Zadanie 1: Reflected XSS (Pasek Wyszukiwania)

- **Lokalizacja:** Pasek wyszukiwania (Search Bar) na stronie głównej.
- **Payload (Rozwiązanie):** <iframe src="javascript:alert(1)">
- **Instrukcja wykonania:**
    1. Kliknij ikonę lupy (wyszukiwanie).
    2. Wklej powyższy payload w pole tekstowe i wciśnij Enter.
- **Wyjaśnienie techniczne:** Klasyczny atak *Reflected XSS*. Aplikacja odbija dane wejściowe użytkownika bezpośrednio do widoku strony bez odpowiedniej sanityzacji. Przeglądarka interpretuje tag <iframe> jako kod HTML i wykonuje zawarty w nim JavaScript.

Zadanie 2: DOM XSS (Śledzenie Zamówień)

- **Lokalizacja:** Podstrona "Track Orders".
- **Payload (Rozwiązanie):** <iframe src="javascript:alert(1)">
- **Instrukcja wykonania:**
    1. Przejdź do zakładki "Track Orders".
    2. W polu ID zamówienia wpisz payload i kliknij "Track".
- **Wyjaśnienie techniczne:** Atak *DOM-based XSS*. Skrypt po stronie klienta (JavaScript) pobiera wartość z inputa i dynamicznie modyfikuje strukturę DOM, wstrzykując niebezpieczny kod. Dane nie muszą nawet trafiać do serwera.

Zadanie 3: Stored XSS (Rejestracja przez API)

- **Lokalizacja:** Formularz rejestracji / Narzędzia deweloperskie (F12).
- **Payload (Rozwiązanie):** "email": "<iframe src=javascript:alert(1)>@test.com"
- **Instrukcja wykonania:**
    1. Otwórz DevTools -> zakładka Network.
    2. Wypełnij formularz rejestracji, kliknij "Register".
    3. Znajdź request POST /api/Users, edytuj go (Edit and Resend).
    4. W ciele JSON podmień wartość email na payload i wyślij.
    5. Zaloguj się jako Admin (np. SQL Injection: ' OR 1=1 -- jako email).
    6. Wejdź w panel /administration – alert wyskoczy przy ładowaniu listy użytkowników.

- **Wyjaśnienie techniczne:** Walidacja frontendowa jest pomijana poprzez bezpośrednie zapytanie do API. Atak *Stored XSS* – kod zapisuje się w bazie danych i wykonuje u każdego (np. administratora), kto wyświetli ten rekord.

Zadanie 4: Obejście zabezpieczeń (Sanitizer Bypass)

- **Lokalizacja:** Formularz "Customer Feedback".
- **Payload (Rozwiązanie):** <<iframe src="javascript:evil"/>iframe src="javascript:alert(xss)">
- **Instrukcja wykonania:**
    1. Wybierz "Customer Feedback".
    2. W polu komentarza wklej payload, wypełnij resztę i wyślij (z CAPTCHA).
    3. Jako Admin wejdź na /administration, aby uruchomić kod.
- **Wyjaśnienie techniczne:** Błąd logiczny w bibliotece czyszczącej kod (sanitize-html v1.4.2). Usunięcie fragmentu javascript:evil powoduje "sklejenie się" pozostałych części w poprawny, złośliwy tag iframe.

Zadanie 5: Zaawansowane Obejście (Sanitizer + CSP Bypass)

- **Lokalizacja:** Profil użytkownika (Username i Image URL).
- **Payload 1 (Username):** <<a|ascript>alert('xss')</script>
- **Payload 2 (Image URL):** ; script-src 'unsafe-inline'
- **Instrukcja wykonania:**
    1. Ustaw Username na Payload 1 (obejście filtra usuwającego słowo "script").
    2. Ustaw Image URL na Payload 2 (obejście Content Security Policy).
    3. Kliknij "Set Username".
- **Wyjaśnienie techniczne:** Pierwszy payload omija filtr tekstowy (zagnieżdżenie tagów). Drugi payload wstrzykuje dyrektywę CSP unsafe-inline poprzez pole URL, osłabiając politykę bezpieczeństwa przeglądarki i pozwalając na wykonanie skryptu.

Zadanie 6: API Product Tampering

- **Cel:** Zmień nazwę produktu bezpośrednio w bazie danych poprzez API, omijając interfejs graficzny sklepu.
- **Wprowadzenie:** Aplikacje internetowe często komunikują się z serwerem za pomocą API. Jeśli deweloper nie zablokuje odpowiednio metod HTTP (takich jak PUT) dla zwykłych użytkowników, możliwe jest modyfikowanie danych, do których teoretycznie nie powinniśmy mieć dostępu (np. nazwy produktów czy ceny).
- **Instrukcja:**
    1. Otwórz narzędzia deweloperskie (F12) i przejdź do zakładki **Network**.
    2. Odśwież stronę sklepu i znajdź zapytanie zwracające listę produktów (np. GET /api/Products).
    3. Zidentyfikuj ID produktu, który chcesz zmienić (np. produkt nr 5).
    4. Przejdź do zakładki **Console** w przeglądarce.
    5. Skonstruuj i wyślij polecenie fetch, które metodą PUT nadpisze dane produktu. W ciele zapytania (body) ustaw format JSON i zmień nazwę produktu na payload XSS: <iframe src="javascript:alert('XSS')">.
- **Potwierdzenie:** Zrzut ekranu strony głównej sklepu, na którym nazwa produktu została zmieniona na puste pole (iframe), a na ekranie widoczny jest wyskakujący alert.

Zadanie 7: API Review Injection

- **Cel:** Dodaj recenzję pod produktem, wykorzystując bezpośrednie zapytanie do API.
- **Wprowadzenie:** Czasami formularze na stronie posiadają zabezpieczenia walidujące wprowadzane dane. Jednak API, które odbiera te dane, może być mniej restrykcyjne. Twoim zadaniem jest wysłanie recenzji "obok" formularza, bezpośrednio do serwera.
- **Instrukcja:**
    1. Wejdź w szczegóły dowolnego produktu i spróbuj dodać normalną recenzję, obserwując zakładkę **Network**, aby poznać strukturę zapytania (endpoint, np. /api/Reviews lub /rest/products/[id]/reviews).
    2. Przejdź do zakładki **Console**.

3. Przygotuj polecenie fetch z metodą PUT (lub POST, zależnie od obserwowanego wcześniej ruchu).
4. W treści wiadomości (payloadzie) umieść kod HTML wywołujący alert, zamiast zwykłego tekstu recenzji.
5. Wykonaj polecenie.

- **Potwierdzenie:** Zrzut ekranu podstrony produktu, na którym widać wyskakujący alert uruchomiony natychmiast po załadowaniu sekcji komentarzy.

Zadanie 8: API BOLA Comment Manipulation

- **Cel:** Zmodyfikuj treść recenzji należącej do innego użytkownika.
- **Wprowadzenie:** Podatność Broken Object Level Authorization (BOLA) występuje, gdy API pozwala na dostęp do obiektu (np. komentarza) jedynie na podstawie jego ID, nie sprawdzając, czy użytkownik wykonujący akcję jest jego właścicielem.
- **Instrukcja:**
    1. W zakładce **Network** znajdź zapytanie pobierające recenzje produktów.
    2. Zidentyfikuj ID recenzji (komentarza), która nie należy do Ciebie.
    3. W konsoli przeglądarki użyj polecenia fetch z metodą PATCH lub PUT, kierując je na endpoint konkretnej recenzji (np. /api/Reviews/[ID_cudzej_recenzji]).
    4. W ciele zapytania prześlij nową treść komentarza zawierającą złośliwy kod (payload).
    5. Odśwież stronę, aby zobaczyć efekt.
- **Potwierdzenie:** Zrzut ekranu sekcji recenzji, gdzie widoczny jest alert, a treść komentarza innego użytkownika została zmieniona przez Twój atak.