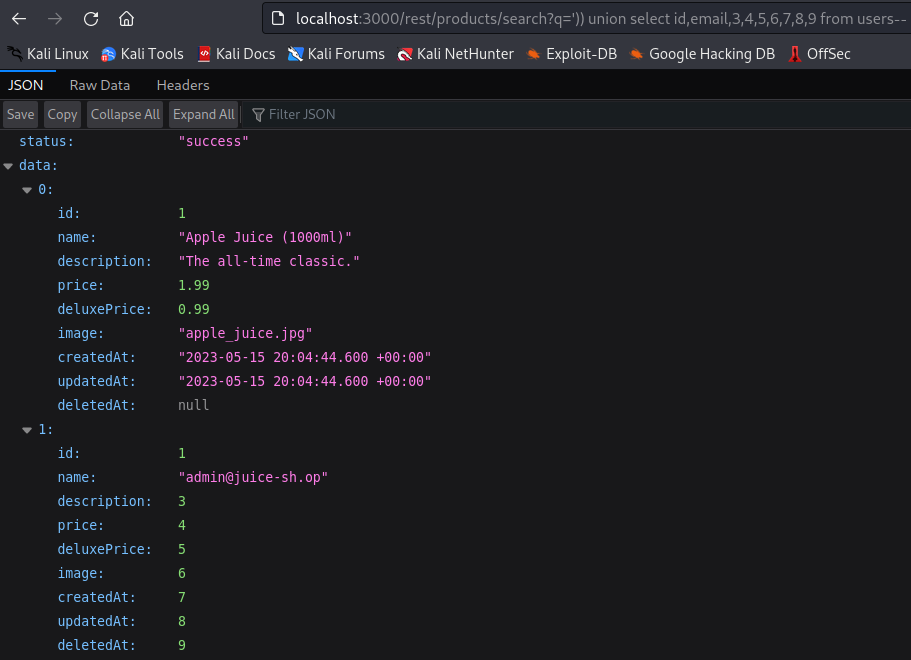
Za pomocą SQL injection, należy zdobyć dane uwierzytelniające wszystkich użytkowników.



Zastosowano atak SQL injection union-based. Przykładowe tabele, które próbowano odgadnąć to users. Udało się znaleźć część danych, jednak nie pozwoliło to na wykonanie wyzwania:



Jednak dzięki temu znaleziono nazwy danych. Próbowano zdobyć kolejne dane:



W końcu udało się wyświetlić hasło (pod nazwą description):

