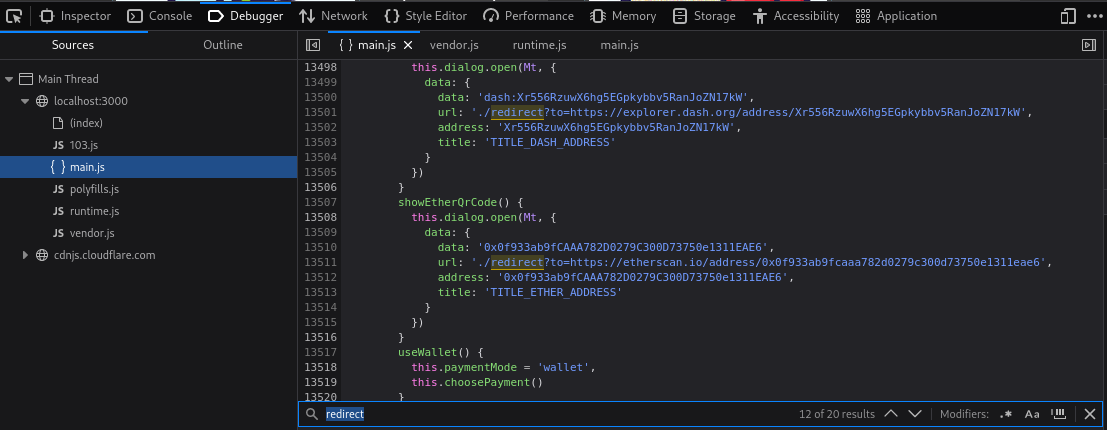
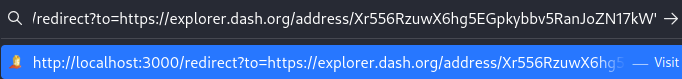
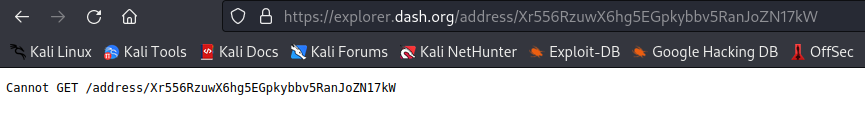
## Allowlist Bypass

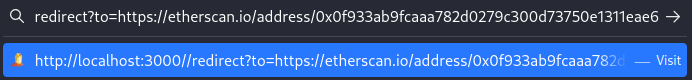
Zadanie jest podobne do wcześniejszego jednogwiazdkowego wyzwania Unvalidated Redirects. Dlatego do jego wykonania przystąpiono w podobny sposób – wyszukując redirect w Debuggerze w DevTools:

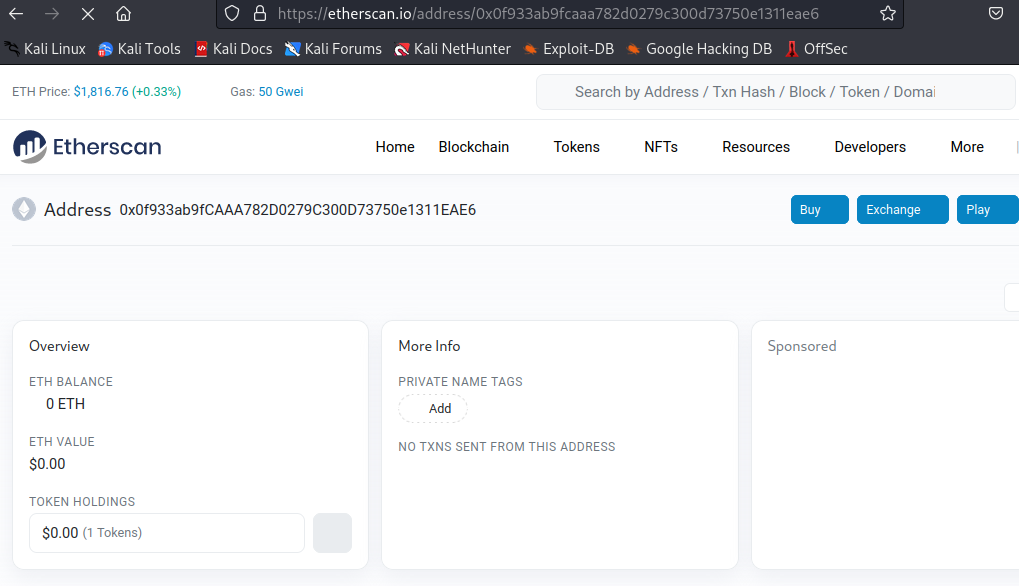


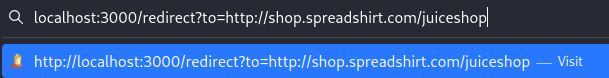
Jak widać, przekierowania można dokonać za pomocą ścieżki /redirect?to=<adres>. Rozpoczęto testowanie wszystkich linków po kolei:

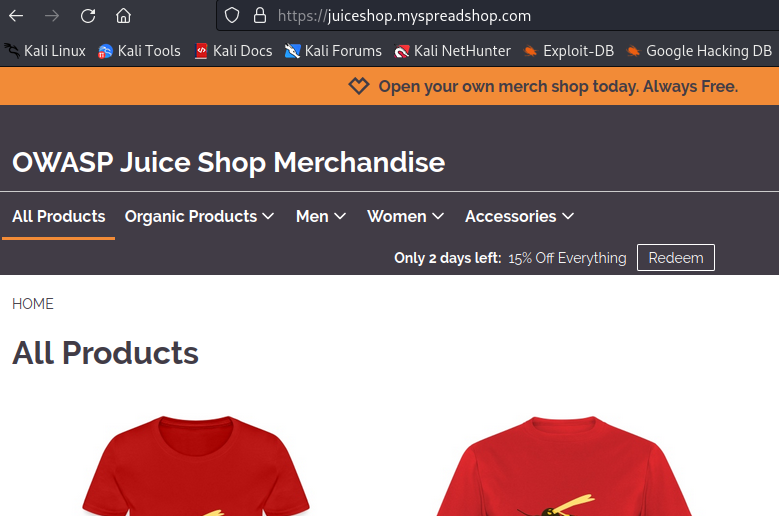


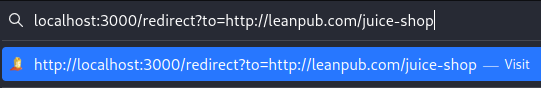


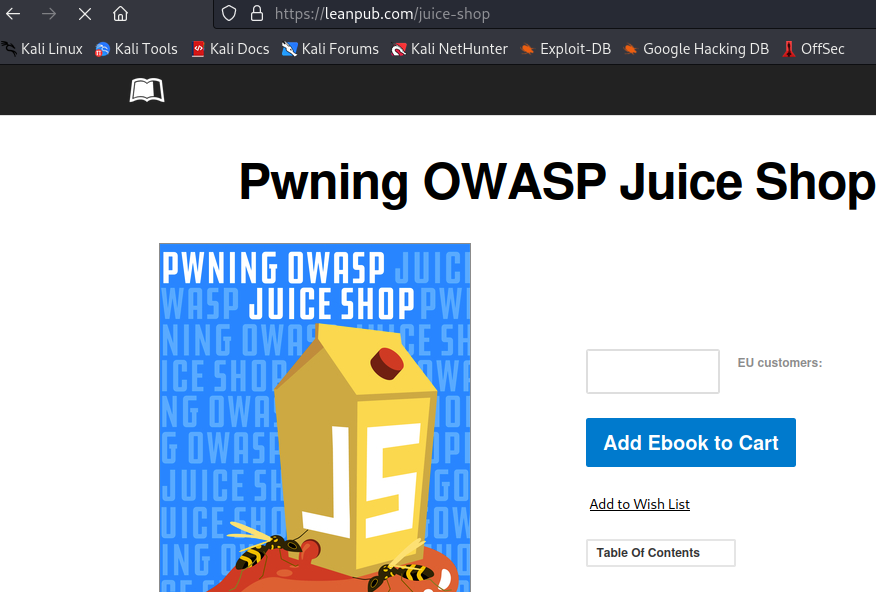


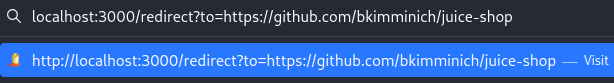


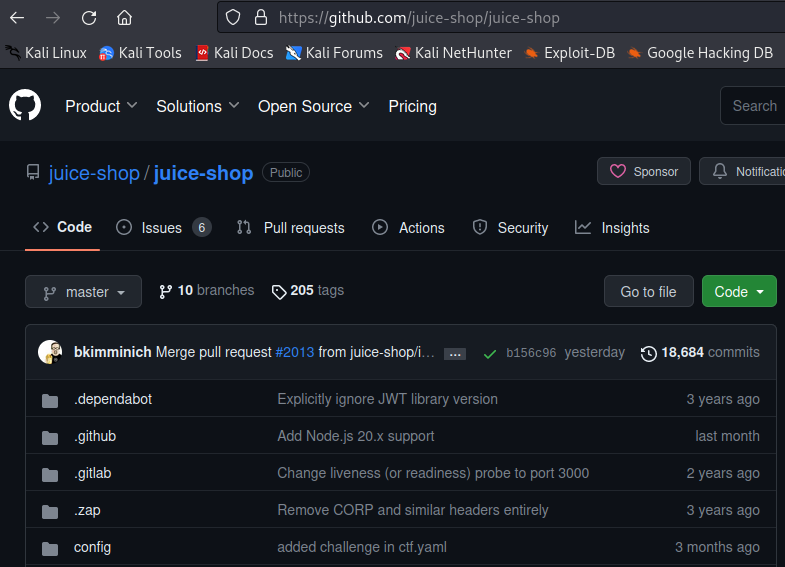




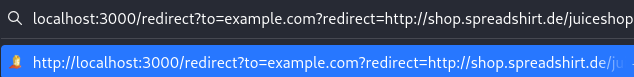




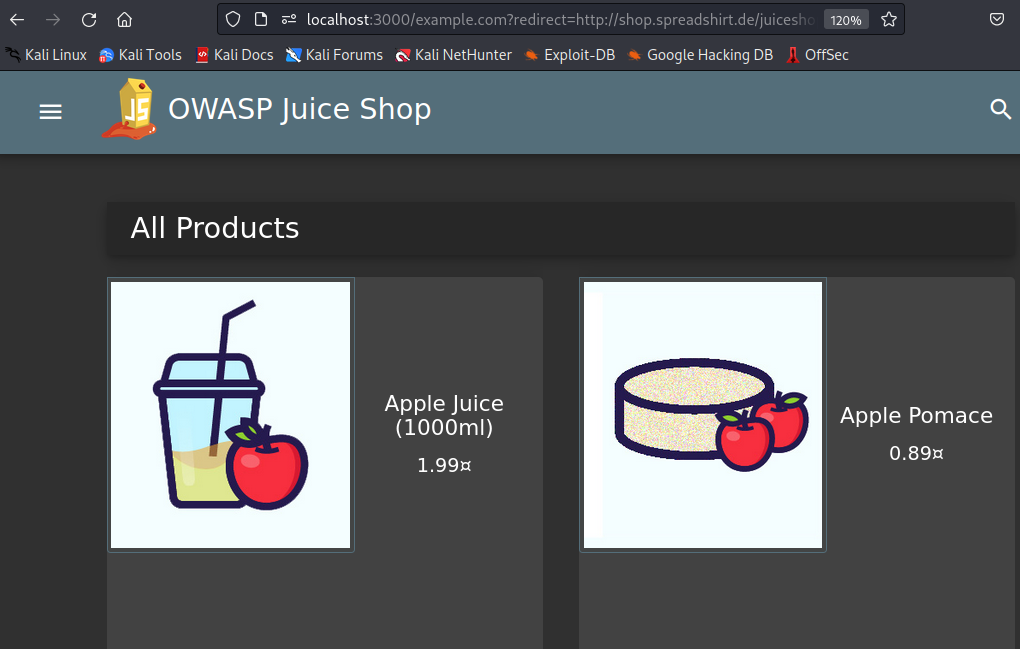




Intuicja podpowiadała, że nie jest to odpowiednia droga, dlatego poszukano kolejnych wskazówek. Okazuje się, że chodzi o podwójne przekierowanie – przez niedozwoloną stronę na dozwoloną:

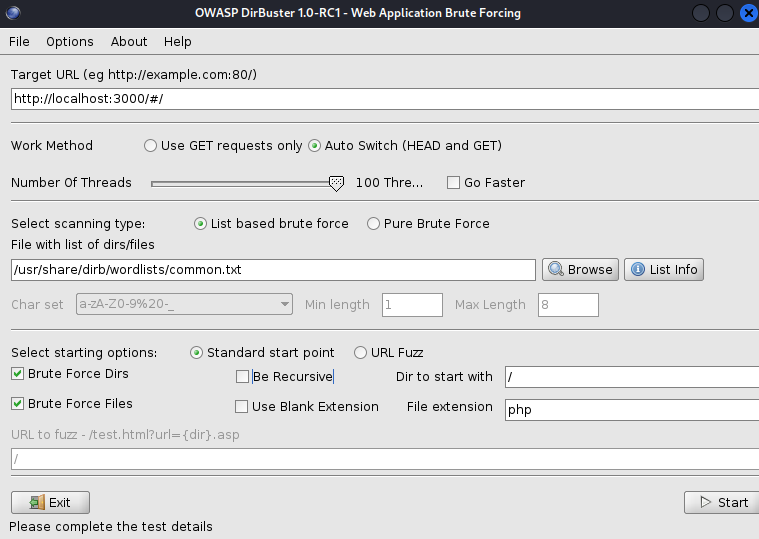


Tym razem udało się ukończyć zadanie:

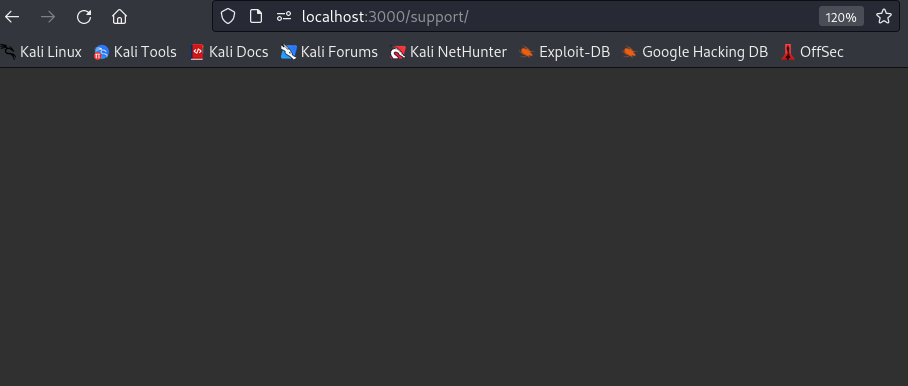


## Access Log

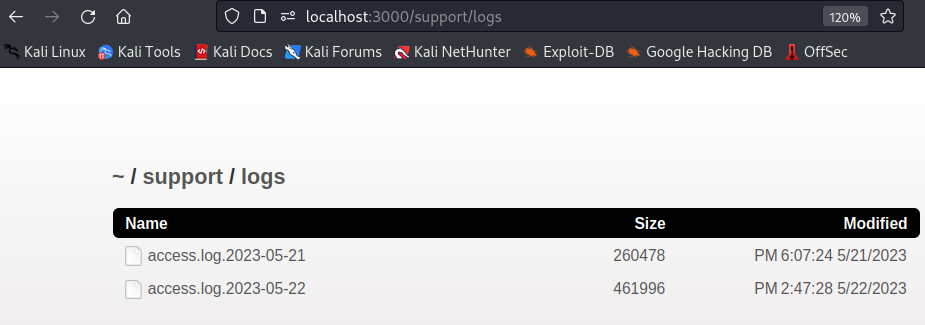
Wyzwanie polega na znalezieniu ścieżki na serwerze, która zawiera logi. W tym celu użyto narzędzia DirBuster, używając gotowej listy z popularnymi słowami, która już znajduje się w Kali Linux:



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:



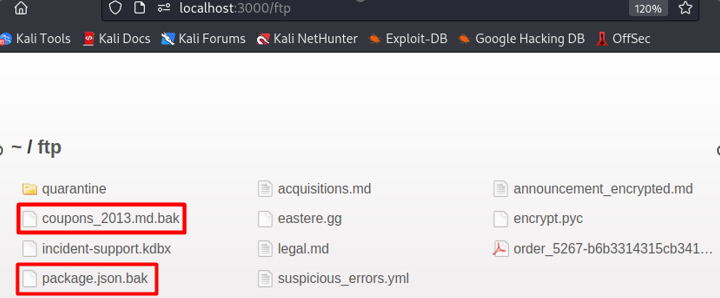
Spróbowano do ścieżki dopisać /logs:



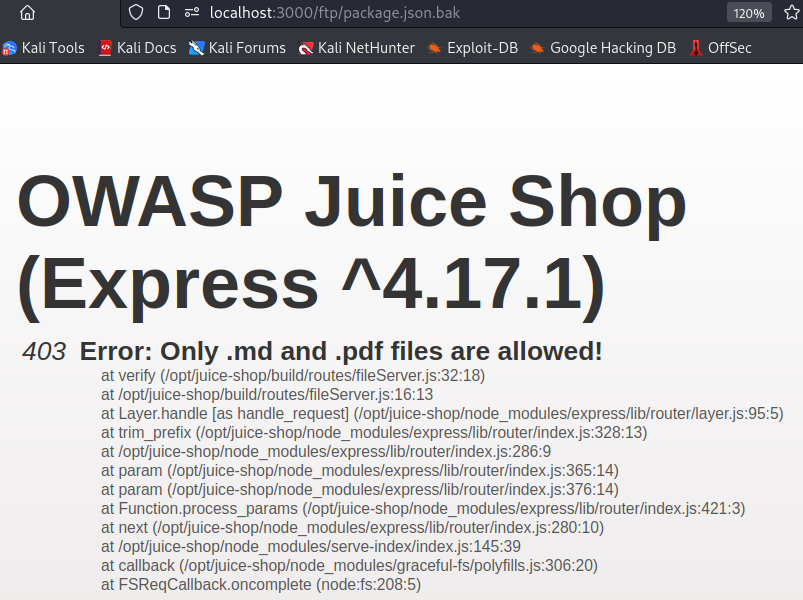
Udało się znaleźć 2 pliki access log, jednak zadanie nie zostało odznaczone jako ukończone w Score Board, dlatego postanowiono pobrać jeden z plików. Tym samym udało się ukończyć wyzwanie.

## Forgotten Developer Backup & Poison Null Byte

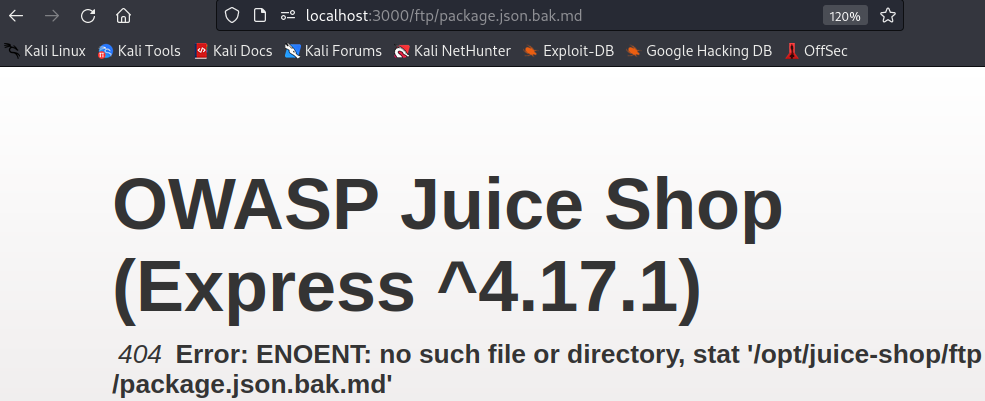
W tym zadaniu chodzi o znalezienie pliku „backup”. W pierwszej kolejności przeszukano miejsce, gdzie wcześniej udało się znaleźć wiele plików, czyli FTP:



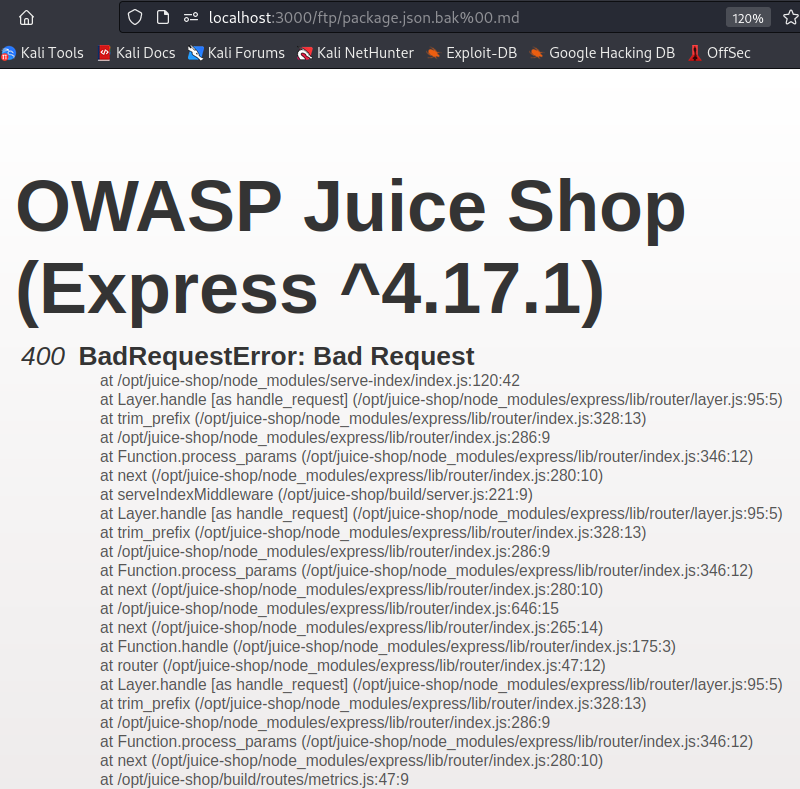
Od razu można zauważyć pliki z rozszerzeniem .bak, który wskazuje na backup. Spróbowano pobrać jeden z nich:



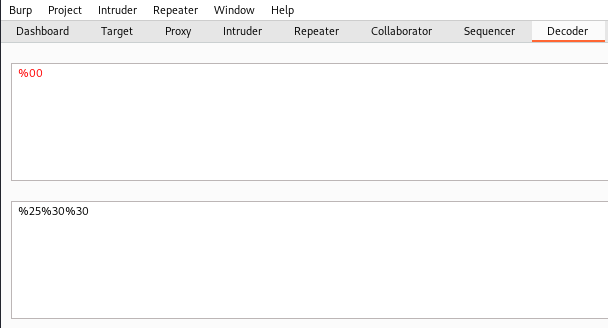
Jak widać, wyłącznie pliku o rozszerzeniami .md oraz .pdf są dozwolone. Postanowiono dopisać rozszerzenie, jednak nie przyniosło do oczekiwanego skutku:



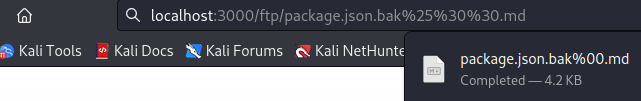
Spróbowano pobrać inne z tych plików, aby poszukać jakichś wskazówek. Niestety nie udało się znaleźć nic interesującego. Poszukano podpowiedzi – okazuje się, że należy zastosować atak Null byte injection. Chodzi o dopisanie %00 przed rozszerzeniem:



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



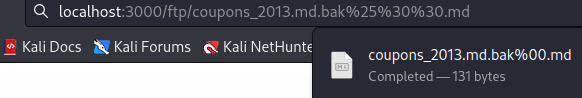
Wpisano to w adres URL:



Udało się rozwiązać przez przypadek 2 wyzwania jednocześnie.

## Forgotten Sales Backup

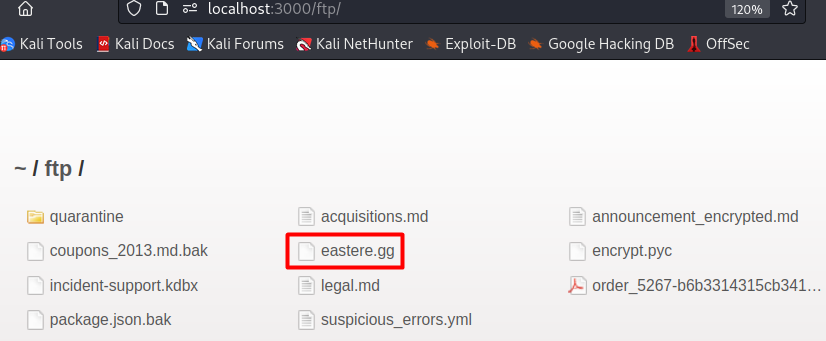
W wyzwaniu chodzi o pobranie drugiego pliku, tym razem prawdopodobnie coupons\_2013.md.bak. Spróbowano pobrać plik tym samym sposobem:



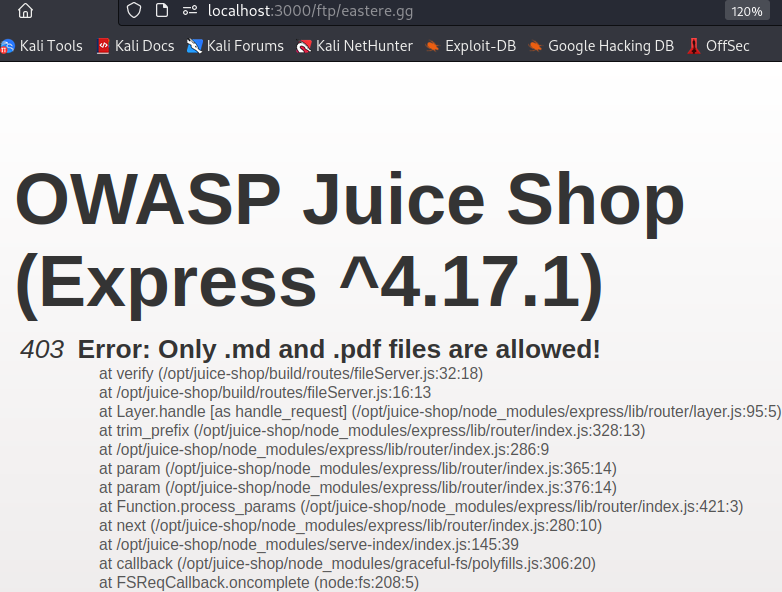
Udało się pobrać w ten sam sposób i ukończyć wyzwanie.

## Easter Egg

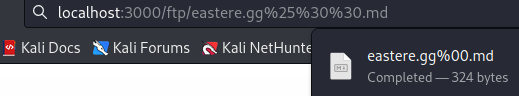
Wyzwanie jest związane z pobraniem Easter Egga. Ponownie w katalogu /ftp można zauważyć docelowy plik – eastere.gg:



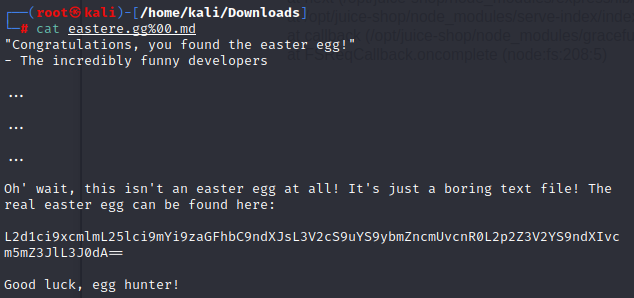
Spróbowano go pobrać:



Kolejny raz postanowiono zastosować ten sam sposób do pobrania pliku:



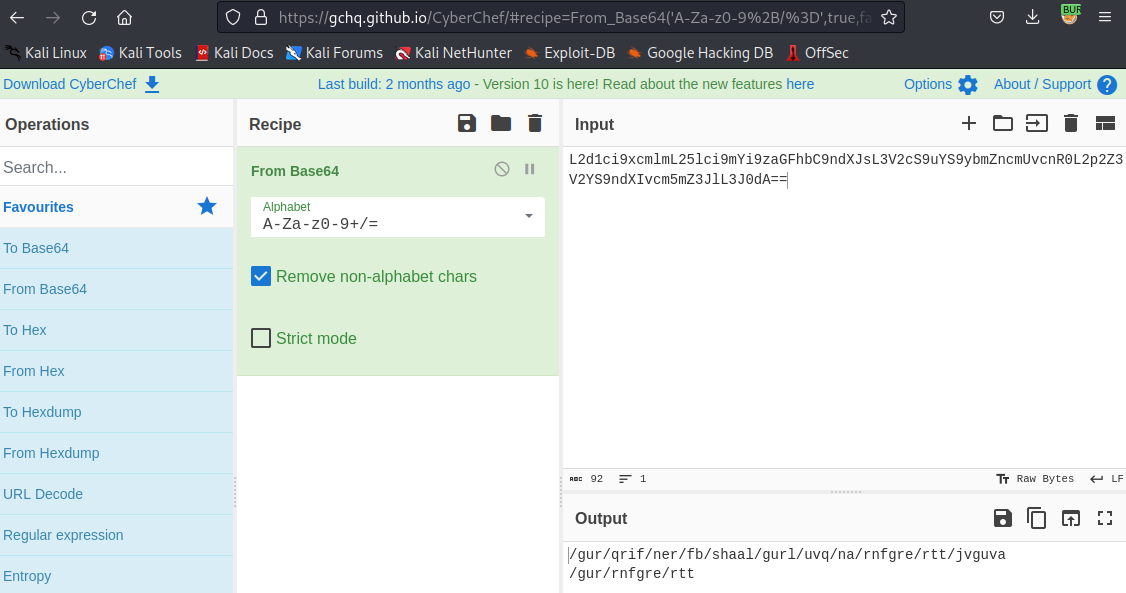
Po pobraniu sprawdzono narzędziem cat, co znajduje się wewnątrz pliku:



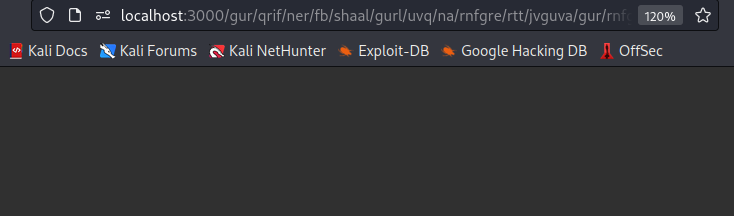
Na tym etapie wyzwanie zostało już ukończone, jednak postanowiono sprawdzić, czym jest ciąg znaków zawarty w pliku.

## Nested Easter Egg

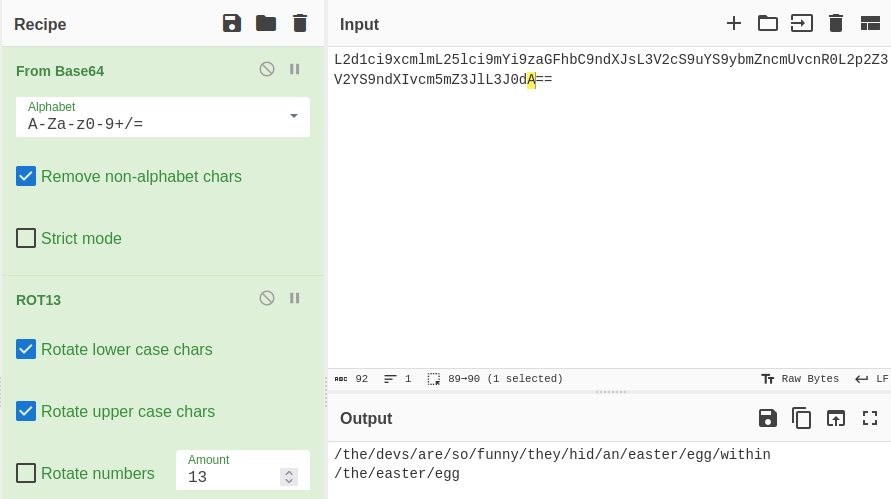
Wyzwanie polega na kryptoanalizie pliku eastere.gg. Na pierwszy rzut oka można zauważyć == na końcu, co może wskazywać na Base64. Wrzucono ten ciąg do CyberChefa:



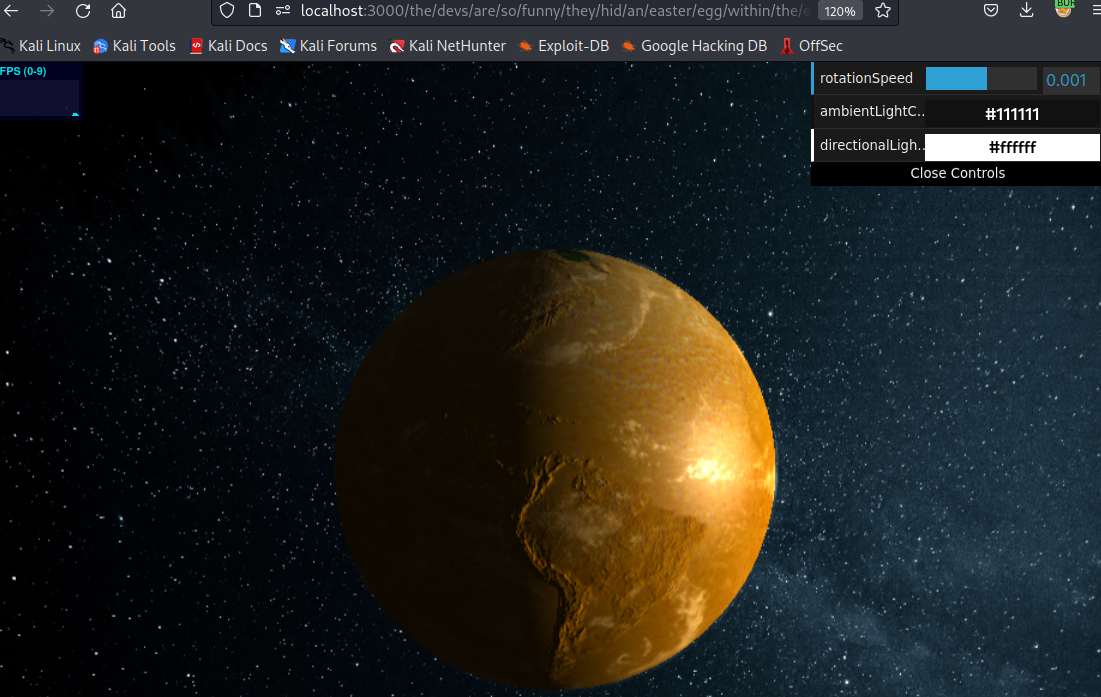
Odnaleziono ścieżkę, na pierwszy rzut oka wygląda na ponownie zaszyfrowaną jakimś szyfrem przestawnym, np. ROT13. Mimo wszystko sprawdzono najpierw, czy ta ścieżka do czegoś prowadzi:



Następnie spróbowano wykorzystać szyfr ROT13:



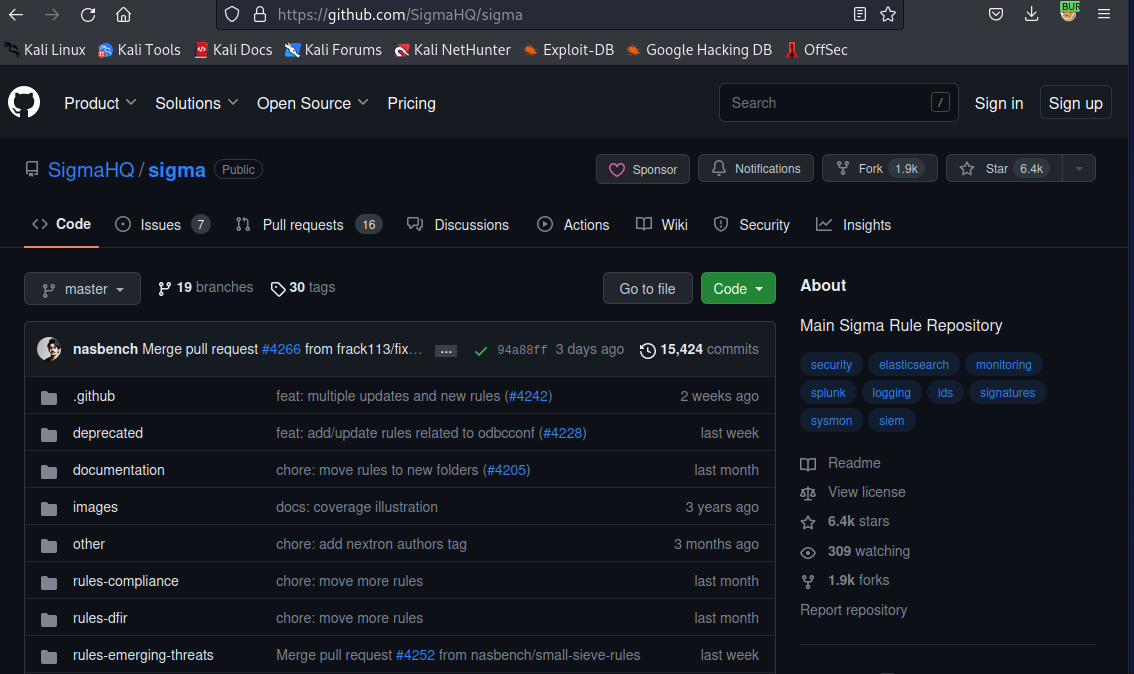
Udało się znaleźć ścieżkę, która niestety nie wygląda na ścieżkę. Mimo tego sprawdzono, czy prowadzi do czegoś:



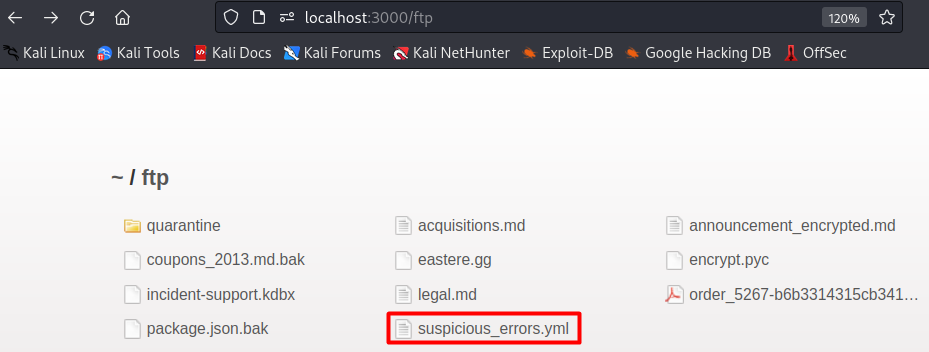
W ten sposób udało się ukończyć wyzwanie.

## Misplaced Signature File

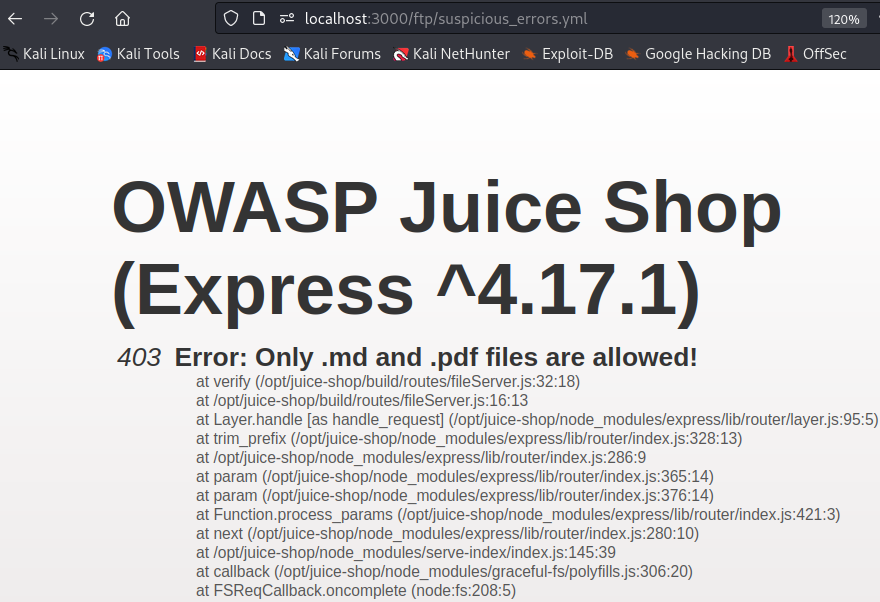
Wyzwanie polega na znalezieniu pliku z podpisem SIEM. Link przekierowuje na githuba:



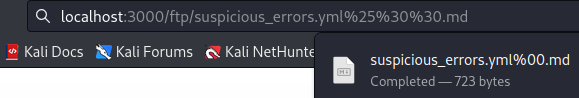
Przykładowe pliki znajdujące się tam, są o rozszerzeniu .yml. Ponownie sprawdzano katalog ftp, gdzie jeden plik jest właśnie w tym formacie: suspicious\_errors.yml.



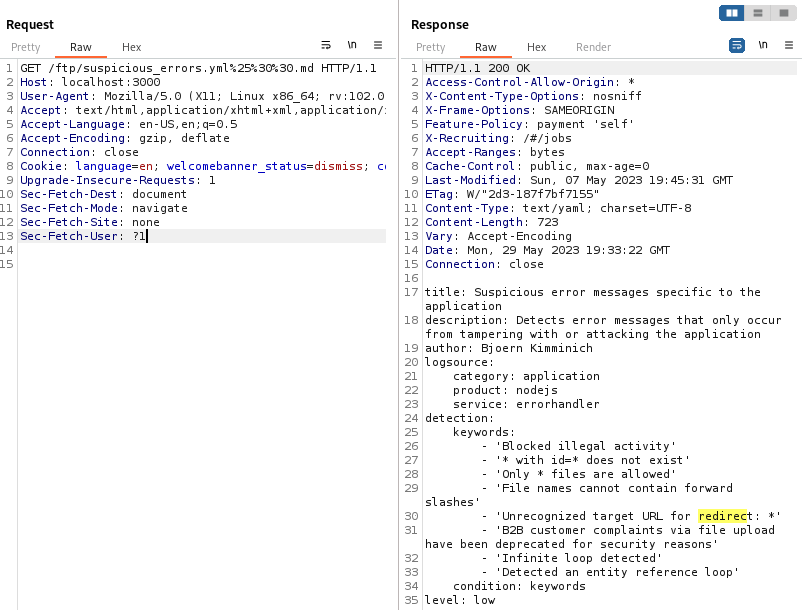
Jak w poprzednich przypadkach – pliku nie da się pobrać:



Spróbowano ponownie użyć Poison Null Byte:

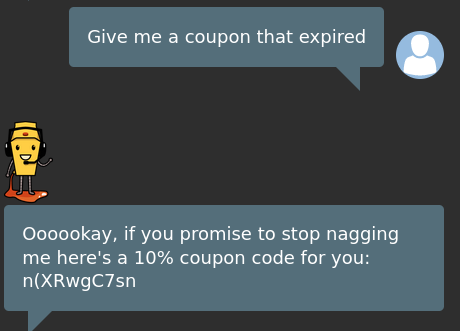


Plik udało się pobrać:

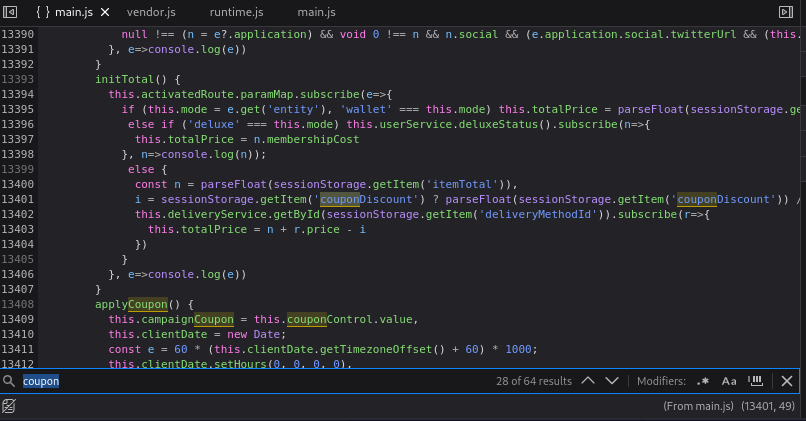


## Expired Coupon

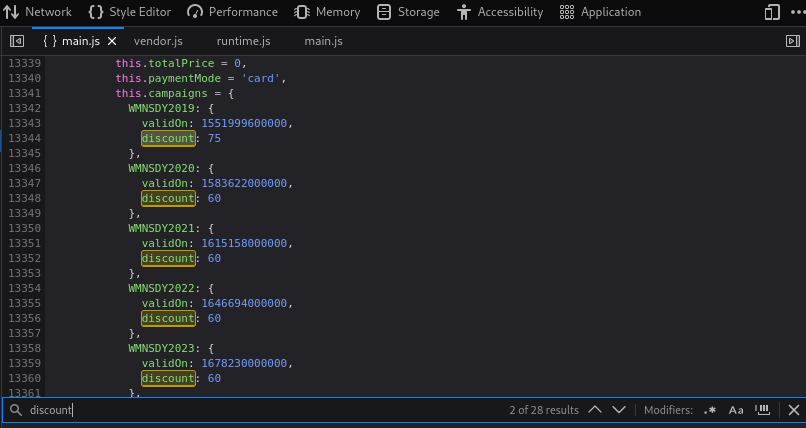
Zadanie polega na zastosowanie kuponu, który już wygasł. Z uwagi na fakt, że w 1-gwiazdkowym challenge’u pojawił się bot, który przekazał kupon, w pierwszej kolejności spytano go o przestarzałe kupony:



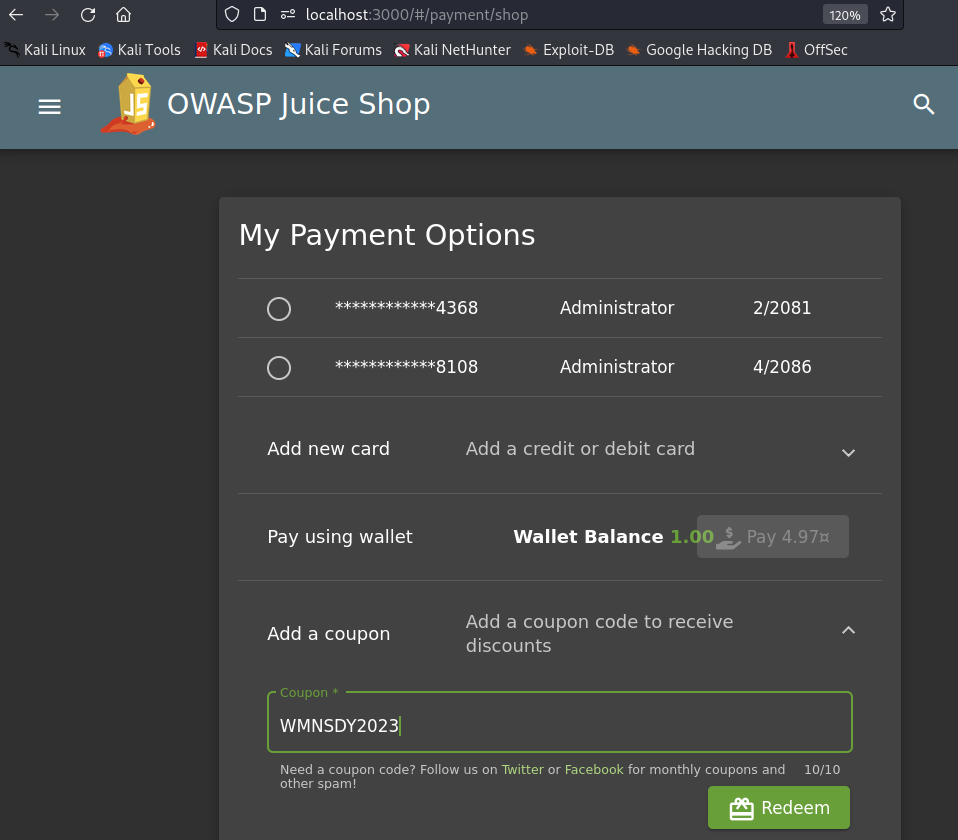
Niestety bot podał tylko aktualny kod rabatowy. W związku z tym postanowiono ponownie przeszukać kod strony:



Spróbowano także odszukać słowo „discount”:



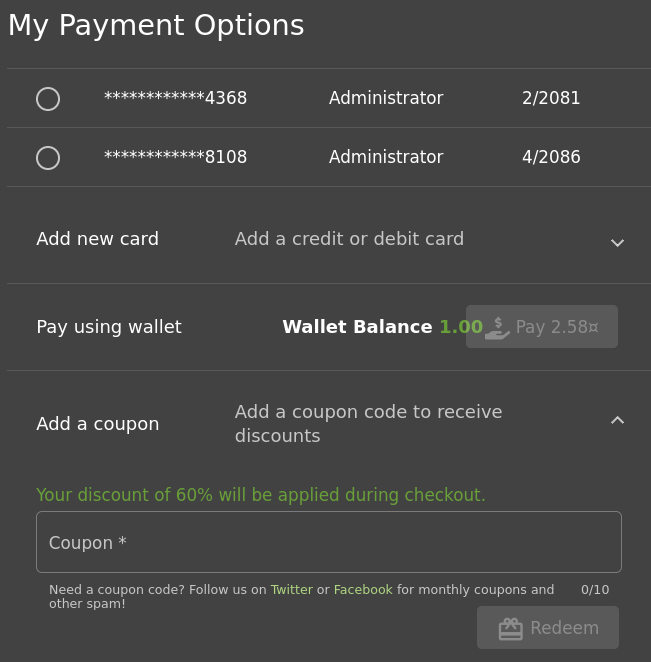
Znaleziono kilka kodów rabatowych. Spróbowano użyć tego ostatniego:



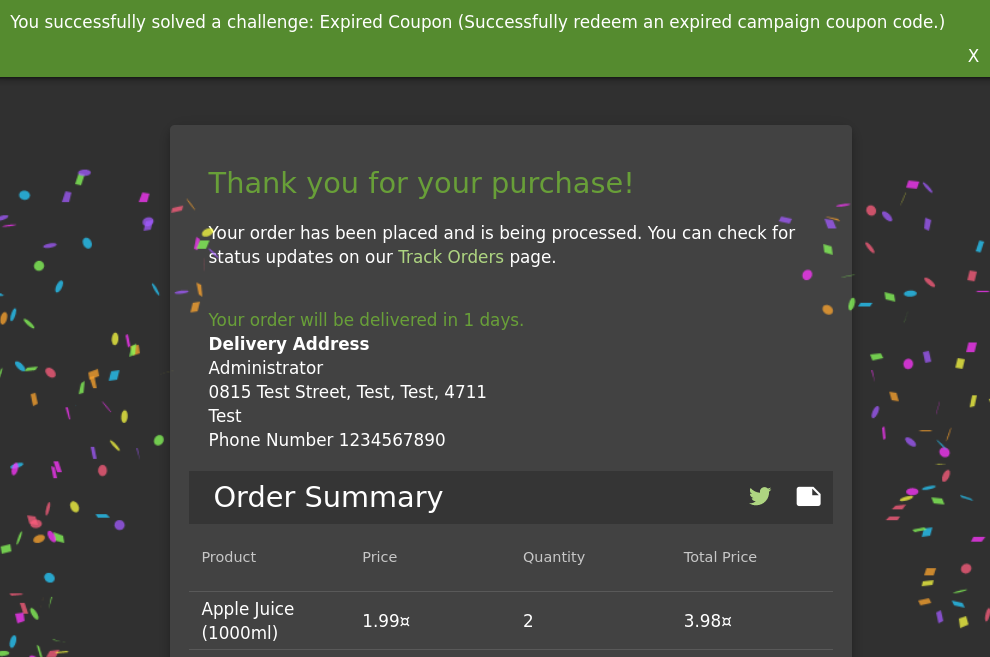
Wyświetliła się informacja, że kupon jest nieważny. Kupon wskazuje na dzień kobiet (women’s day), dlatego postanowiono zmienić czas wirtualnej maszynie za pomocą timedatectl:



W ten sposób udało się zastosować kod:



Następnie dokonano zakupu:

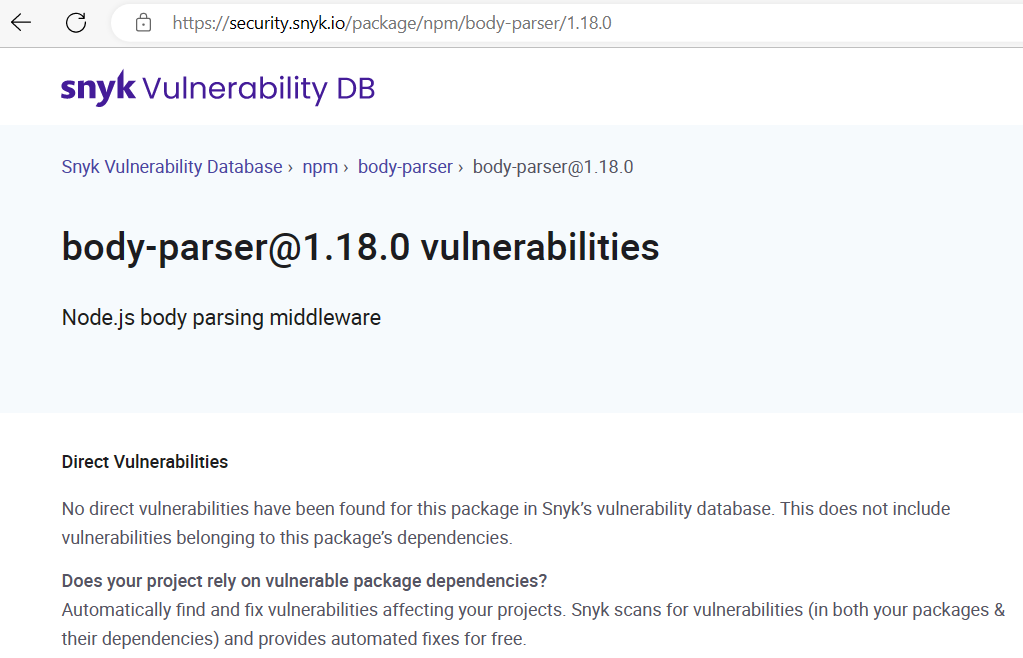


## Vulnerable Library

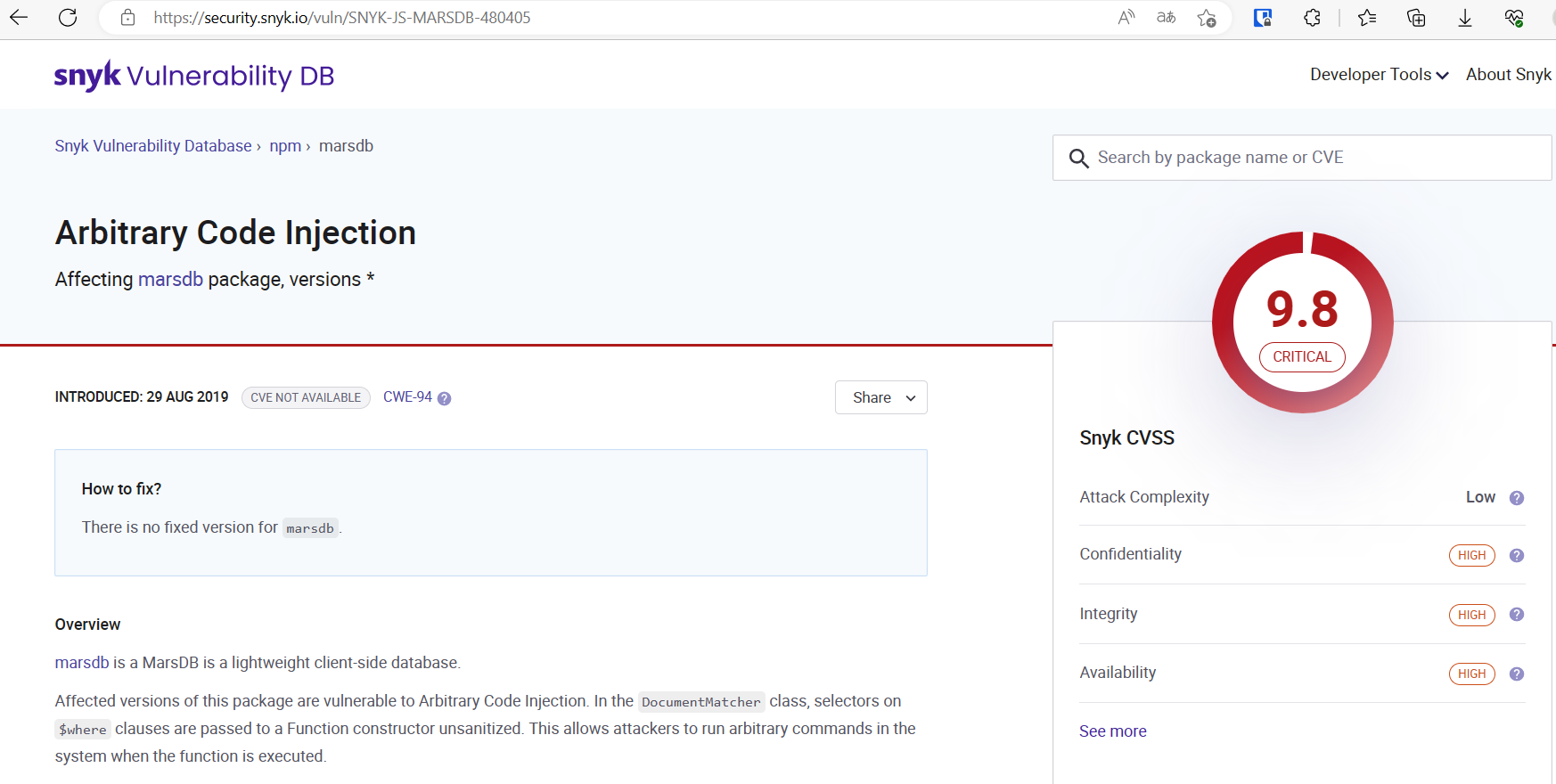
Zadanie jest podobne do wcześniejszego wyzwania, w którym należało wysłać feedback do sklepu, wówczas odnośnie słabego algorytmu kryptograficznego. Tym razem należy poinformować o podatnej bibliotece. Wszystkie biblioteki udało się znaleźć w pliku package.json.bak:



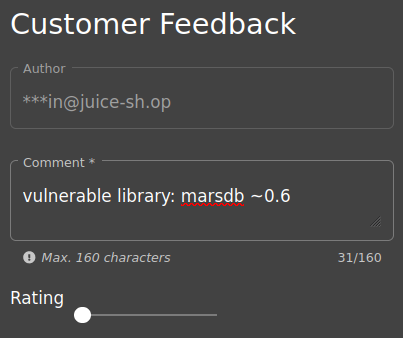
Po kolei sprawdzono w Internecie, które biblioteki są podatne w następujący sposób:



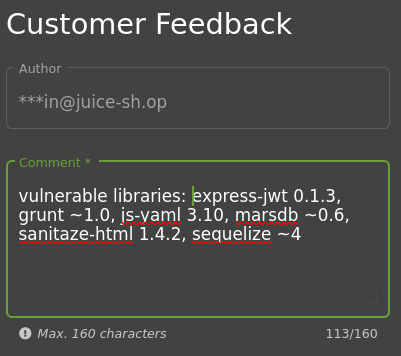
Znaleziono podatności w bibliotece marsdb 0.6:



Postanowiono sprawdzić, czy o tę bibliotekę chodzi:



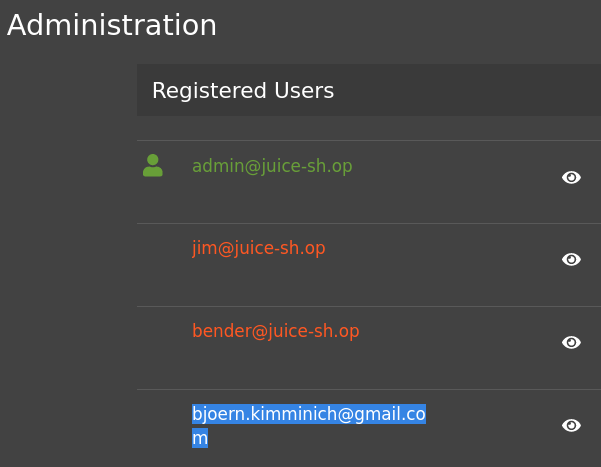
Niestety to nie wystarczyło, poszukano kolejnych bibliotek. Znaleziono jeszcze kilka podatnych i poinformowano sklep o nich:



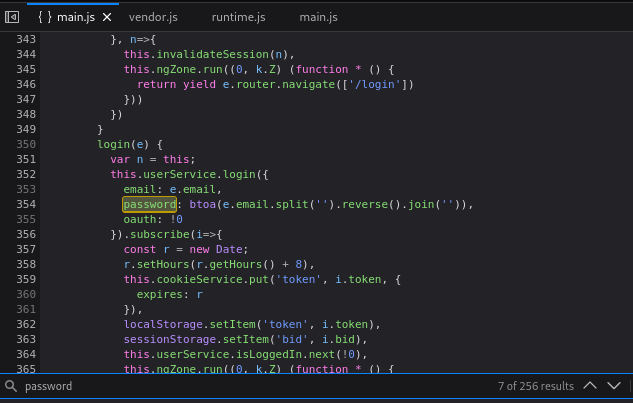
Dopiero w ten sposób udało się rozwiązać zadanie.

## Login Bjoern

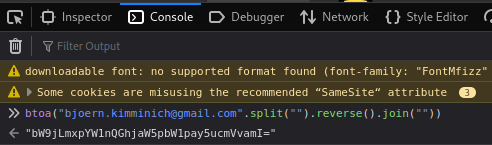
Wyzwanie polega na zalogowaniu się jako Bjoern (na konto gmail) bez zmieniania hasła, SQL injection, czy też hackowania konta. Jako pierwszy krok wyświetlono listę użytkowników na koncie admina:



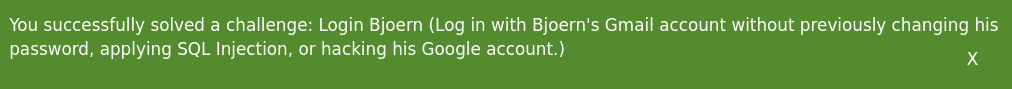
Adres Bjoerna: bjoern.kimminich@gmail.com. Z braku pomysłów postanowiono przeszukać ponownie kod strony:



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:

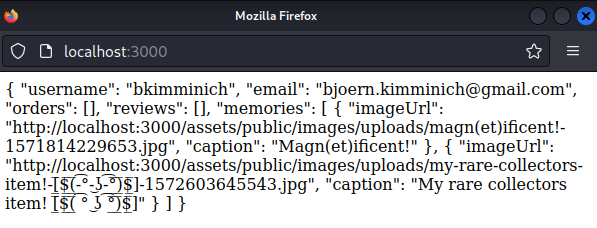


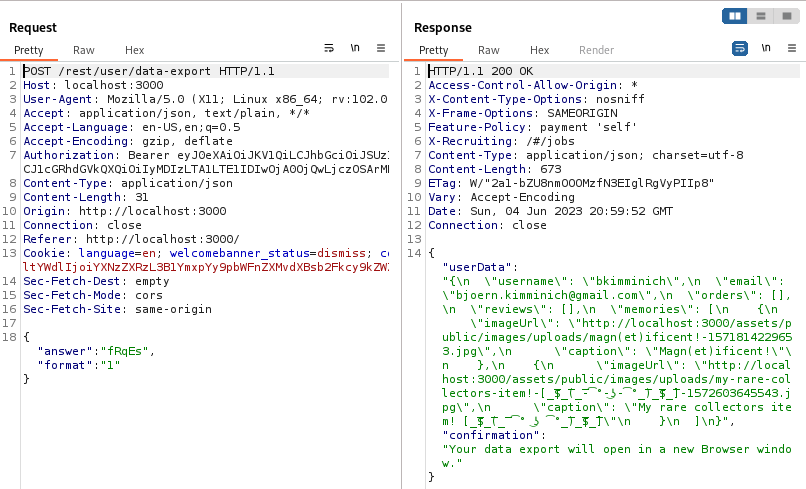
Spróbowano wpisać ciąg znaków (bW9jLmxpYW1nQGhjaW5pbW1pay5ucmVvamI=) jako hasło:



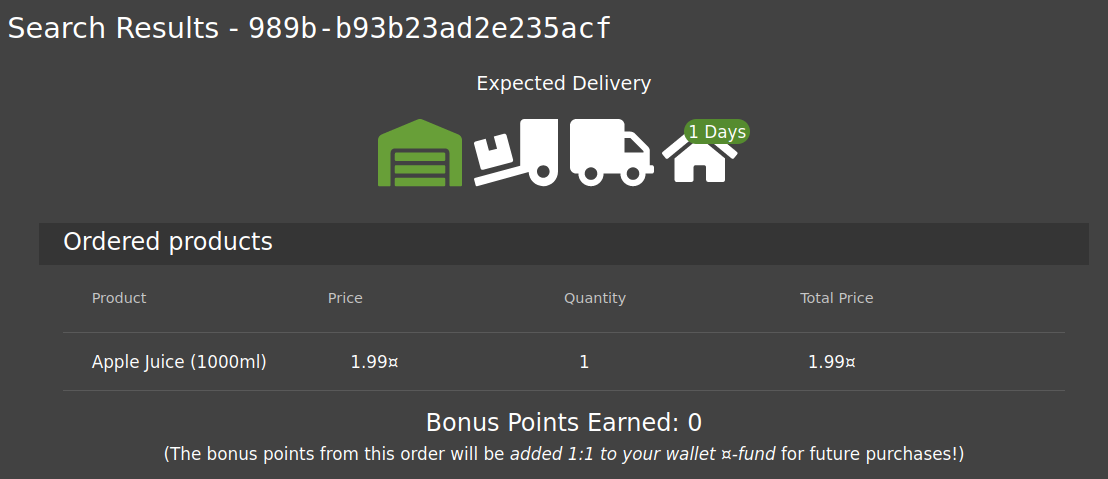
## GDPR Data Theft

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





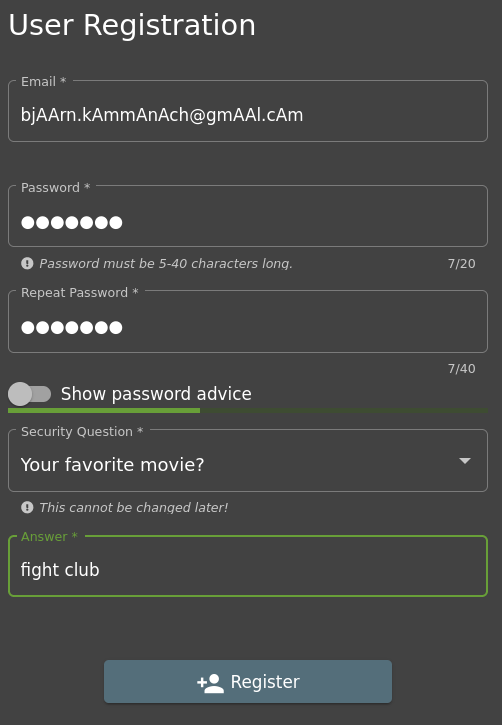
Zgodnie ze wskazówkami, warto przyjrzeć się procesowi zakupu.



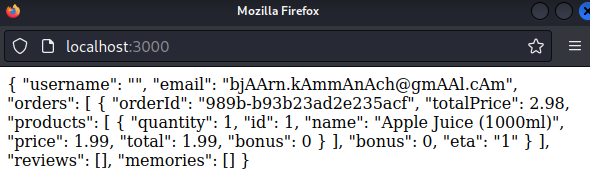
Znaleziono interesującą rzecz, email został wygwiazdkowany:



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

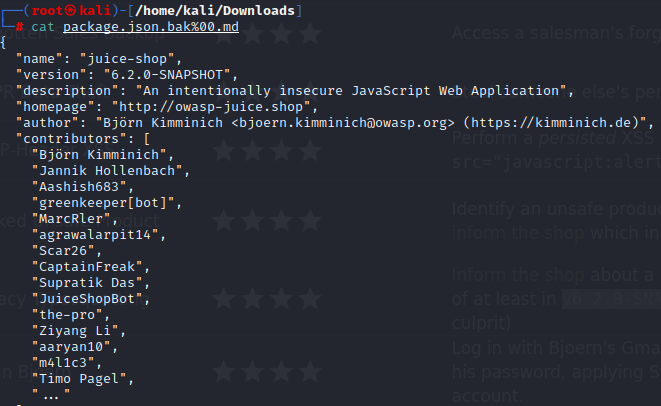


Zalogowano się na utworzone konto i pobrano dane, w ten sposób udało się rozwiązać zadanie:

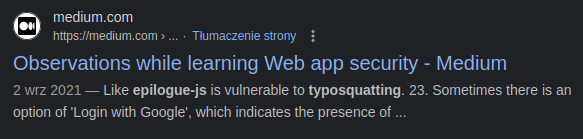


## Legacy Typosquatting

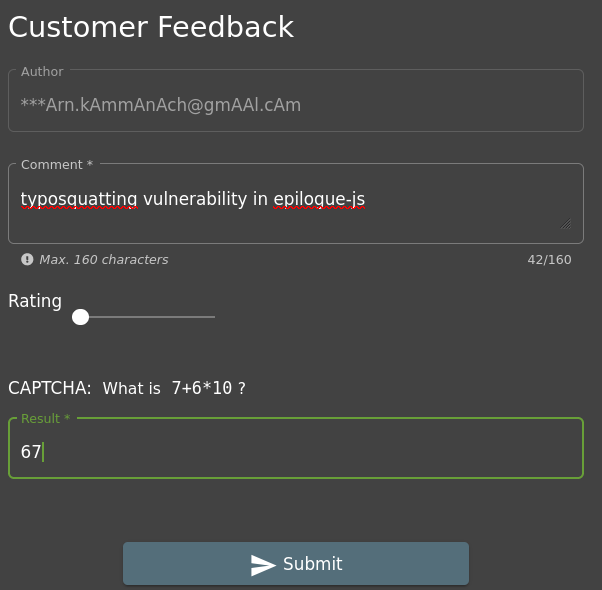
Zadanie polega na poinformowaniu sklepu o typosquattingu powiązanym ze snapshotem v6.2.0-SNAPSHOT. Typosquatting jest to utworzeniem strony o podobnej nazwie, np. ze zmienioną literą. Zajrzano ponownie do pliku backup, ponieważ pojawił się tam właśnie ten snapshot:



Postanowiono wpisać nazwy bibliotek z dopiskiem ‘typosquatting’.



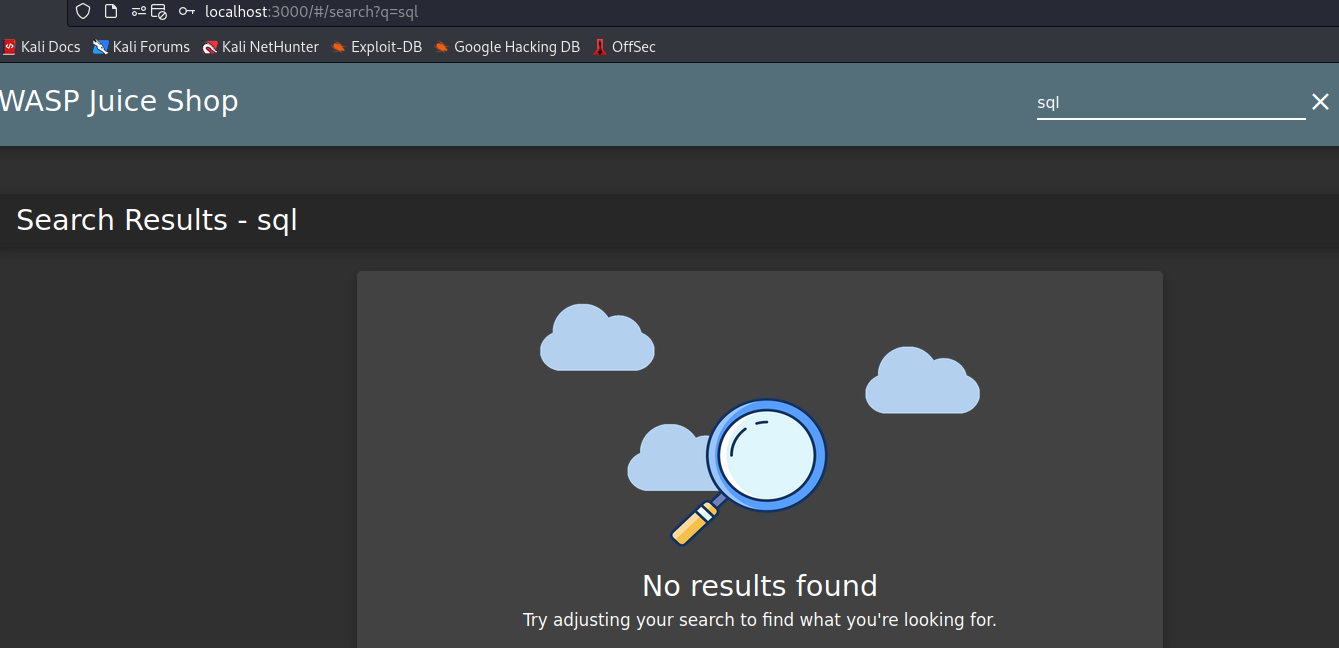
Wpisano epilogue-js jako feedback:



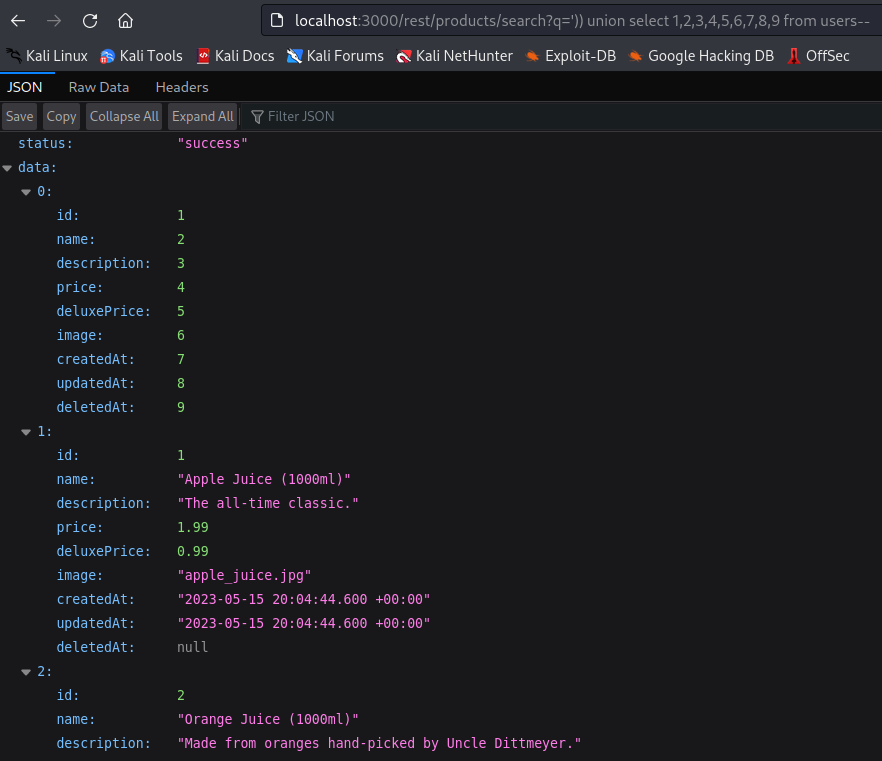
W ten sposób ukończono wyzwanie.

## User credentials

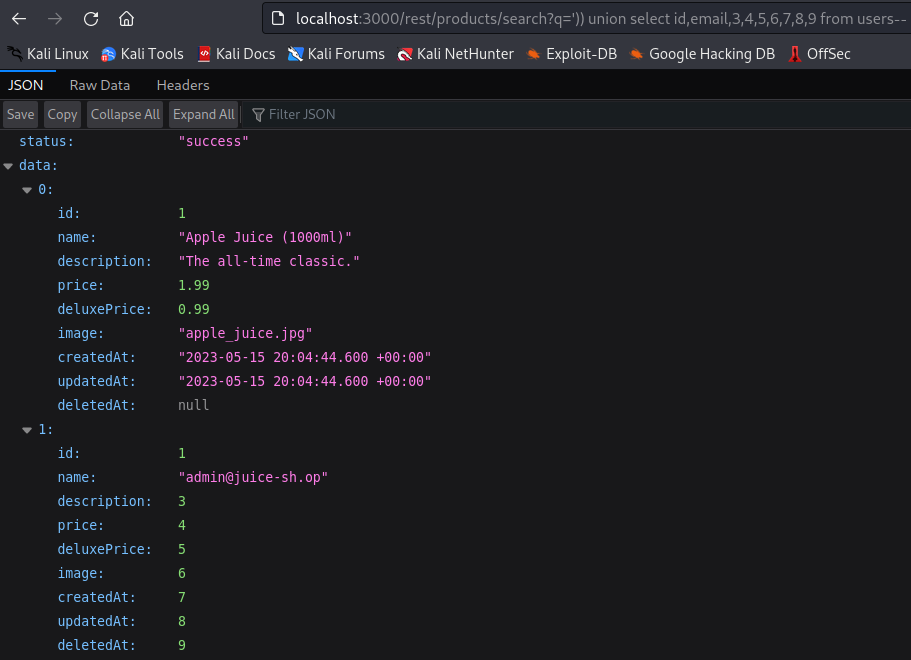
Za pomocą SQL injection, należy zdobyć dane uwierzytelniające wszystkich użytkowników.



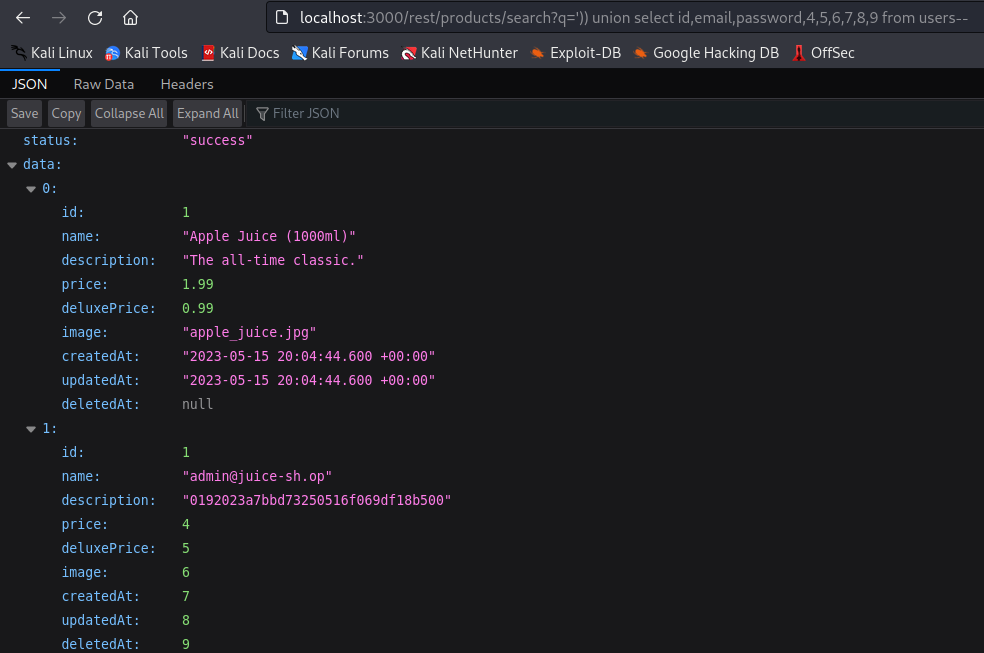
Zastosowano atak SQL injection union-based. Przykładowe tabele, które próbowano odgadnąć to users. Udało się znaleźć część danych, jednak nie pozwoliło to na wykonanie wyzwania:



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



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



## Christmas Special

W tym zadaniu należy zdobyć (dodać do koszyka i ‘zakupić’) specjalny box, dostępny tylko na święta 2014. W celu znalezienia go poszukiwania rozpoczęto od przeszukiwania listy produktów dostępnych w sklepie Juice Shop.

Obraz zawierający tekst, butelka, zrzut ekranu, napój

Opis wygenerowany automatycznie

W sklepie znajdowało się 35 produktów, nie było tam jednak żadnych boxów specjalnych, w tym świątecznego z roku 2014.

Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne, design

Opis wygenerowany automatycznie

Również przeszukiwanie sklepu różnymi frazami dotyczącymi tego boxa nie przyniosło rezultatów.

W narzędziu developer można jednak było zauważyć, że podczas przeszukiwania sklepu za pomocą funkcji ‘search’ wysyłane jest żądanie GET na adres http://localhost:3000/rest/products/search?q=

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Po przejściu na ten adres ukazała się lista wszystkich produktów w formacie JSON.

Obraz zawierający tekst, oprogramowanie, Oprogramowanie multimedialne, Ikona komputerowa

Opis wygenerowany automatycznie

Niestety tutaj również wyszukiwania fraz takich jakich ‘christmas’, ‘box’, czy ‘2014’ nie przyniosło żadnych efektów. Jednak przyglądając się liście produktów można było zauważyć, że ich ID, mimo że nie ułożone po kolei, są tworzone prawdopodobnie inkrementując o 1 poprzedni identyfikator produktu. Spisane zostały więc wszystkie wartości ID produktów z tej podstrony.

Po ułożeniu ich w kolejności, okazało się, że są numery, których brakuje – były to 10, 11, 12, 27, 28, 31, 39, 40

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

Możliwe więc, że są to produkty, które kiedyś były na stronie, lecz zostały usunięte. Pasowałoby to do produktu limitowanego na święta 2014.

Nie udało się jednak znaleźć adresu, który byłby podstroną konkretnego produktu. Po kliknięciu na produkt z listy na stronie juice shop, nie byliśmy przenoszeni na inną stronę, lecz otwierało się okno dialogowe.

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Nie pozwoliło to na próby przejścia na adres nie istniejącego już produktu z bazy.

W narzędziu developer zauważono jednak, że podczas dodawania produktu do koszyka, wysyłane jest żądanie POST o następujących parametrach:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Posiadając tą informację można było spróbować dodać do koszyka nieistniejący produkt, wysyłając odpowiednio zmodyfikowane żądanie POST.

Po wysłaniu zmodyfikowanego żądania ze zmienionym ID produktu na 10 (z listy brakujących), do koszyka został dodany box świąteczny z roku 2014.

Po podaniu danych do wysyłki i złożeniu zamówienia zadanie zostało rozwiązane:

Obraz zawierający zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne, tekst

Opis wygenerowany automatycznie

## NoSQL Manipulation

Zadanie polega na dodaniu wielu opinii do produktów w tym samym czasie. Jest ono również otagowane jako Injection, co może oznaczać, że powinniśmy wstrzyknąć jakąś wartość do żądania wysyłanego na serwer. W podpowiedziach do zadania znaleziono link do operatorów żądań bazy MongoDB, której używa aplikacja Juice Shop - <https://www.mongodb.com/docs/manual/reference/operator/query/>. Przyglądając się tym operatorom, obiecująco prezentowały się te dotyczące porównań, który pozwalały np. na dopasowanie wszystkich wartości mniejszych/większych/nierównych tej podanej w żądaniu.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Oznacza to, że wysyłając żądanie z ID produktu, serwer dopasuje wszystkie inne ID, które np. będą większe lub nierówne podanemu ID. W celu sprawdzenia, czy rzeczywiście tak będzie to działać, dodana została opinia do jednego z produktów, by zobaczyć jakie żądanie jest wtedy wysyłane na serwer.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Tak wyglądało wynikającego z tego żądanie PUT. Nie pozwala to na modyfikacje parametrem ID, ponieważ jest on zakodowany w URL, a nie w parametrach przekazywanych do bazy. Do tego ID musiałoby być podawane w ciele żądania.

Podjęta więc została próba, w której ID zostało usunięte z adresu i dodane w sekcji body, jako operator Mongo DB biorący za ID każdą wartość większą od 0. Powinno to dopasować żądanie do każdego produktu z bazy aplikacji. Skończyło się to jednak niepowodzeniem, ponieważ serwer zwrócił wiadomość, że nie rozpoznaje takiej ścieżki.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

W poszukiwaniu informacji w internecie, natrafiliśmy na informację, że do zmiany częściowej informacji dla wielu produktów, lepszą metodą niż PUT powinno być PATCH. Po zmianie metody HTTP na PATCH udało się wysłać żadanie, jednak dalej nie poskutowało to dodaniem żadnej recenzji.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Po wypróbowaniu innych operatorów porównań z bazy MongoDB sukces przyniosło użycie operatora $ne:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

## Reset Bender’s Password

W tym zadaniu należy zresetować hasło użytkownika Bender. W zadaniu można doczytać, że Bender to imię fikcyjnej postaci z serialu ‘Futurama’ i należy korzystać podczas OSINTU z publicznych źródeł.

Z poprzednich zadań znany jest już adres email tego użytkownika: bender@juice-sh.op

Po przejściu na podstronę forgot-password służącą do resetowania hasła użytkownikom, którzy go zapomnieli, możemy zobaczyć jakie pytanie bezpieczeństwa ustawione ma użytkownik Bender.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Wiadomo już więc, jakiej informacji należy szukać w internecie.

Ponieważ zadanie mówiło o szukaniu w ogólnodostępnych źródłach, pierwszym tropem była Wikipedia. Polska wersja strony nie zawierała wielu informacji, poszukiwania przeniosły się więc na jej angielskojęzyczny odpowiednik. Szukając informacji o miejscu pracy Bendera natrafiono na fragment idealnie opisujący poszukiwaną informację:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Przed innymi wydarzeniami z serialu, Bender miał pracę przy produkcji ‘suicide booths’. Niestety, nie ma podanej nazwy firmy, dla której wykonywał to zlecenie, a to jest przedmiotem pytania bezpieczeństwa. Kierując się jednak do odnośnika dotyczącego tych urządzeń, w sekcji dotyczącej Futuramy, można natrafić na wzmiankę o marce ‘Stop-and-Drop’.

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Nie było to jednak prawidłowa odpowiedzią. Przeszukując dalej internet, na stronie prowadzonej przez fanów serialu, futurama.fandom.com, zauważone zostało, że w rzeczywistości firma ta nazwana była tak samo, lecz inaczej pisana była jej nazwa – co źle było przedstawione na Wikipedii.

Obraz zawierający tekst, zrzut ekranu, Kubeł na śmieci

Opis wygenerowany automatycznie

Po wpisaniu Stop’n’Drop jako odpowiedź na pytanie bezpieczeństwa Bendera udało się zmienić hasło i zaliczyć zadanie.

Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne, oprogramowanie

Opis wygenerowany automatycznie

## Steganography

Kolejne zadanie polegało na steganografii. Według opisu zadania należało wydać postać ukrywającą się w widocznym miejscu w sklepie. Łącząc to z tytułem zadania, chodziło o znalezienie postaci, która była zaszyta w jakimś zdjęciu dostępnym na stronie i napisać w polu ‘Contact’ na stronie dokładną nazwę tej postaci.

W tym celu użyte zostało narzędzie OpenStego. Pierwszym tropem była podstrona http://localhost:3000/#/photo-wall, zawierająca kilka zdjęć. Mimo obiecującej zawartości, żaden z obrazów nie krył za sobą żadnej postaci.

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Kolejnym miejscem, gdzie znaleziono obrazy, była podstrona ‘about’ a na niej sekcja ‘customer feedback’.

Nie dało się ich jednak pobrać bezpośrednio z tamtego miejsca, ponieważ była to ‘karuzela’ opinii użytkowników. Zamiast tego zbadano element. Pokazały się nazwy wszystkich plików ze zdjęciami:

Obraz zawierający tekst, Czcionka, numer, Strona internetowa

Opis wygenerowany automatycznie

Po przejściu na adres URL znaleziony w ten sposób można było pobrać obrazy http://localhost:3000/assets/public/images/carousel/1.jpg

Pobrany w ten sposób obraz 5.png z sukcesem został rozszyfrowany przez oprogramowanie.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Obraz, który został wyekstraktowany z 5.png wyglądał następująco:

Obraz zawierający kreskówka, Postać fikcyjna, Animacja, zabawka

Opis wygenerowany automatycznie

Jest to postać Pickle Rick z serialu Rick and Morty. Po wpisaniu tej nazwy w pole ‘Contact’ w komentarzu udało się rozwiązać zadanie.