

Shape

**Kurs:**

Bezpieczeństwo aplikacji webowych

**Tytuł projektu:**

Damn Vulnerable Web Application

**Autorzy projektu:**

1. Robert Lotawiec 277027
2. Amadeusz Kołaczek 253097
3. Wojciech Marcinkowski

Wrocław, 2024

## Spis treści:

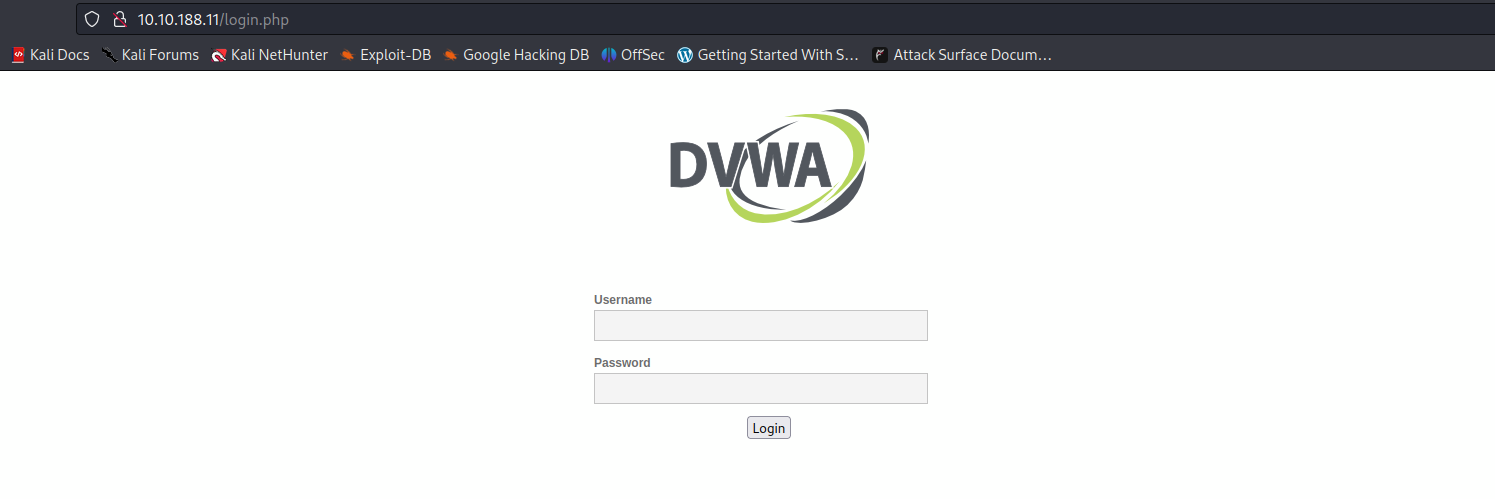
**No table of contents entries found.**

## 

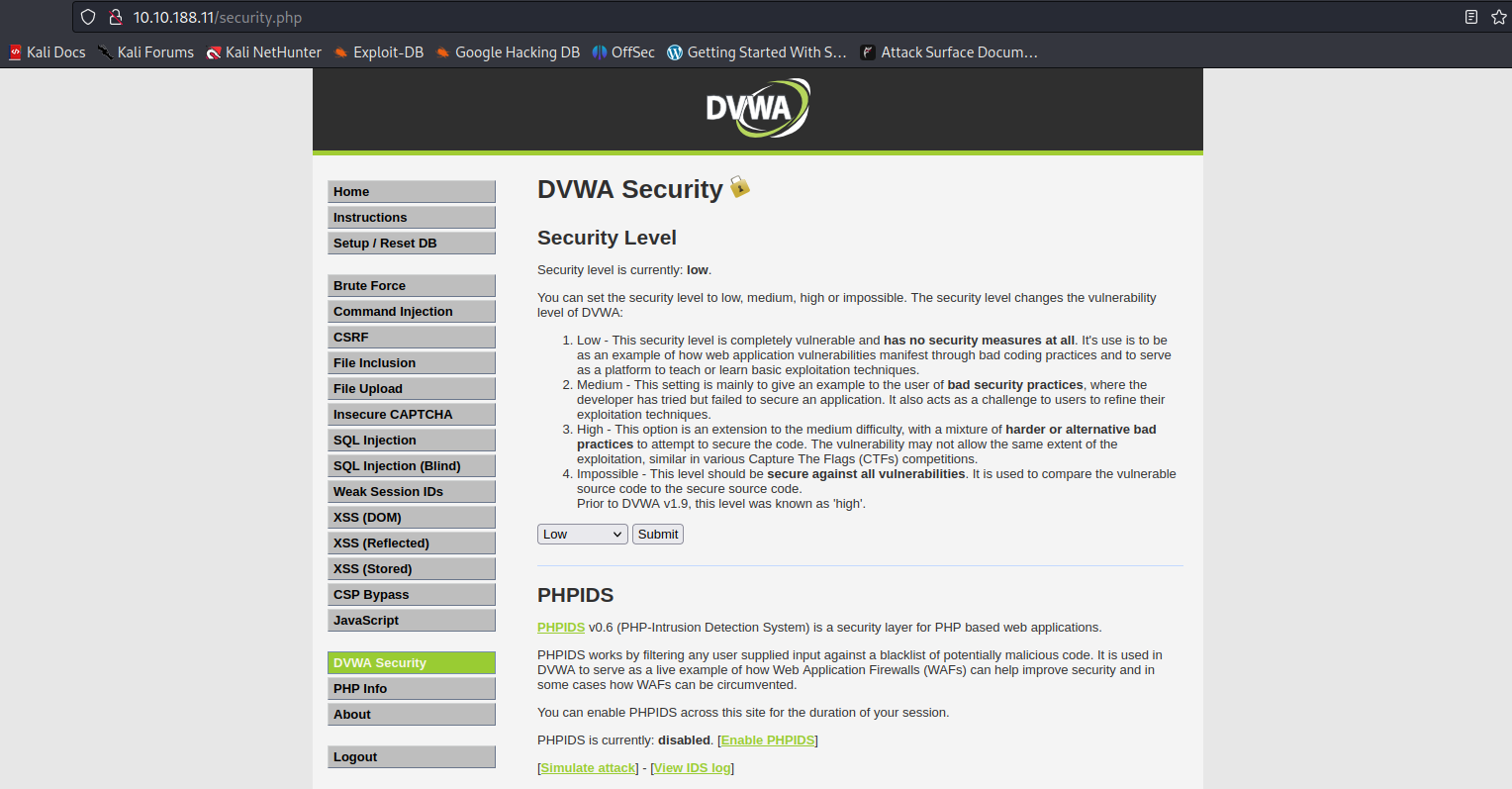
## 1. Brute force

Atak brute to metoda prób i błędów, która polega na wypróbowaniu wszystkich możliwych kombinacji, aby znaleźć właściwe dane, takie jak hasło czy klucz szyfrujący. Jest to jedna z najstarszych i najprostszych technik ataków na bezpieczeństwo, która jest nadal stosowana przez atakujących w celu przełamania zabezpieczeń.

Podstawową ideą ataku brute force jest powtarzające się wypróbowanie wszystkich możliwych kombinacji, aż do znalezienia właściwego rozwiązania. W przypadku ataku na hasło, atakujący próbuje wszystkich możliwych kombinacji znaków (liter, cyfr, znaków specjalnych) w celu odgadnięcia hasła dostępu do konta.



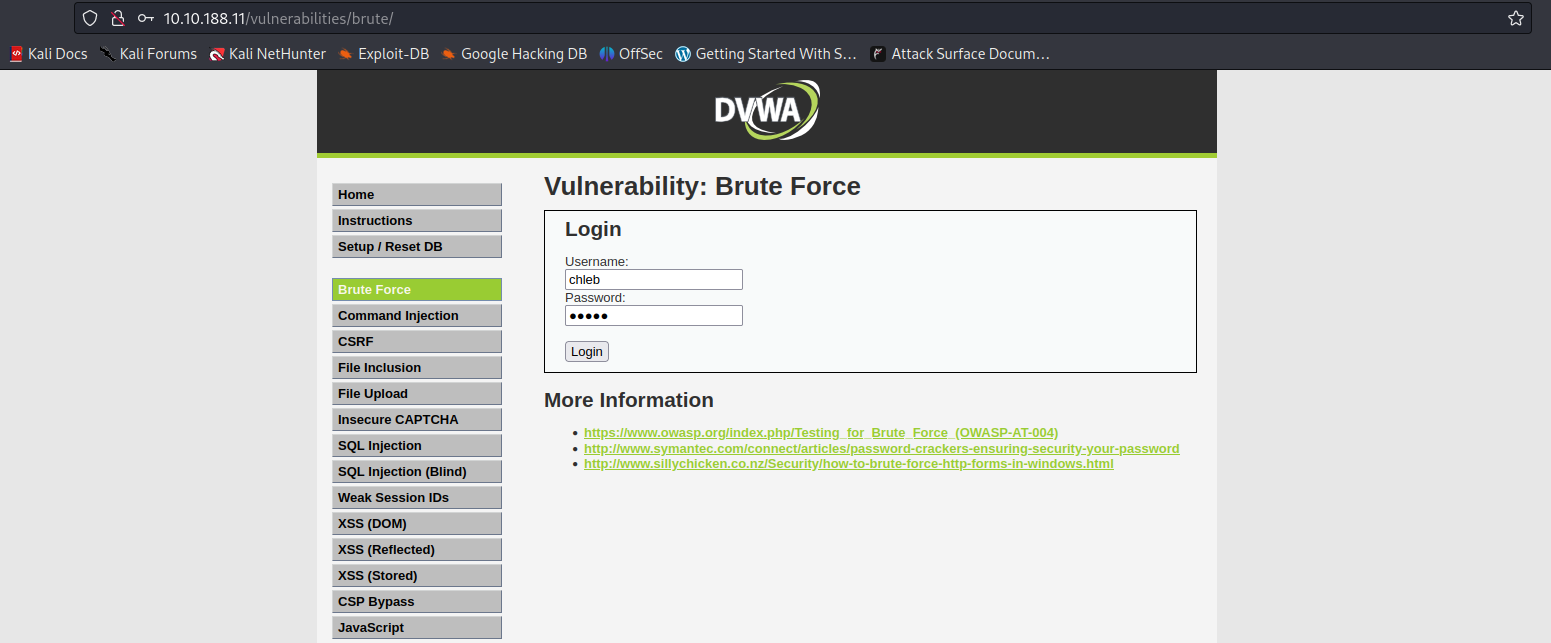
W celu realizacji tego testu trzeba było obniżyć poziom zabezpieczeń strony, na poziom low



*Rys. Zmiana poziomu zabezpieczeń aplikacji DVWA*

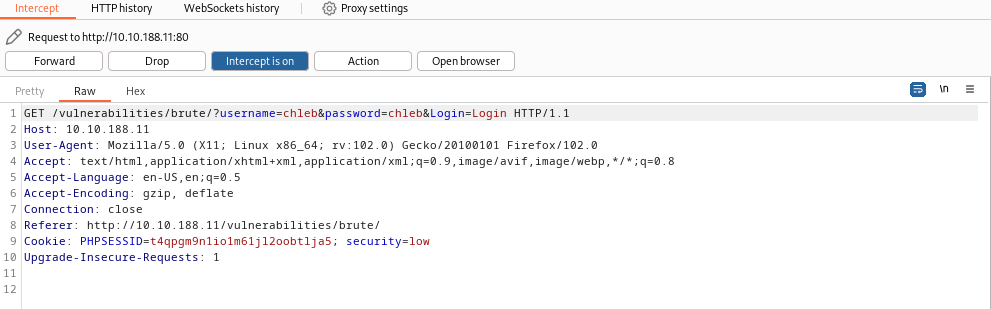
Test realizuję w zakładce Vulnerability: Brute Force

zaczynam od podania dowolnych danych dostępowych w formularzu w celu wysłania wypełnionego zapytania



*Rys. Formularz logowania*

Po włączeniu interceptora w Burp Suite wszelkie żądania wysyłane z naszej przeglądarki będą zatrzymywane przez serwer proxy. Następnie możemy sprawdzić, zmodyfikować, porzucić lub przekazać dalej żądanie.



*Rys. Przechwycone zapytanie przy próbie logowania*

Zapytanie zostało przechwycone w celu poznania jego budowy oraz uzyskania identyfikatora pliku cookie, który będzie potrzebny używając narzędzia Hydra.

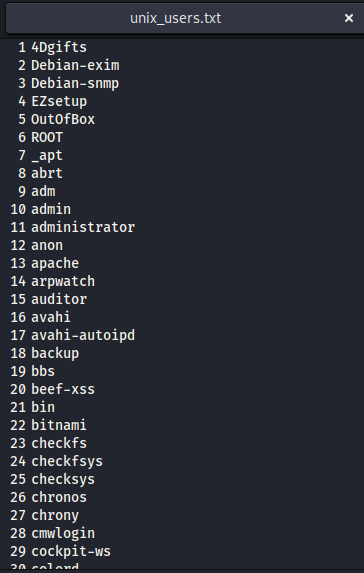
Hydra to narzędzie do brute-forcingu, które pomaga testerom penetracyjnym i etycznym hakerom łamać hasła do usług sieciowych.

Hydra może przeprowadzać szybkie ataki słownikowe na ponad 50 protokołów. Obejmuje to telnet, FTP, HTTP, HTTPS, SMB, bazy danych i kilka innych usług.

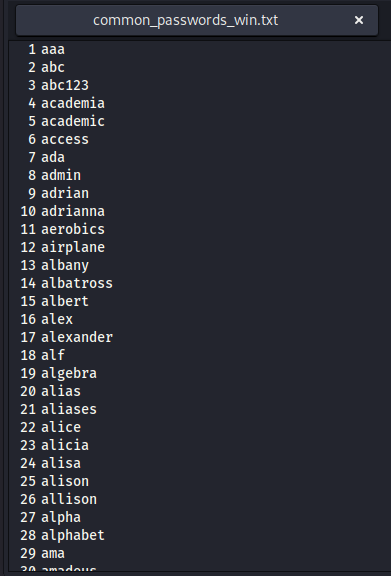
Z wykorzystaniem tego narzędzia zostanie wykonany atak słownikowy w celu uzyskania informacji na temat istniejących w aplikacji użytkowników i ich haseł.

W polecenieniu używamy pliku cookie, ponieważ jeśli nie jesteśmy uwierzytelnieni podczas próby logowania, zostaniemy przekierowani na domyślną stronę logowania.

Jako parametr polecenia podaję najpierw po fladze -L listę przykładowych loginów w pliku “unix\_users.txt”, a następnie po fladze -P podaję listę przykładowych haseł znajdujących się w pliku “common\_passwords\_win.txt”.



*Rys. Fragment listy przykładowych nazw użytkowników użytej do ataku słownikowego*



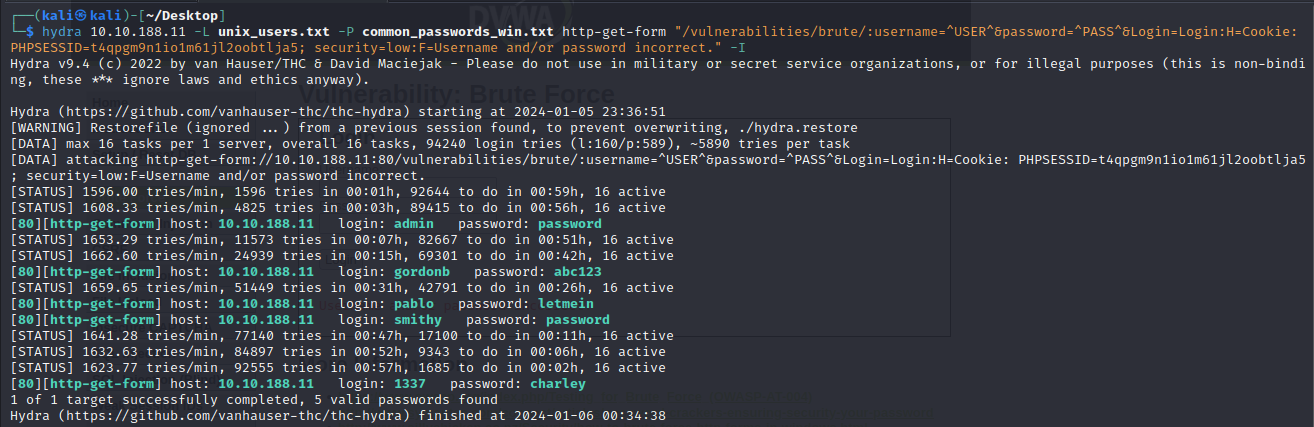
*Rys. Fragment listy przykładowych haseł użytej do ataku słownikowego*

Komunikat o niepowodzeniu to odpowiedź, którą otrzymujemy z formularza logowania po przesłaniu nieprawidłowego loginu. Jest to po prostu ciąg znaków, który Hydra przeszukuje w odpowiedzi HTML, aby sprawdzić, czy logowanie się powiodło, czy nie. Na przykład komunikat, który otrzymujemy na czerwono pod formularzem logowania po złej próbie logowania, brzmi: “Username and/or password incorrect.”.

Polecenie realizujące te zadanie wygląda następująco:

|  |
| --- |
| hydra 10.10.188.11 -L unix\_users.txt -P common\_passwords\_win.txt http-get-form "/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:H=Cookie: PHPSESSID=t4qpgm9n1io1m61jl2oobtlja5; security=low:F=Username and/or password incorrect." -I |

Po wypróbowaniu wszystkich możliwość Hydra zwróciła następujące wyniki:



*Rys. Uzyskane wyniki z wykorzystaniem narzędzia Hydra*

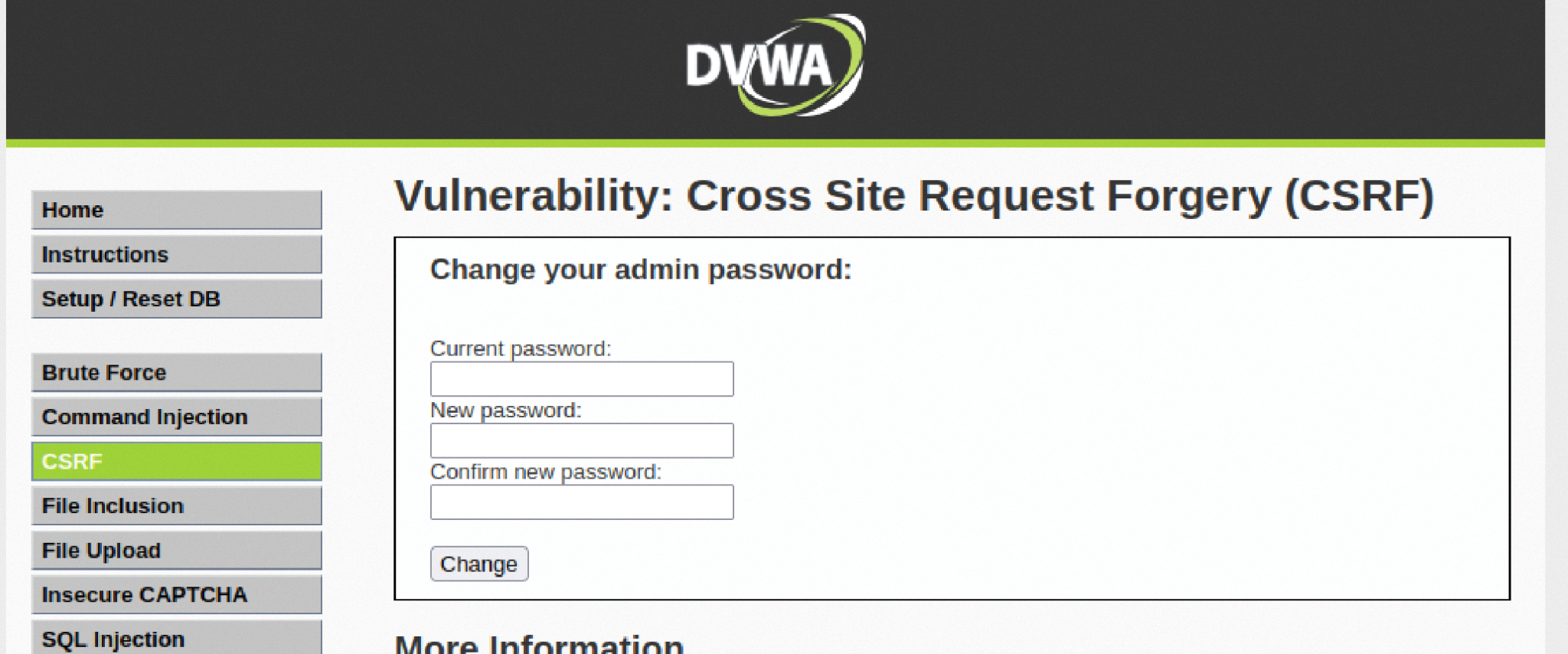
Udało się ustalić dane pięciu użytkowników oraz ich hasła.

## 2. Cross-Site Request Forgery

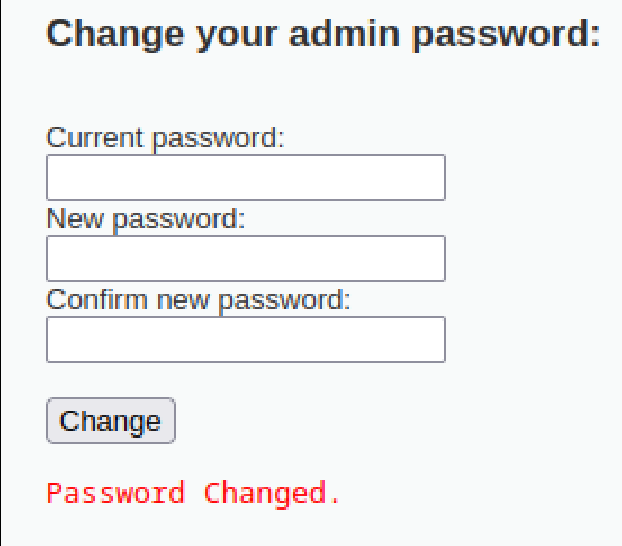
Cross-Site Request Forgery (CSRF) to rodzaj ataku na bezpieczeństwo aplikacji internetowych, który polega na wymuszaniu niechcianych działań w imieniu zalogowanego użytkownika poprzez wykonanie fałszywych żądań HTTP do aplikacji, do której użytkownik jest zalogowany. Atakujący wykorzystuje to, że aplikacja internetowa wykonuje żądania HTTP, nie uwzględniając kontekstu, co pozwala atakującemu na podsunięcie żądania w miejsce zalogowanego użytkownika.

Atak CSRF występuje, gdy aplikacja internetowa nieprawidłowo zabezpiecza żądania HTTP, które wykonują działania o znaczeniu zmieniającym stan, takie jak zmiana hasła, wysłanie wiadomości e-mail, dokonanie płatności itp. Atakujący może wykorzystać to, aby zmusić zalogowanego użytkownika do wykonania niechcianych działań poprzez przekierowanie go na specjalnie spreparowaną stronę zawierającą złośliwe żądanie HTTP.

Po zalogowaniu się na maszynę, należy ustawić bezpieczeństwo na Low, aby atak się udał. Następnie należy wejść w zakładkę CSRF.

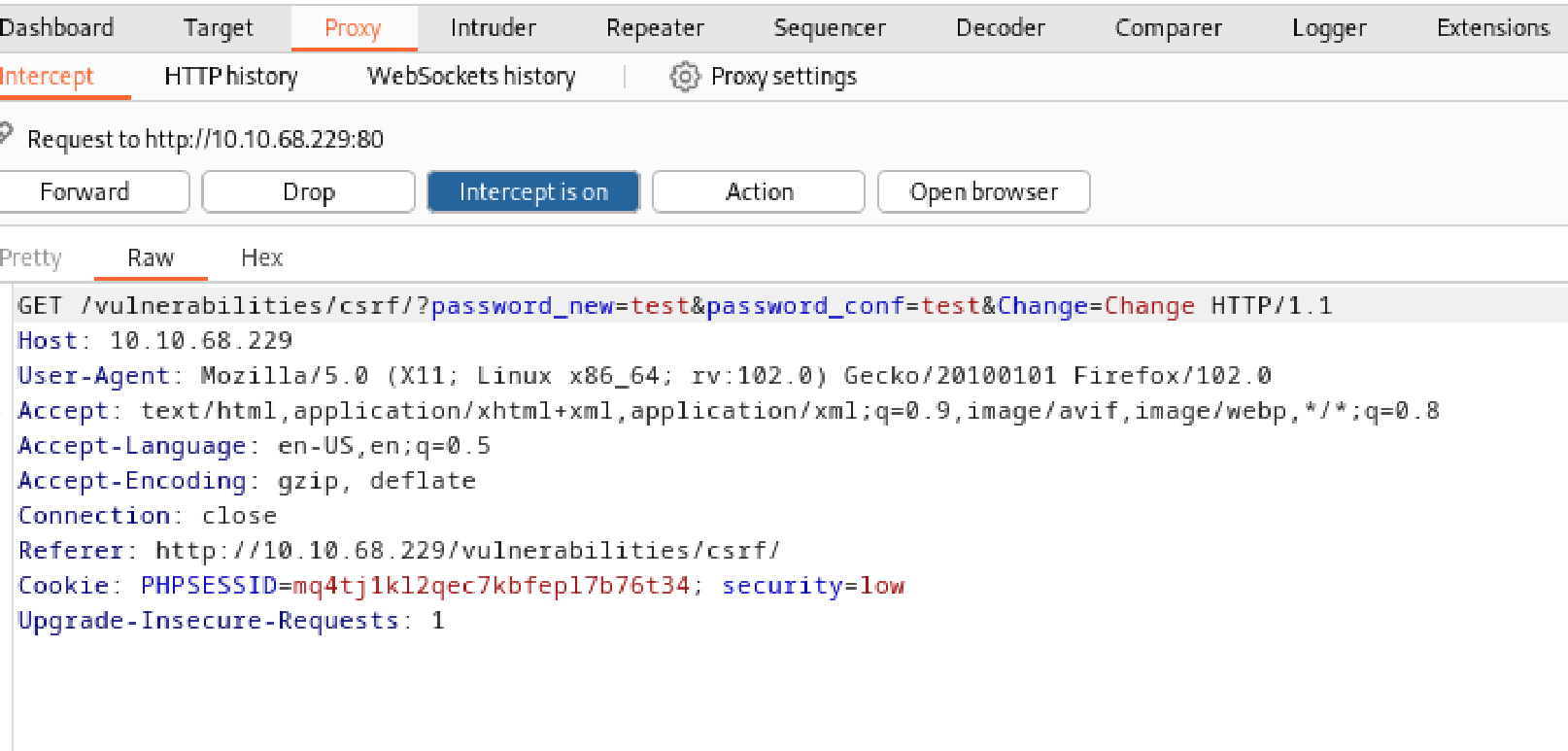


*Rys. Maszyna DVWA dla CSRF*



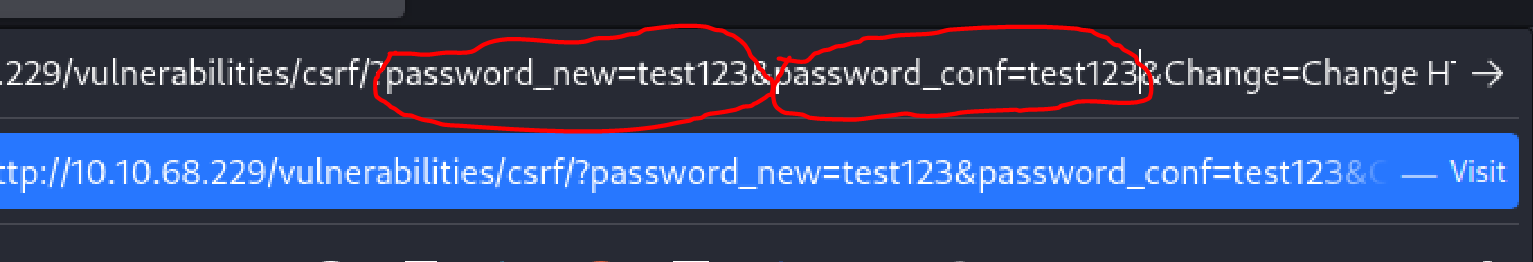
*Rys. Poprawne działanie bez żadnego ataku*

Na początku należy przechwycić za pomocą narzędzia Burp Suite, dostęp do strony podczas próby zmiany hasła. Dzięki temu otrzymamy wartość GET wysłanego Requesta.



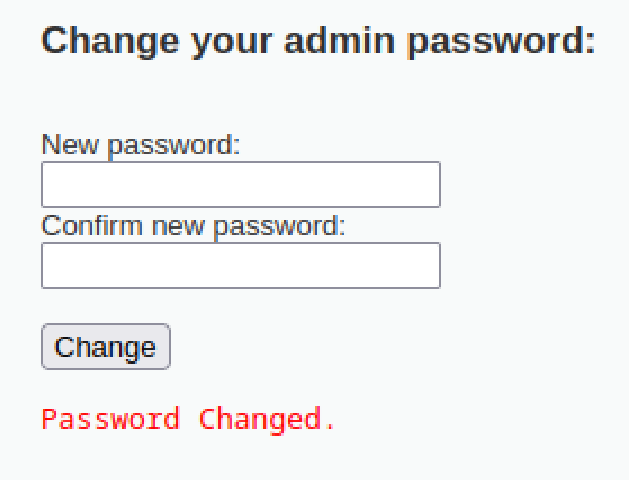
*Rys. Wartość parametru GET*

Otrzymaną wartość GET wklejono w miejsce oryginalnego adresu źródłowego strony. Za pomocą utworzonego linku można tworzyć nowe hasła dla użytkownika danego konta, zmieniając wartość “password\_new” i “password\_conf”, na nowe oczekiwane hasło.



*Rys. Zmiana hasła poprzez użycie odpowiednio zmodyfikowanego linku*

Po zatwierdzeniu linka, otrzymano komentarz zwrotny od maszyny DVWA, który informuje o powodzeniu zmiany hasła.



*Rys. Potwierdzenie zmiany hasła*

## 3. Cross-Site-Scripting

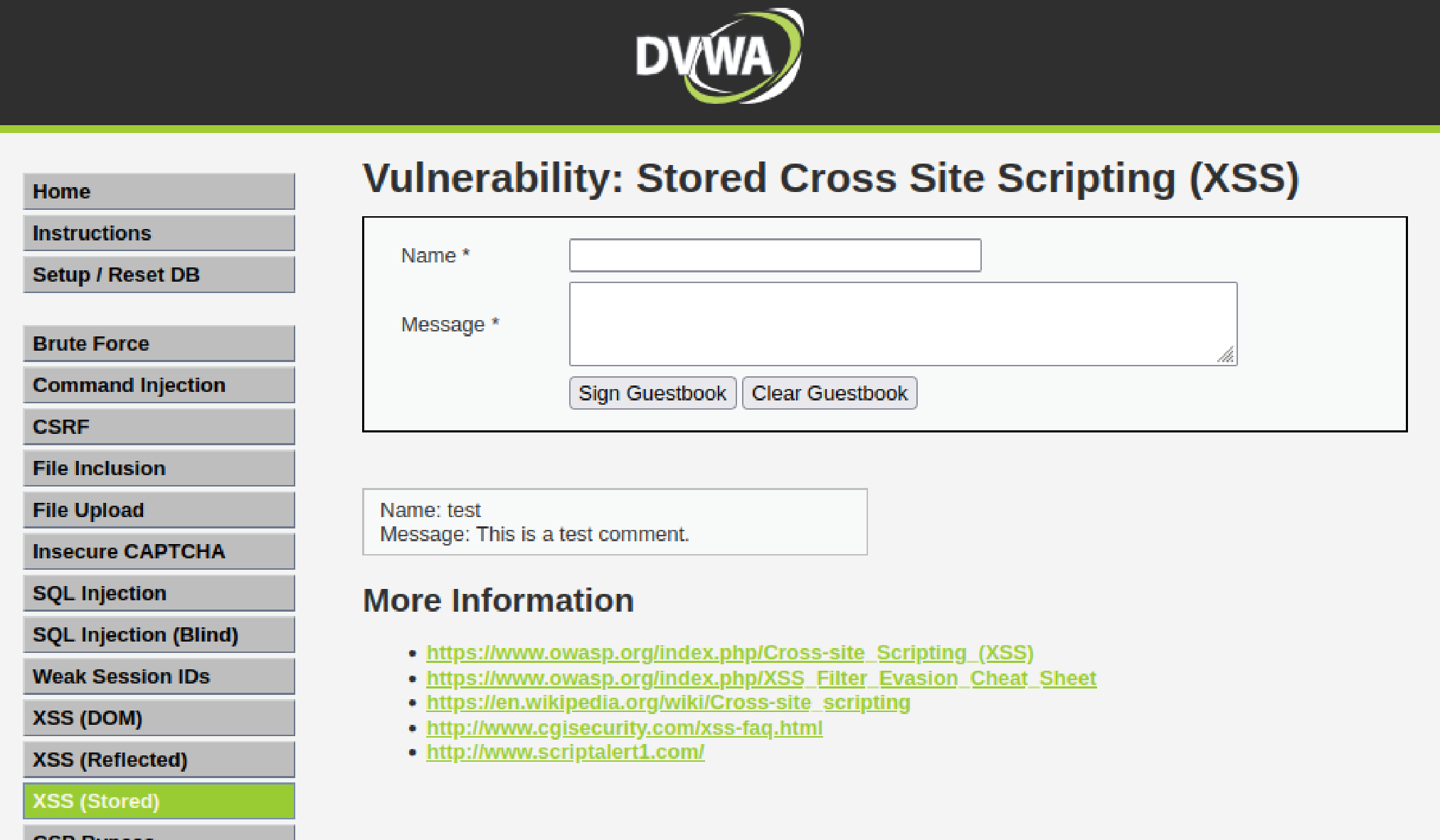
Cross-Site Scripting (XSS) to rodzaj ataku na bezpieczeństwo aplikacji internetowych, który polega na wstrzyknięciu złośliwego kodu JavaScript do stron internetowych odwiedzanych przez inne użytkowników. Atakujący wykorzystuje tę podatność, aby osadzić złośliwy skrypt w stronie internetowej, który zostanie wykonany w przeglądarce użytkownika, który odwiedzi tę stronę.

Atak XSS występuje, gdy aplikacja internetowa nieprawidłowo obsługuje dane wprowadzane przez użytkownika i umożliwia wstrzyknięcie złośliwego kodu JavaScript do treści strony internetowej. Atakujący może wykorzystać to, aby wyświetlić fałszywe komunikaty, przekierować użytkownika na złośliwe strony internetowe, ukraść sesje użytkowników lub przechwycić dane osobowe.

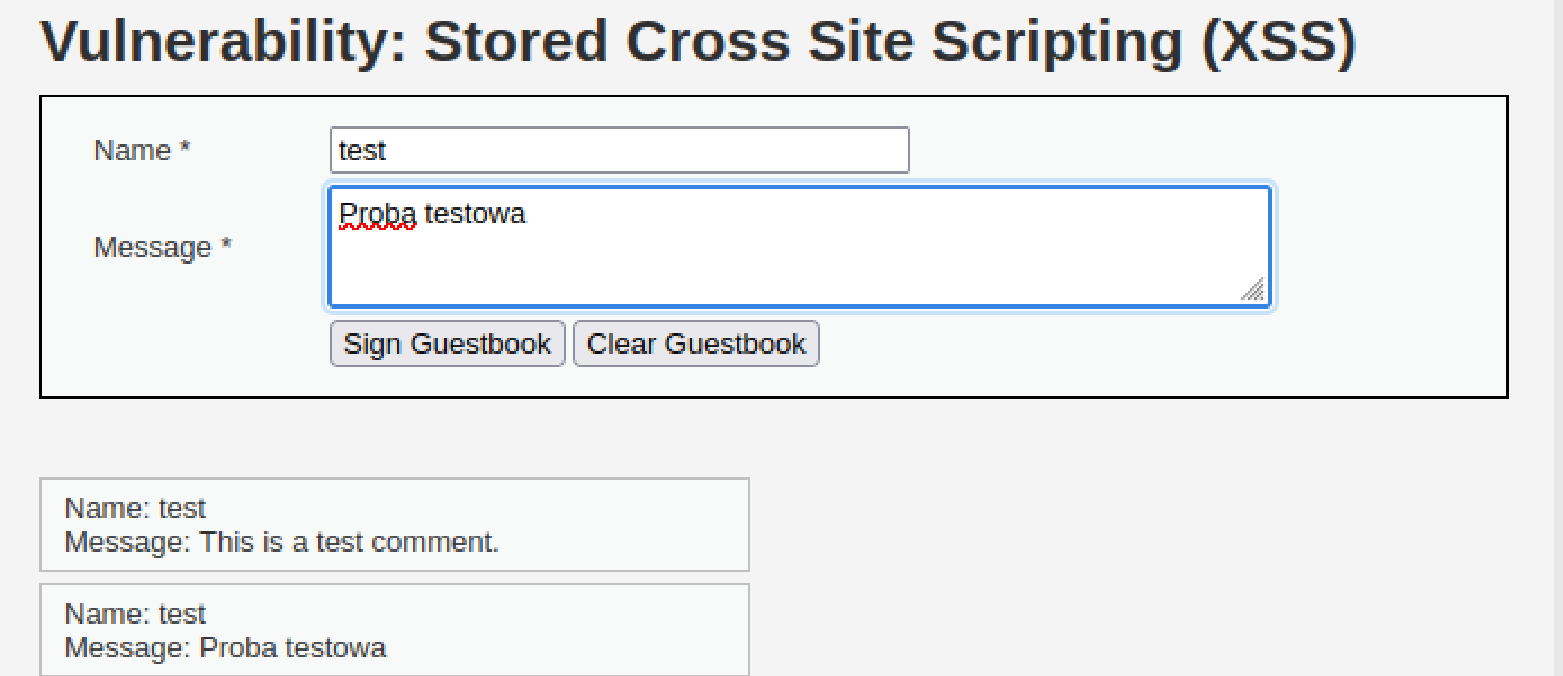
Istnieją trzy główne rodzaje ataków XSS:

* Stored XSS (Persisted XSS): W tym przypadku atakujący wstrzykuje złośliwy kod JavaScript do bazy danych lub plików na serwerze, które są później wyświetlane wszystkim odwiedzającym daną stronę.
* Reflected XSS: W ataku tego typu, złośliwy kod JavaScript jest wstrzykiwany w parametry URL lub formularzy, a następnie wyświetlany w odpowiedzi serwera. Użytkownik musi kliknąć na specjalnie spreparowany link, aby atak został wykonany.
* DOM-based XSS: Ten rodzaj ataku XSS występuje, gdy atakujący manipuluje drzewem DOM (Document Object Model) w przeglądarce użytkownika, wykorzystując nieprawidłowe posługiwanie danych wprowadzanych przez użytkownika przez skrypty JavaScript na stronie.

Po zalogowaniu się na maszynę, należy ustawić bezpieczeństwo na Low, aby atak się udał. Następnie należy wejść w zakładkę XSS (Stored).

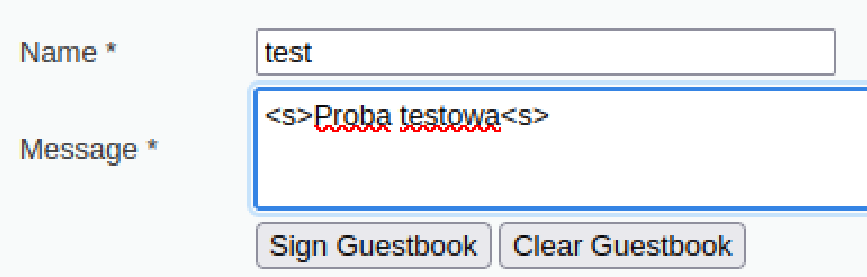


*Rys. Maszyna DVWA*



*Rys. Poprawne działanie strony*

Aby wykonać atak XSS należy w polu Message dodać odpowiednie tagi HTML. Aby uzyskać przekreśloną wiadomość, należy wprowadzić flagi “<s></s>”.

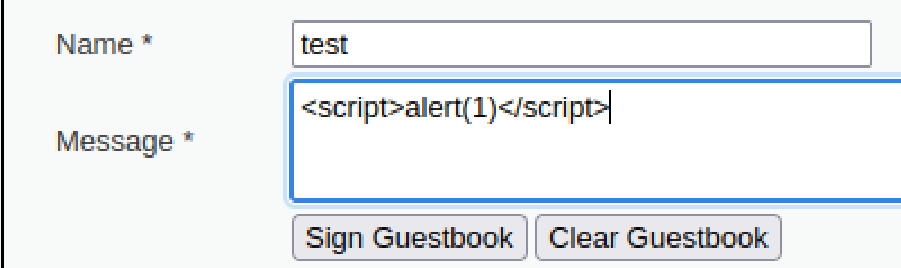


*Rys. Wprowadzona wiadomość*



*Rys. Skreślona wiadomość tagu*

Istnieje możliwość wprowadzenia własnego skryptu JavaScript.

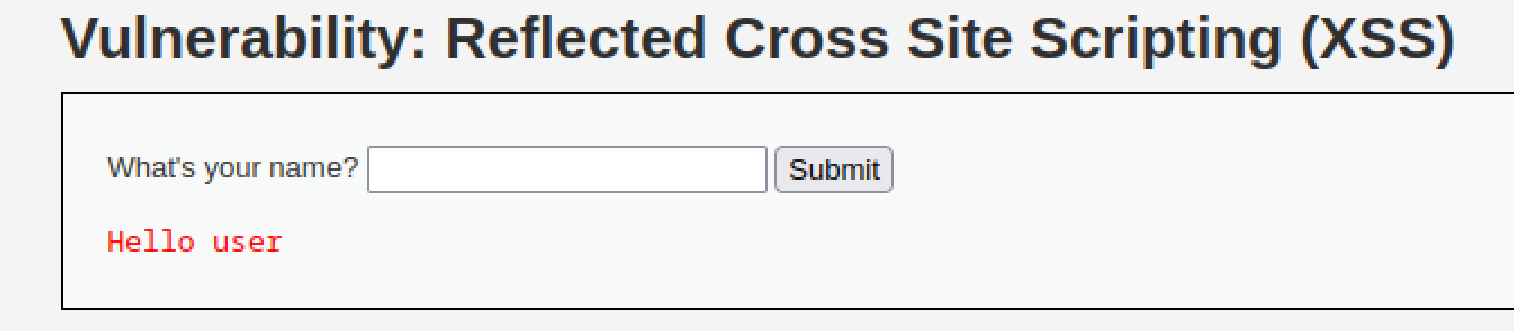


*Rys. Wprowadzenie skryptu*



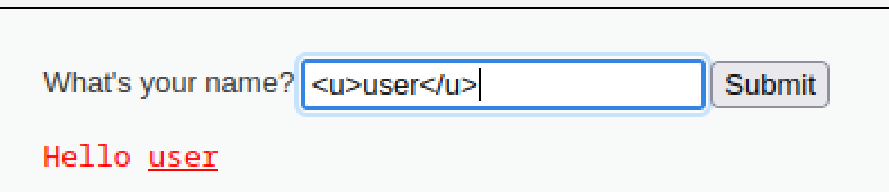
*Rys. Otrzymany alert*

Następnie należy przejść do zakładki XSS (Reflected). Na początek pokazano jak zachowa się strona bez użycia żadnych skryptów.



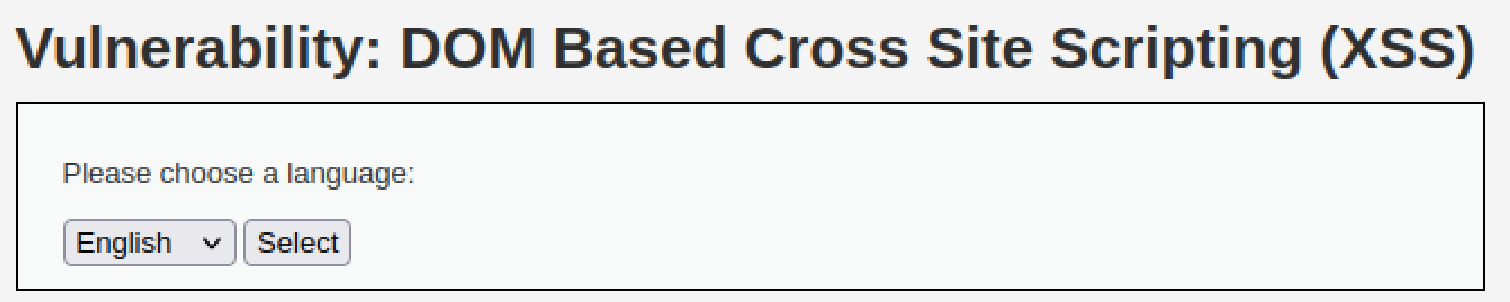
*Rys. Poprawne zachowanie strony na komendę “user”*

Następnie należy sprawdzić reakcję strony poprzez użycie tagu HTML, który powinien spowodować podkreślenie wyniku.



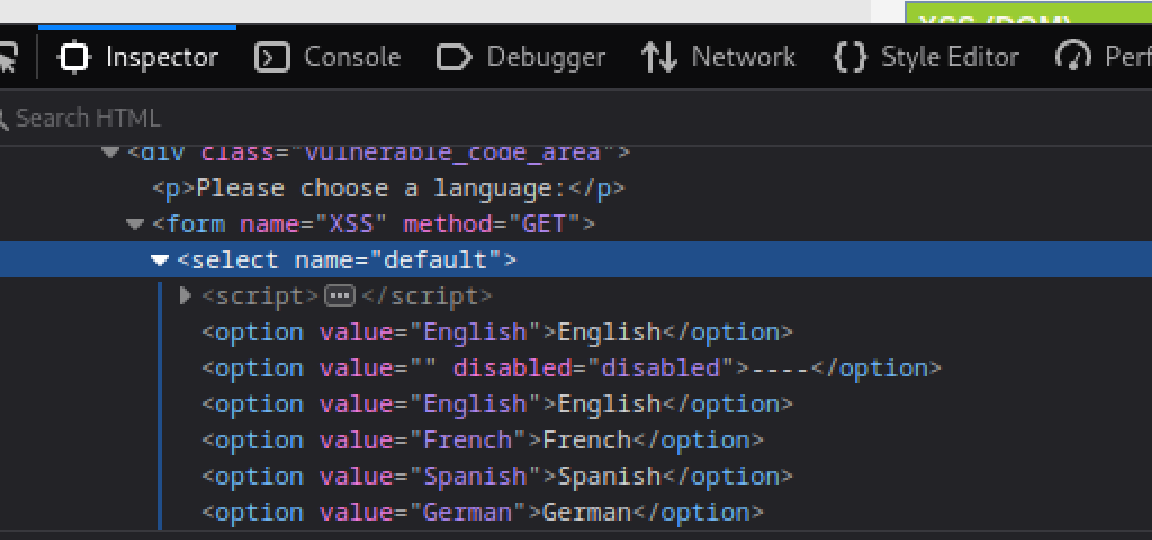
*Rys. Pokazanie udanego ataku XSS (Reflected)*

Kolejnym atakiem XSS jest atak XSS (DOOM).



*Rys. Poprawne działanie strony*

Żeby móc zaatakować stronę trzeba wejść w komunię i odnaleźć sekcję odpowiedzialną za kafelek, który zmienia języki. W miejscu gdzie jest disable należy wymienić disable na skrypt, który ma zaatakować wspomniany kafelek, oraz można dopisać alert jaki będzie się pokazywać, po uruchomieniu wspomnianego kafelka.



*Rys. Znalezieni w konsoli miejsca, gdzie można dopisać własny kafelek*



*Rys. Użyty skrypt do napisania własnego kafelka*



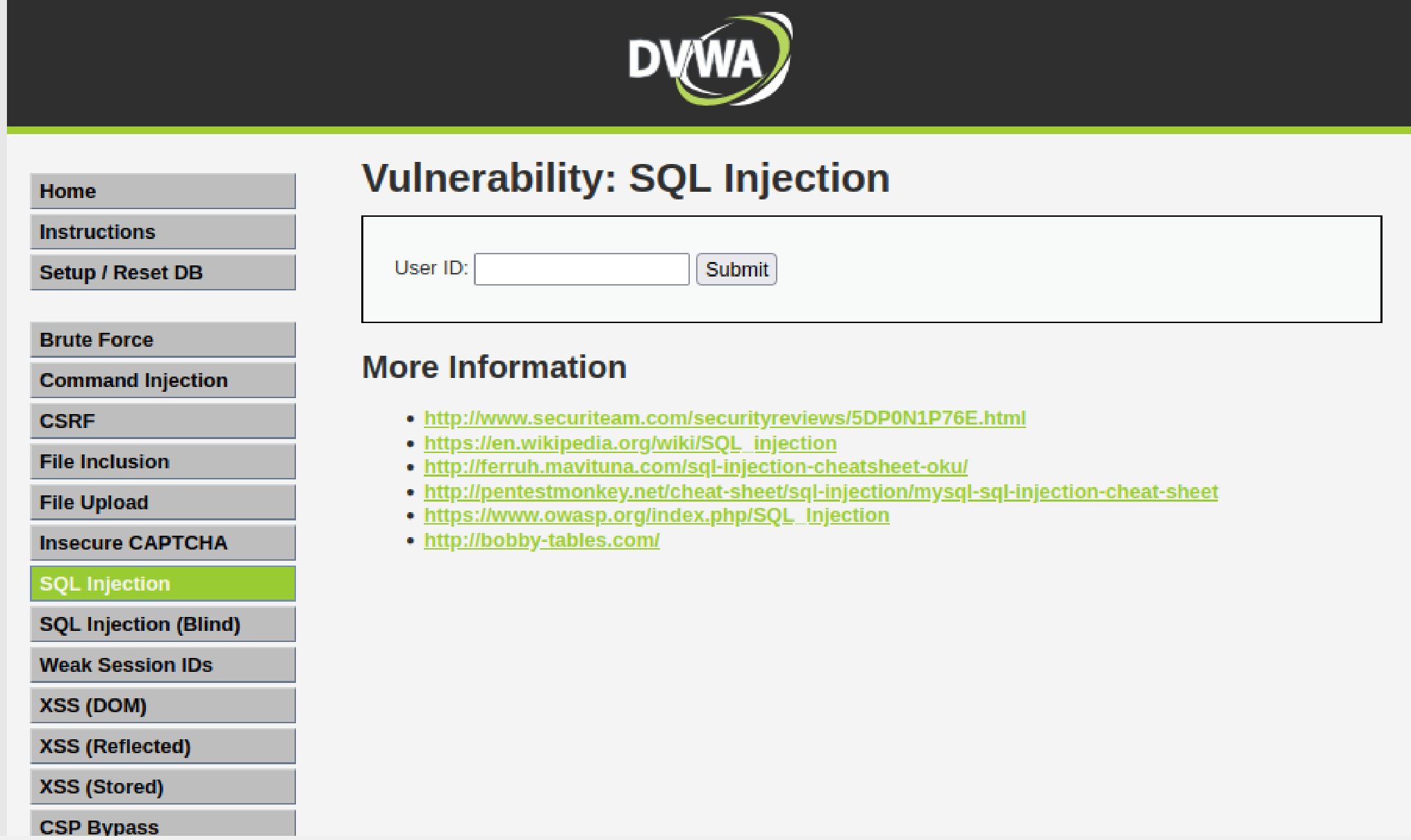
*Rys. Uruchomienie stworzonego kafelka, wraz z stworzonym alertem w skrypcie*

## 4. SQL Injection

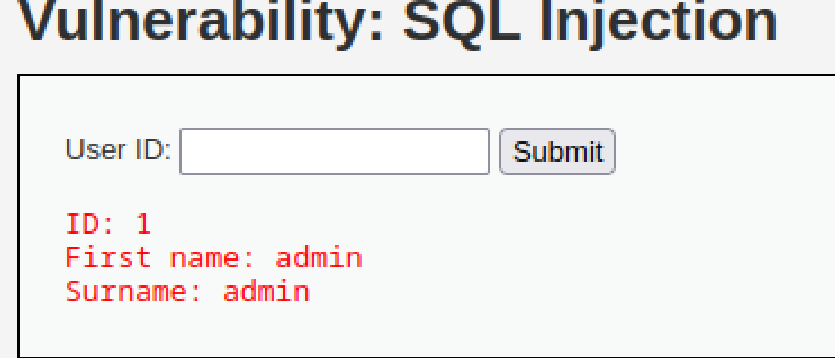
SQL Injection to rodzaj ataku na bezpieczeństwo aplikacji internetowych, który polega na wstrzyknięciu złośliwego kodu SQL do zapytań SQL wykonywanych przez aplikację. Atakujący wykorzystuje tę podatność, aby uzyskać dostęp do danych w bazie danych, modyfikować dane, omijać mechanizmy uwierzytelniania, a nawet przejąć kontrolę nad serwerem baz danych.

Atak SQL Injection występuje, gdy aplikacja internetowa nieprawidłowo waliduje dane wprowadzane przez użytkownika i bezpośrednio wprowadza te dane do zapytań SQL bez odpowiedniego zabezpieczenia. Atakujący może wykorzystać to, wprowadzając specjalnie spreparowane dane wejściowe, które modyfikują działanie zapytania SQL, co prowadzi do nieoczekiwanych efektów.

Po zalogowaniu się na maszynę, należy ustawić bezpieczeństwo na Low, aby atak się udał. Następnie należy wejść w zakładkę SQL Injection.

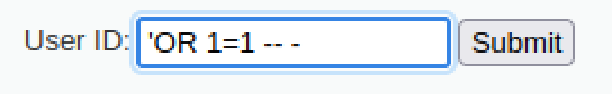


*Rys. Maszyna DVWA*

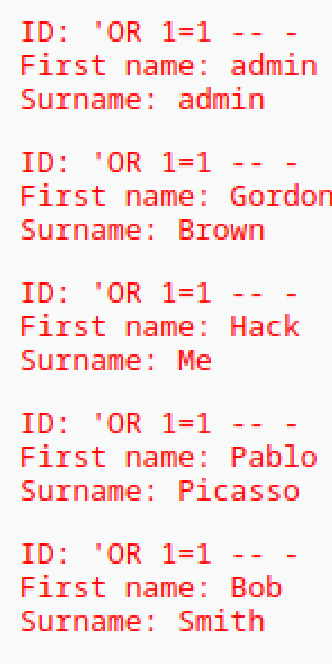


*Rys. Poprawne użycie*

Atak może nastąpić, poprzez złamanie logiki zapytania, aby zwróciło więcej niż jeden wynik. Aby uzyskać ten wynik należy wprowadzić komendę ‘OR 1=1 -- -.

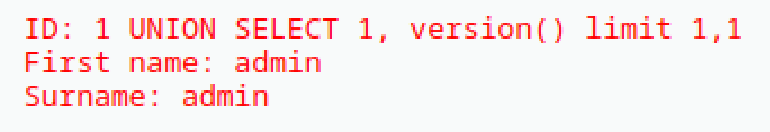


*Rys. Wprowadzenie skryptu*



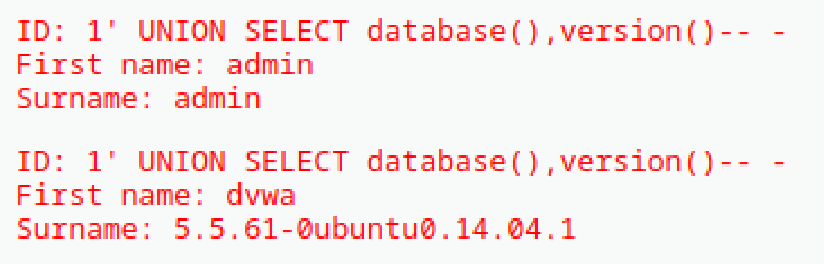
*Rys. Otrzymane wyniki*

Istnieje możliwość zwrócenia wyników ze względu na wybrany wcześniej ID. Aby to uzyskać należy użyć komendy: *1 UNION SELECT 1, version () limit 1,1*.

**

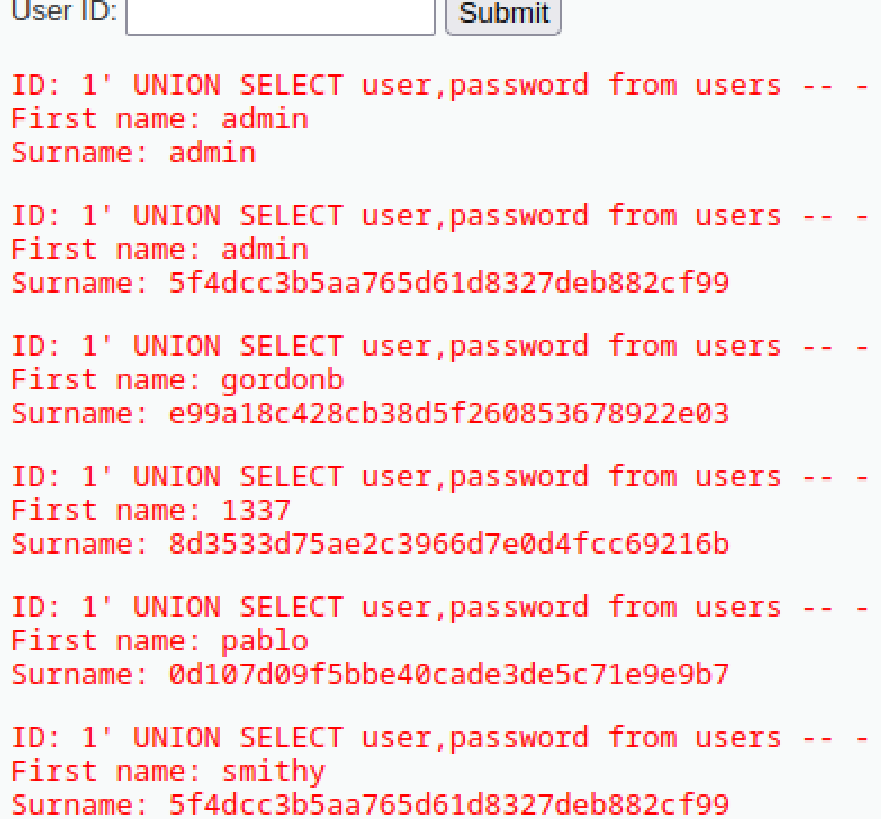
*Rys. Otrzymane wyniki*

Można uzyskać informacje na temat bazy danych. Aby to uzyskać, należy użyć komendy: *1' UNION SELECT database(),version()-- -.*



*Rys. Otrzymane wyniki*

Można zobaczyć hasła użytkowników poprzez użycie komendy: *1' UNION SELECT user,password from users -- -*



*Rys. Otrzymane wyniki*