## API-only XSS oraz Forged Review

Zadanie polega na wykonaniu ataku XSS, lecz w tym przypadku konieczne będzie wykorzystanie wystawionego przez aplikację API. Głównym celem wyzwania jest wykazanie, że aplikacja nieodpowiednio waliduje, oczyszcza lub chroni dane wejściowe otrzymane z zapytań API, co umożliwia wstrzyknięcie i wykonanie kodu JavaScript w aplikacji OWASP Juice Shop. By zrealizować zadanie należy wyświetlić odpowiedni alert, którego treść potwierdza wykonanie ataku XSS.

Pierwszym krokiem w celu odnalezienia API, które mogłoby posłużyć jako cel ataku, było skorzystanie z wbudowanej w przeglądarkę sposobności podglądu kodu pliku main.js, w którym wyszukana została fraza „api”.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Wyświetlonych zostało kilka różnych API – na przykład: powiązane z użytkownikami oraz produktami z menu. Z uwagi na fakt, iż produkty dostępne są z poziomu bazowej strony, wykorzystane zostało ów API.

W celu zainicjowania połączenia z API skierowanym do obsługi produktów zamieszczony został testowy wpis, a następnie ciąg wydarzeń został wyszukany w aplikacji BurpSuite pozwalającej na prześledzenie poszczególnych zapytań kierowanych do aplikacji webowej.

Obraz zawierający tekst, zrzut ekranu, Oprogramowanie multimedialne, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Odnalezione zostało zapytanie kierowane bezpośrednio do API

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

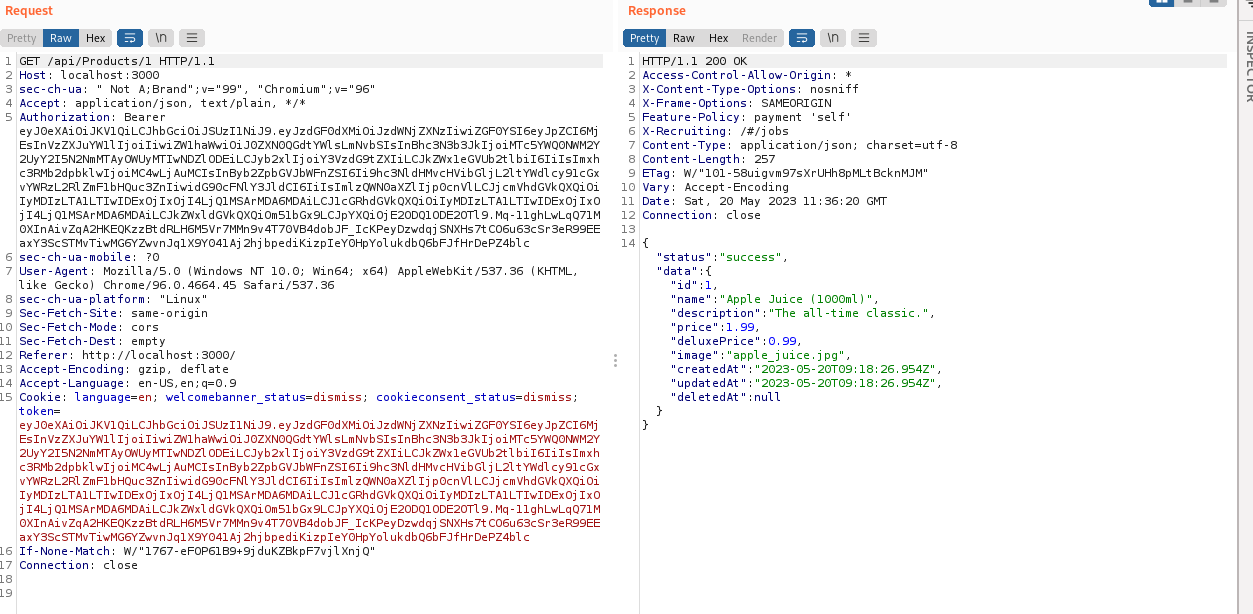
Opis wygenerowany automatycznie

Następnie zmieniona została treść zapytania, tak by odwoływało się ono do API powiązanego z produktami dostępnymi z poziomu głównego menu. Metoda GET doprowadziła do wyświetlenia informacji powiązanych ze wszystkimi rodzajami soków.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Kolejnym krokiem było odfiltrowanie jednego soku, na którym skupiona zostanie uwaga w dalszych krokach przeprowadzanego ataku. W tym przypadku będzie to sok jabłkowy.

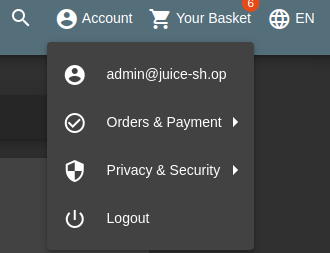


Skupiając swoją uwagę już wyłącznie na soku jabłkowym, wykonano próbę przesłania metody PUT z przekazywanymi atrybutami soku, mając nadzieję, iż dojdzie do zmiany wartości. Próba ta została jednak odrzucona, a zmiana nie została wprowadzona w obrębie aplikacji.

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Pierwszym podejrzeniem odnoszącym się do powodu braku powodzenia było powiązanie konkretnych ograniczeń uprawnień, które może posiadać w obrębie swojego konta przecięty użytkownik portalu. Wykonana została zatem próba zalogowania się jako użytkownik administracyjny, przy wykorzystaniu metody opisywanej w jednym ze wcześniejszych rozdziałów – było to bowiem również dostępne wyzwanie.



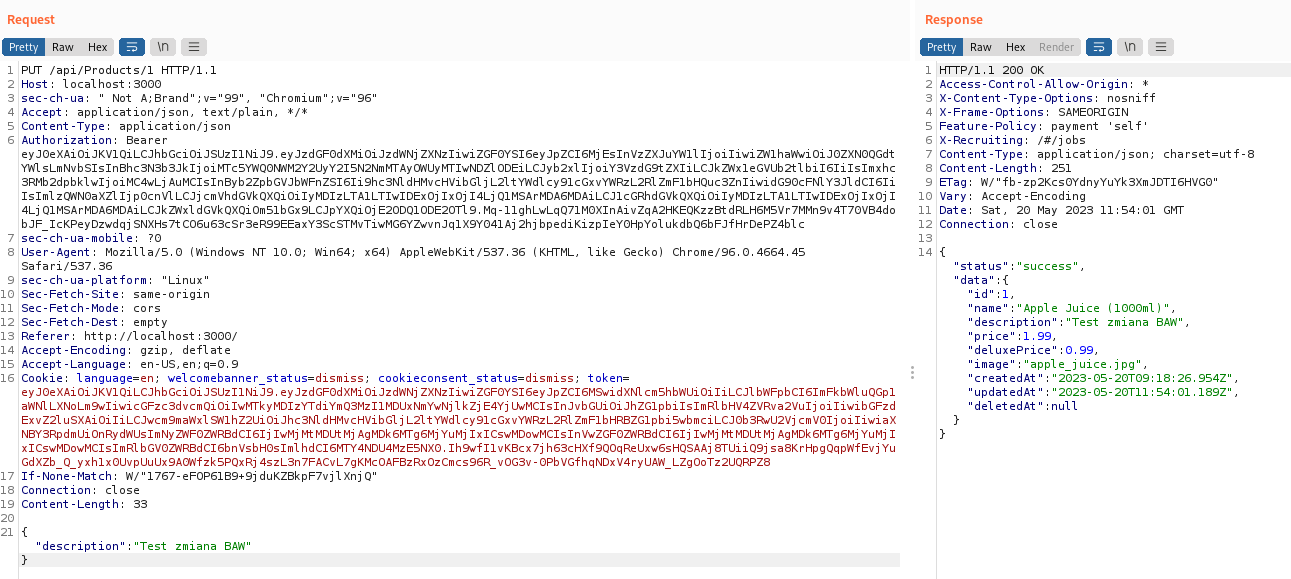
Po poprawnym zalogowaniu się na konto administratora, możliwe było wykonanie dowolnej akcji (tutaj wykorzystana, przeprowadzona akcja logowania się na portal) w celu pozyskania tokenu odpowiadającego za powiązane danej sesji z konkretnym użytkownikiem.

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie

Wspomniany token przekazany został jako podmieniona wartość jednego z atrybutów metody PUT, dzięki czemu wysyłane żądanie było obarczone powiązaniem z administratorem, który przeważnie posiada znacząco wyższe uprawnienia względem standardowego użytkownika.

Ponownie zmieniona została wartość parametru description odpowiedzialnego za opis pozycji z menu. Tym razem jednak akcja zakończyła się powodzeniem, a tym samym zmianą treści opisu na głównej stronie Juice Shop.



Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

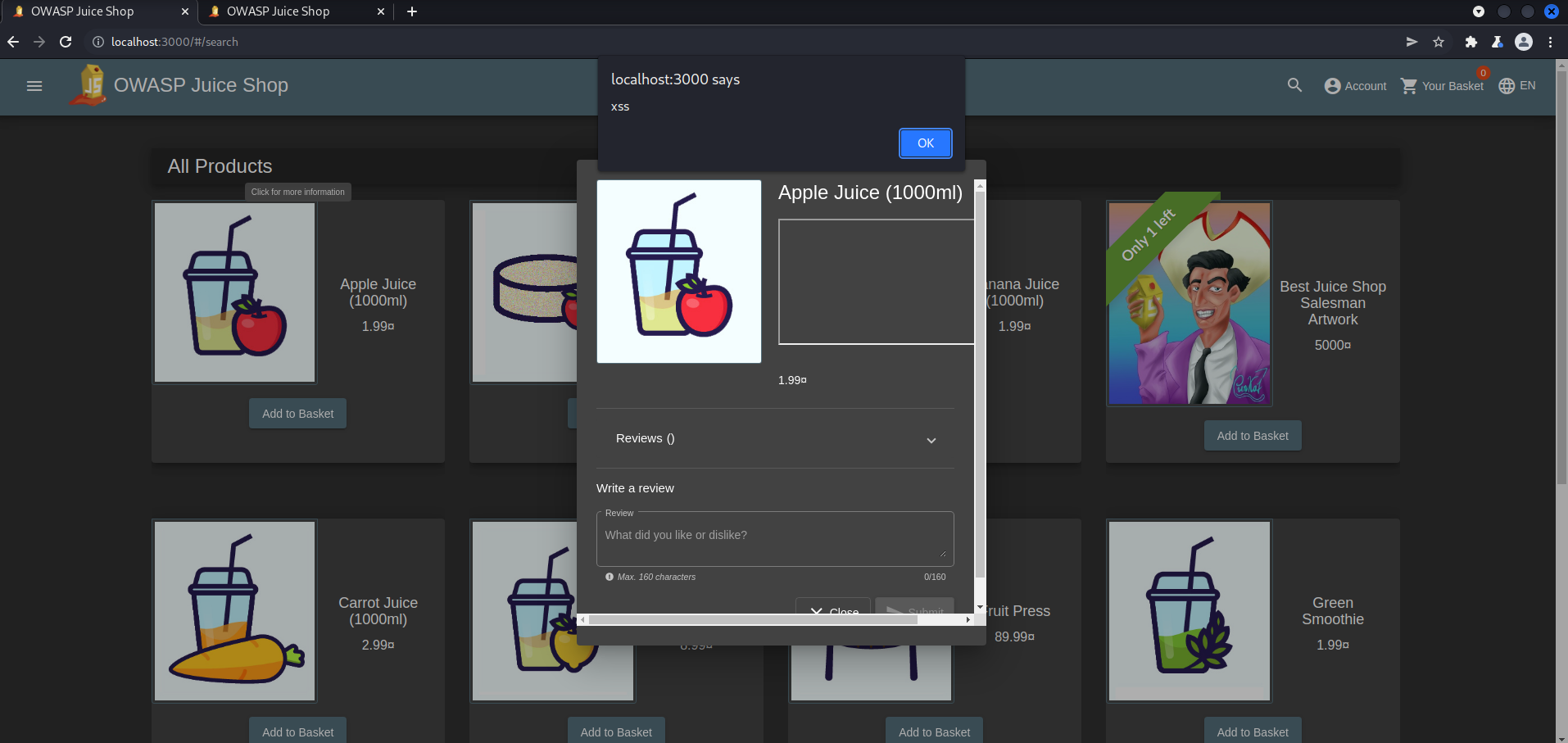
Powyższe grafiki potwierdzają wprowadzoną w obrębie produktu zmianę. Czynność ta zobrazowała, że przytaczane pole tekstowe wchodzi w pełną interakcję z użytkownikiem. Ponadto wykonanie opisywanych akcji przyniosło skutek w postaci uznania przez systemem wyzwania towarzyszącego, którego celem było dokonanie zmiany w obrębie recenzji produktu z konta innego użytkownika, niż to które utworzył użytkownik. Tutaj nie doszło stricte do zmiany treści recenzji, lecz opisu produktu, natomiast – co wynika z przebiegu eksperymentu – to również pozwalało na uznanie wyzwania.

Pozostałą czynnością do wykonania poprawnego ataku XSS realizowanego przy wykorzystaniu API Juice Shop było dokonanie zmiany w obrębie wykorzystywanego pola tekstowego, oraz zaimplementowanie przykładowego kodu XSS (tutaj: wyświetla on komunikat o błędzie o treści „XSS”).

Obraz zawierający tekst, Czcionka, numer, zrzut ekranu

Opis wygenerowany automatycznie

Metoda PUT została zaakceptowana przez API, a zmiana wprowadzona w obrębie aplikacji webowej. W celu weryfikacji poprawności przeprowadzonej akcji udano się na główną stronę Juice Shop, wybrano z dostępnych pozycji sok jabłkowy, któremu poświęcane były wszystkie akcje. Wszystkie te kroki poskutkowały uznaniem wyzwania oraz wyświetleniem komunikatu o błędzie, którego treść odpowiadała wcześniej wprowadzonej w obrębie BurpSuite wartości: „XSS”.



## Client-side XSS Protection

Pierwszym krokiem, który był Konieczny do wykonania w obrębie wyzwania Client-side XSS Protection było zidentyfikowania dowolnej formy zabezpieczeń, które nałożone są na użytkownika Juice Shop. Pierwszym polem tekstowym, które posiadało pewną formę filtracji wprowadzanej treści i które zostało odkryte w ramach poszukiwań był obszar przeznaczony do wpisywania adresu mailowego w formularzu rejestracji nowego użytkownika.



Konieczne było wprowadzenie tekstu, który realnie mógłby reprezentować adres mailowy, a zatem podlegał pod format składni z uwzględnieniem domeny (na przykład: @gmail.com). Wartość została zatem zmieniona na [qwerty@gmail.com](mailto:qwerty@gmail.com), tak by odpowiednio dostosować się do wymogów pola tekstowego. Akcja zakończyła się powodzeniem, a użytkownik został dodany do bazy, co potwierdza wycinek ruchu sieciowego widoczny w narzędziu BurpSuite.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Następnie wartość znajdująca się w polu tekstowym zabezpieczonym przez kod aplikacji webowej odpowiednim zabezpieczeniem odnoszącym się do formatu treści została zmodyfikowana. W obrębie treści wykorzystany został krótki skrypt, pozwalający na wyświetlenie komunikatu o błędzie, którego treść odnosiłaby się do formy ataku – XSS.

Akcja przeprowadzona przy wykorzystaniu BurpSuite Repeater zakończyła się powodzeniem i użytkownik został zaakceptowany.

Obraz zawierający tekst, linia, Czcionka, numer

Opis wygenerowany automatycznie

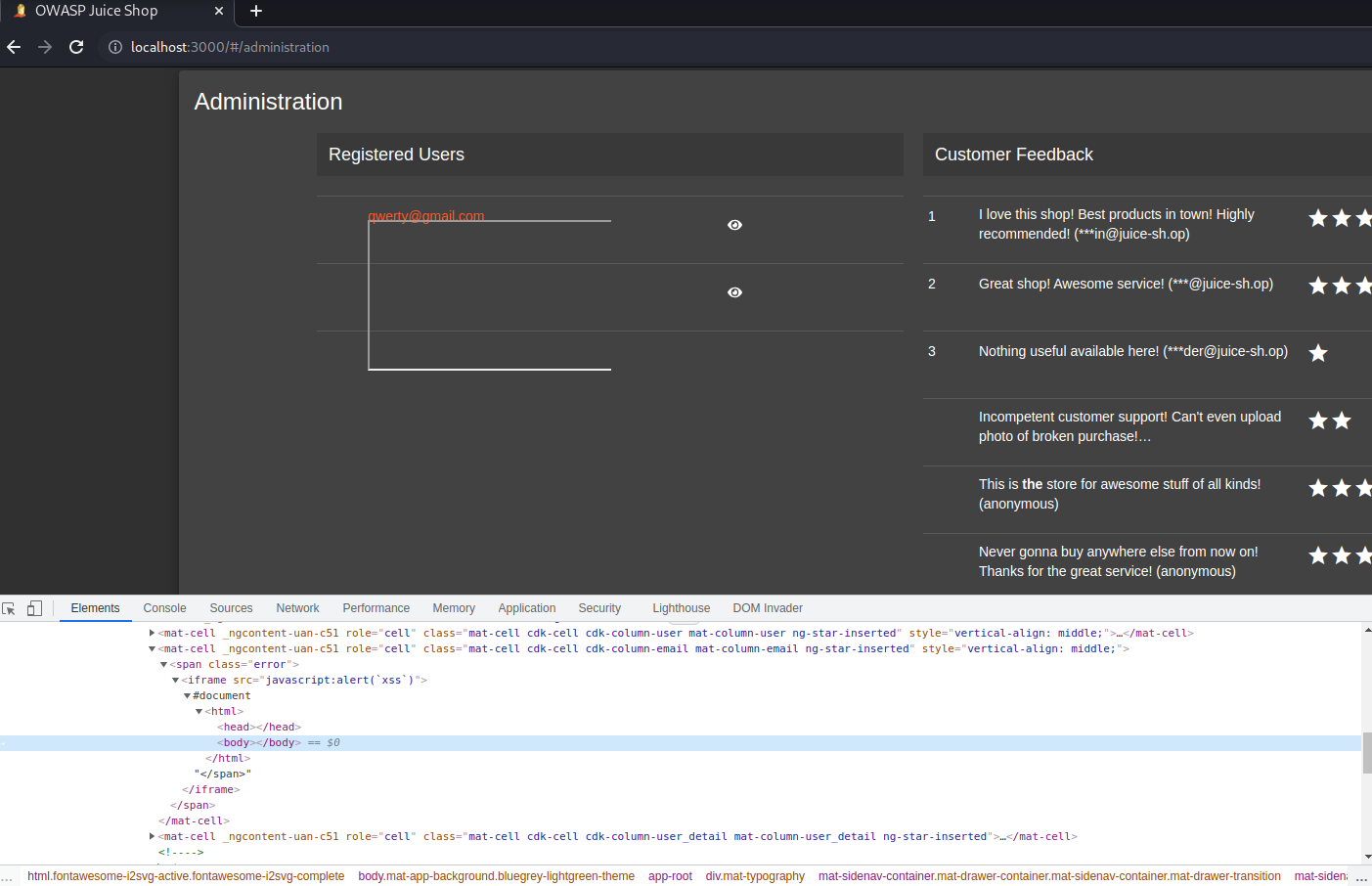
W kolejnym kroku, po przejściu na stronę główną OWASP Juice Shop widoczny był komunikat o uznaniu wyzwania. W celu weryfikacji przeprowadzonych działań, możliwe było przejście do panelu administracyjnego – po zalogowaniu się na koncie użytkownika z ów prawami.

Oczom ukazywał się komunikat strony o zadanej wcześniej treści. Następnie po przejściu do listy użytkowników i pominięciu pierwszych kilku stron listy, widoczne było puste pole odnoszące się do adresu mailowego poszczególnych użytkowników. W tym przypadku było to pole, w którym umiejscowiony został skrypt odpowiadający za wykonanie ataku XSS.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Wcześniejszą tezę można było potwierdzić przy wykorzystaniu narzędzi inspekcji wbudowanych w przeglądarkę internetową wyświetlającą zawartość aplikacji webowej. W obrębie pustej przestrzeni widoczna była treść przekazanego skryptu.



## Database Schema

W celu poznania schematu budowy bazy danych powiązanej z aplikacją webową, konieczne było zidentyfikowanie jakiego rodzaju jest to baza. W tym celu wykorzystano narzędzie wyszukiwania w obrębie głównej strony Juice Shop.

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Wyszukanie zostało następnie odnalezione w obrębie narzędzia Burp Suite, dzięki czemu można było przesłać treść zapytania GET do Repeatera, co przyczyniło się do możliwości wielokrotnego modyfikowania treści i obserwowania rezultatów.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

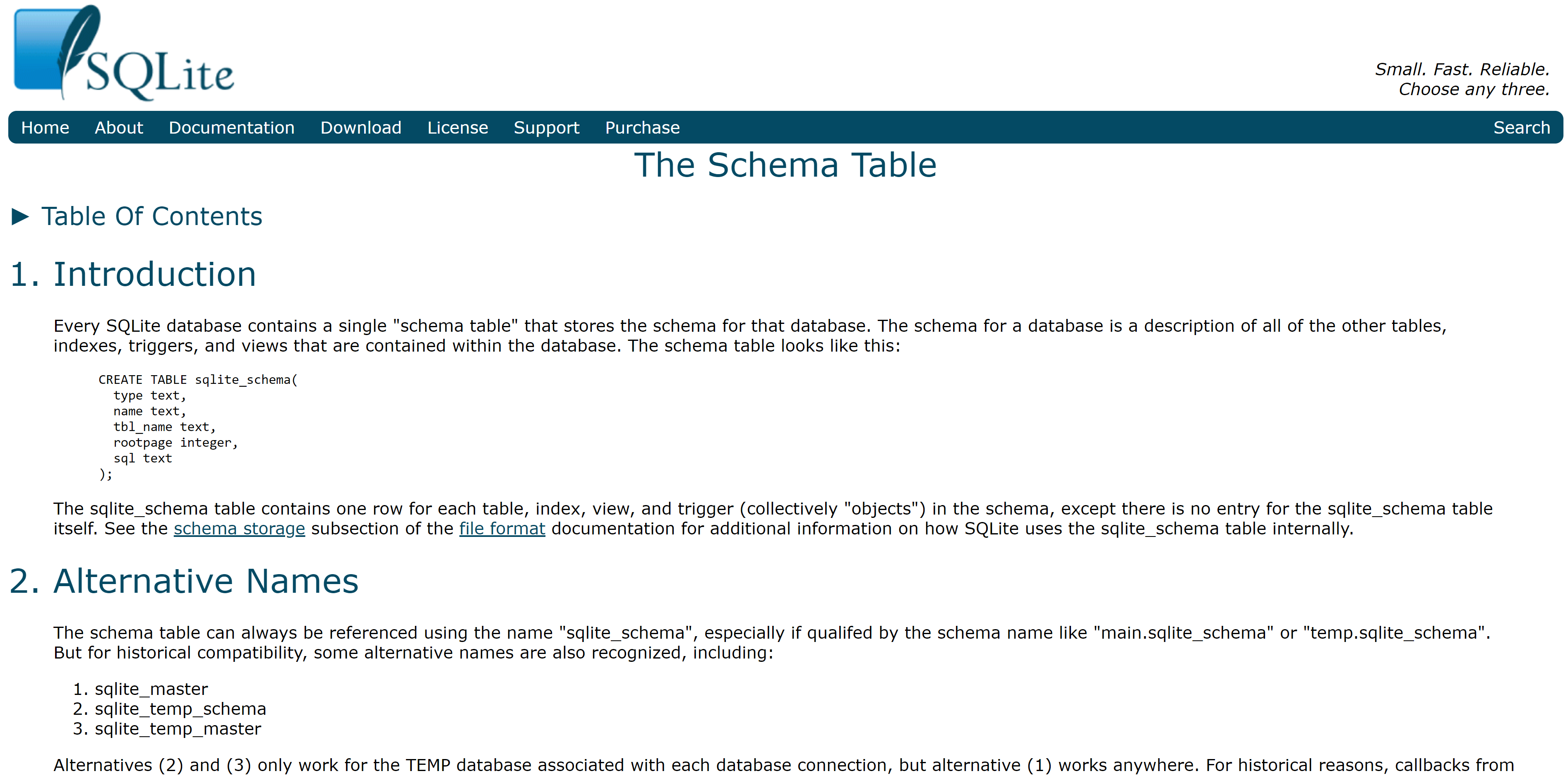
Pierwszymi wyszukanymi wartościami były te zawierające anglojęzyczne nazwy owoców, z których soki można zakupić z aplikacji webowej. Nie przynosiło to jednak żadnych dodatkowych informacji o bazie danych, oprócz samych parametrów poszczególnych pozycji menu.

Następnie rozpoczęto eksperymentowanie ze składnią typową dla ataków typu injection skierowanych w bazy danych.

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

Widoczny efekt pozwolił ocenić, iż zaimplementowana w obrębie aplikacji webowej baza danych to SQLite. To pozwoliło na wyszukiwanie w zasobach Internetu informacji odnoszących się do tego, w jaki sposób możliwe jest pozyskanie informacji o schemacie budowy tejże bazy.



Informacje pozyskane z oficjalnej strony SQLite pozwoliły ustalić, iż głównym zadaniem w obrębie widocznego wyzwania będzie poszukiwanie tabeli, która może przyjmować jedną z czterech nazw w zależności od danej wersji SQLite: sqlite\_schema, sqlite\_master, sqlite\_temp\_schema, sqlite\_temp\_master.

Ponowne eksperymenty związane z formami kierowanych w Burp Suite zapytań doprowadziły do pozyskania pozytywnego efektu i tym samym akceptacji zapytania. Forma budowy nowego parametru zapytania opierała się o wyświetlony wcześniej komunikat błędu.

Obraz zawierający tekst, Czcionka, numer, zrzut ekranu

Opis wygenerowany automatycznie

Kolejno przy każdej następnej modyfikacji dołączane były elementy składowe zapytania przekazywanego jako argument, które tworzone były wedle porad dostępnych na oficjalnej stronie SQLite.

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie

Obecnie wyświetlana treść błędu zawracanego przez bazę danych sugerowała niezgodność zapytania z dostępną w bazie danych liczbą kolumn. Była to odpowiedź na chęć wyświetlenia wszystkich danych (reprezentowane jako \*) z tabeli sqlite\_master, która – jak zostało wcześniej wspomniane – pozwala odkryć schemat bazy danych SQLite. Konieczne było zatem dostosowanie żądania do wspomnianych zaleceń. Wykorzystano do tego zadania metodę prób i błędów wychodząc z założenia, iż parametrów przypisanych do poszczególnych pozycji bazy danych nie jest znacząco dużo.

Obraz zawierający tekst, Czcionka, numer, zrzut ekranu

Opis wygenerowany automatycznie

Finalnie, po dotarciu do listy 9 parametrów wyświetlona została pozytywna odpowiedź, która przedstawiała listę parametrów poszczególnych pozycji z bazy danych.

Finalnym krokiem było wstawienie w miejsce jednego z parametrów nazwy własnej „sql”, dzięki czemu korzystając z wyrażenia o schemacie „SELECT sql FROM sqlite\_master” możliwe byłoby pozyskanie schematu bazy danych.

Obraz zawierający tekst, Czcionka, numer, linia

Opis wygenerowany automatycznie

Akcja przyniosła oczekiwany efekt w postaci podmiany parametru o indeksie 1 (tutaj był to numer identyfikujący dany obiekt) na wypisane składowe bazy danych. Pozostałe parametry, zważywszy na brak modyfikacji zapytania w ich obrębie pozostały niezmienione względem poprzednich prób.