

# DWVA : Force Brute

---

ATELIER U7 CYBERSECURITE

SPINELLI Dylan  
BTS SIO SISR | PARIS YNOV CAMPUS

## Table des matières

1- Introduction .....	2
2- Exercice Brute Force avec DVWA.....	3
3- Attaque avec Burp Suite.....	7

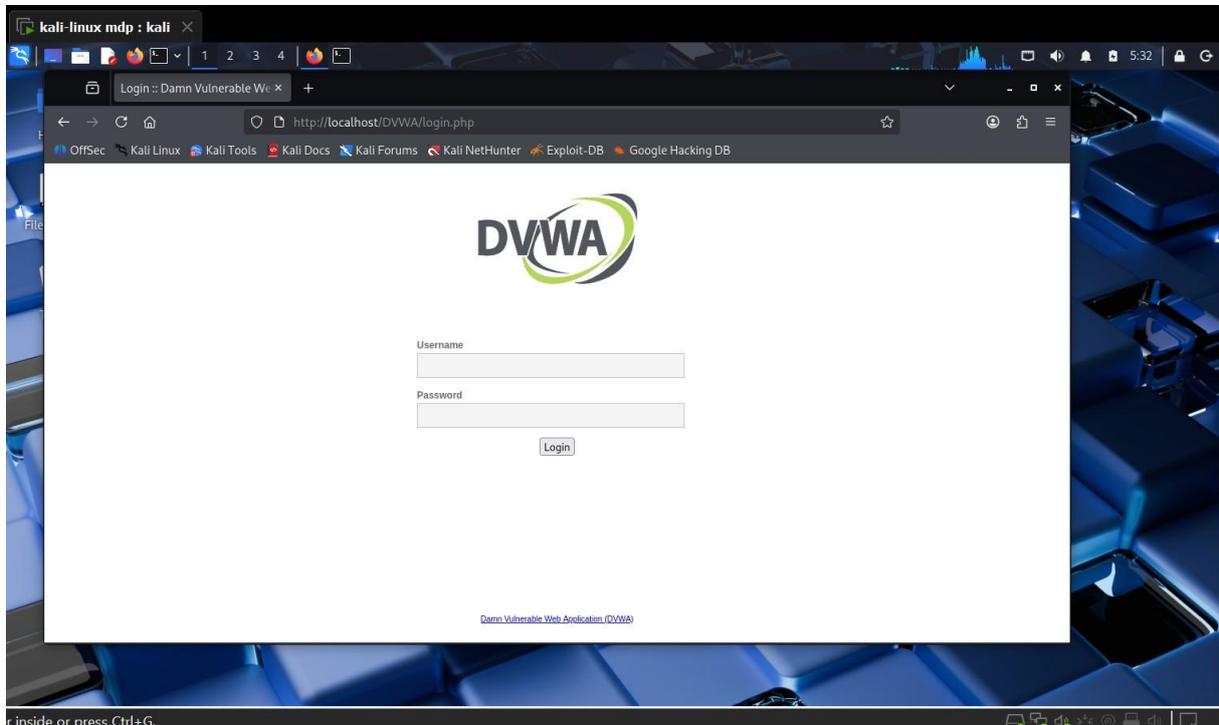
# 1-Introduction

Dans cet atelier, nous allons réaliser une attaque par Force Brute. L'attaque par force brute consiste à envoyer une série de requêtes http avec des informations de connexions différentes à chaque requête. L'objectif est le retrouver les informations de connexion.

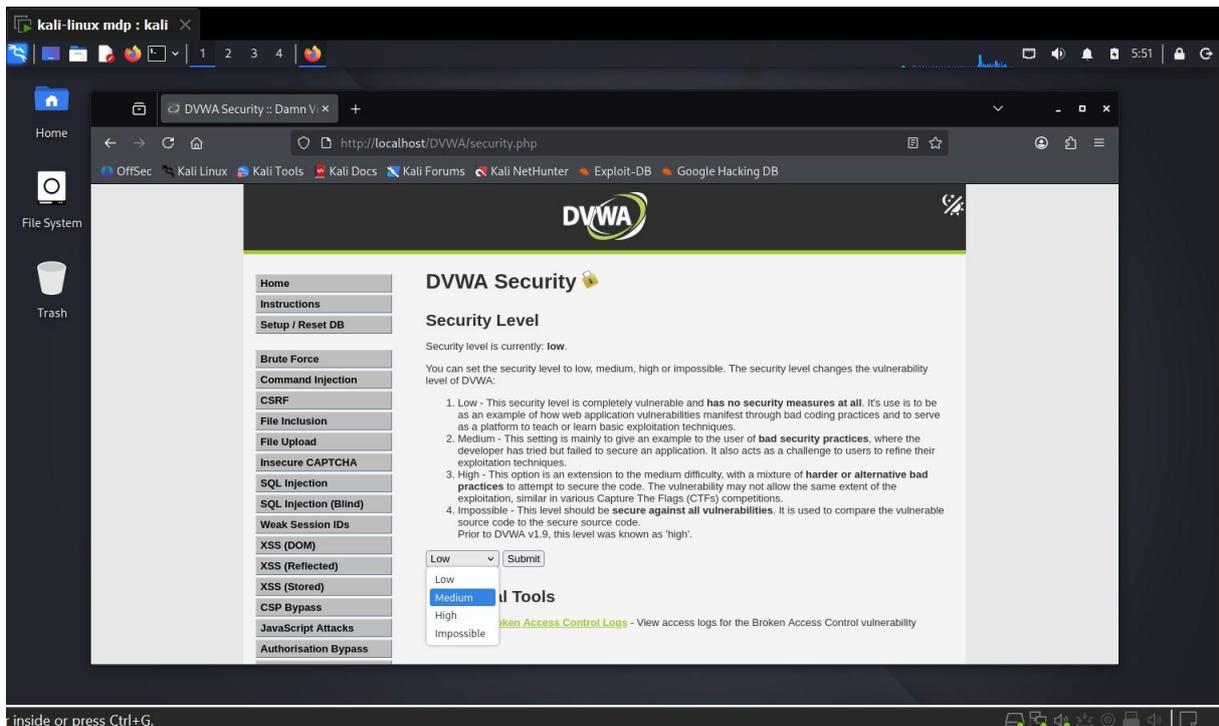
Nous allons utiliser le logiciel Burp Suite afin de recréer le scénario d'une attaque par Force Brute.

## 2-Exercice Brute Force avec DVWA

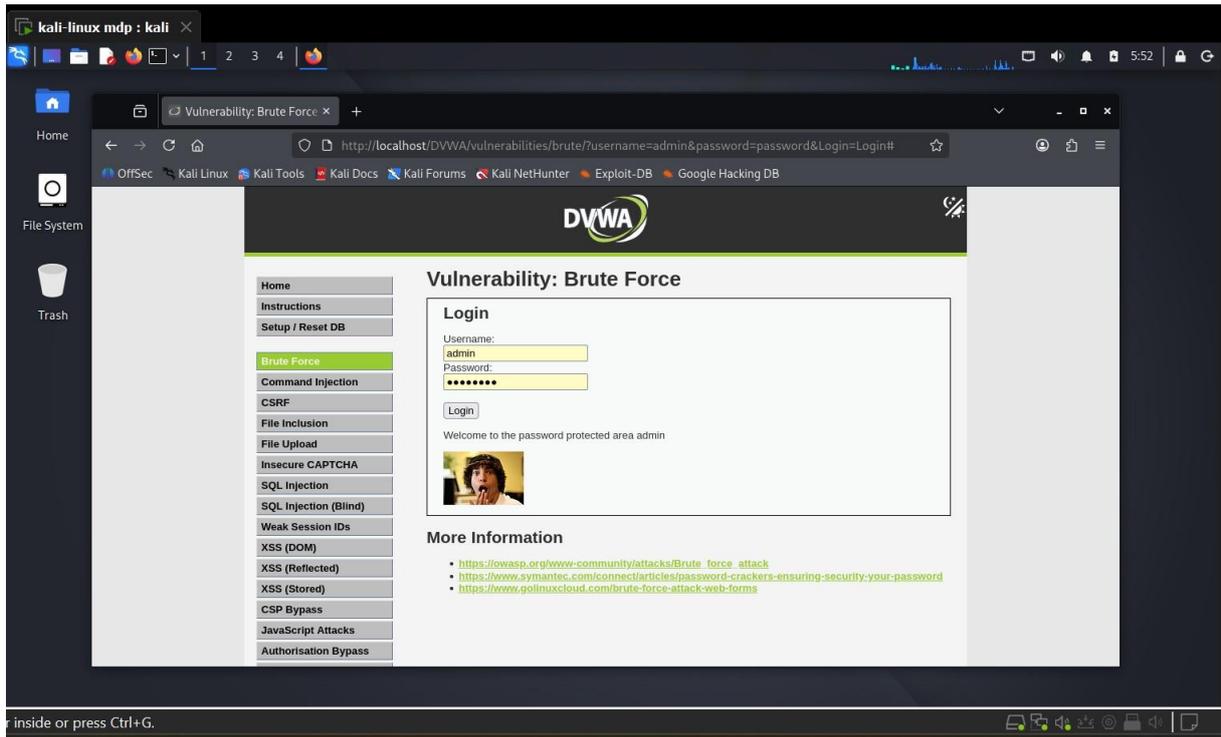
Connectons-nous d'abord sur DVWA :



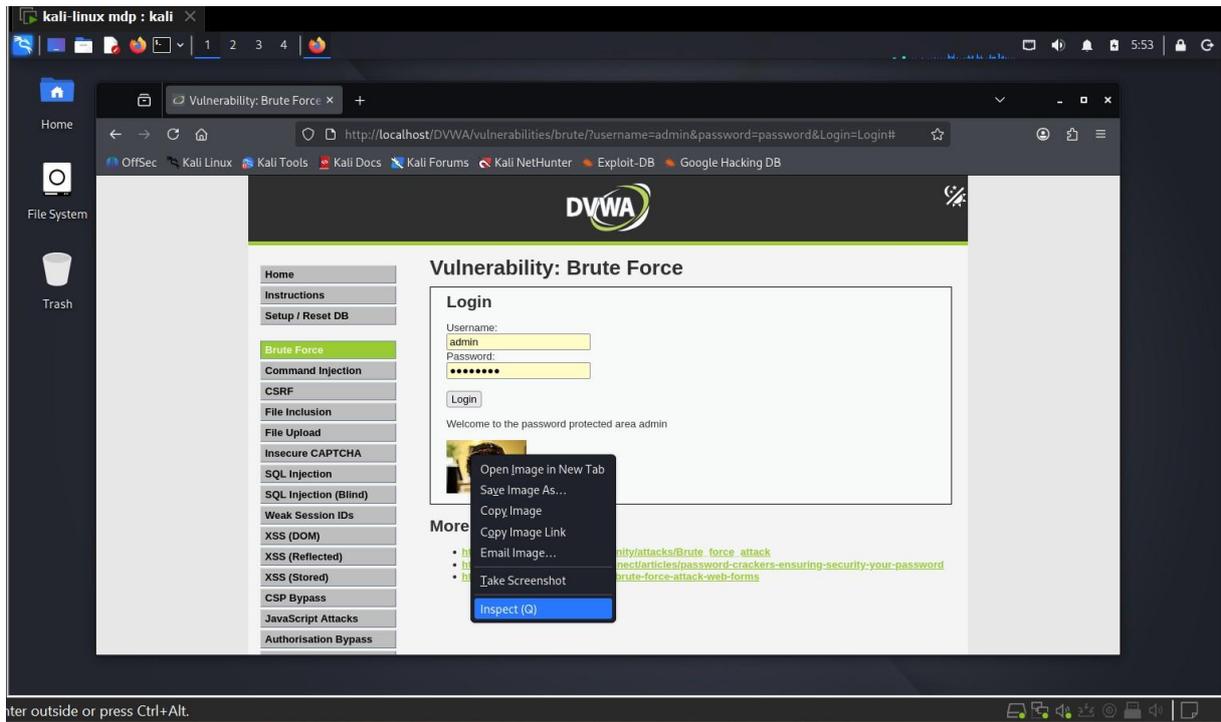
On choisit le niveau de difficulté medium :



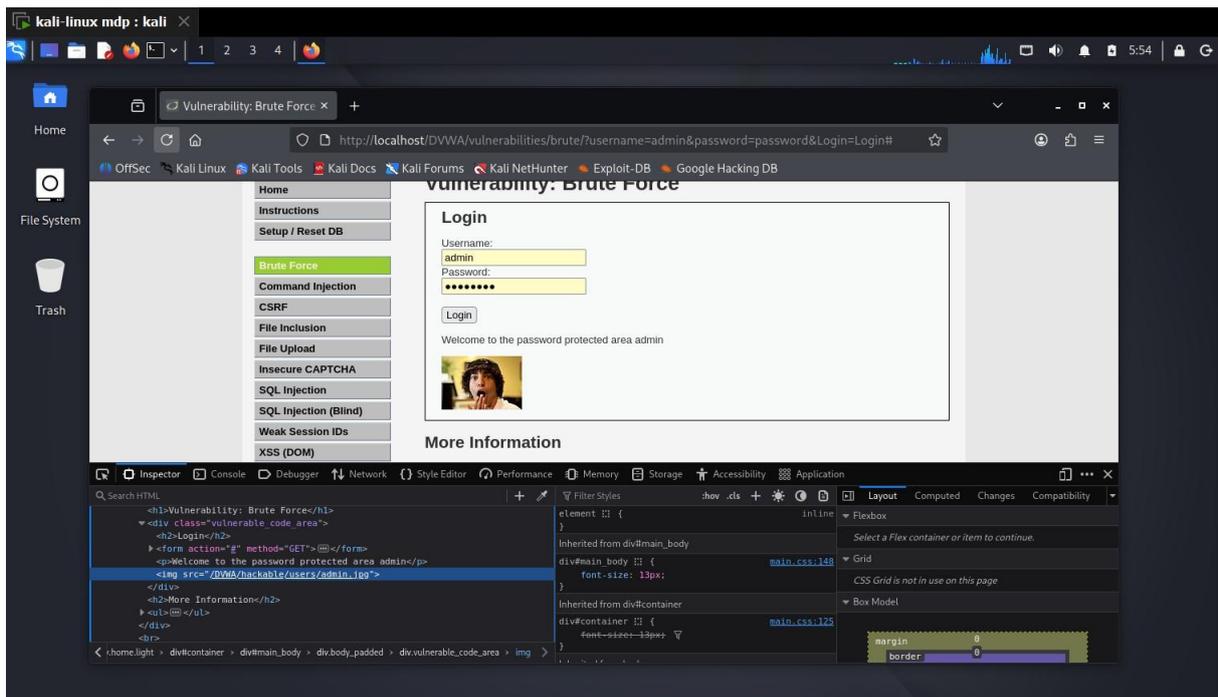
On sélectionne le type d'attaque « Brute force » et on renseigne nos informations de connexion :



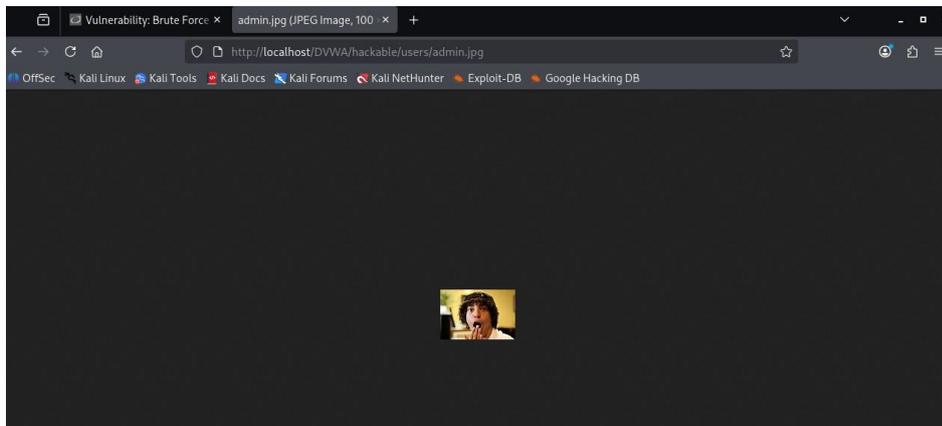
Une image apparaît suite à notre connexion. Examinons cette image :



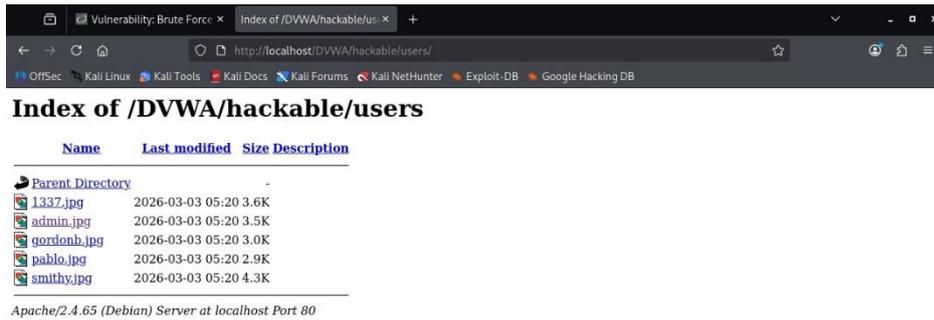
On trouve le chemin où est sauvegardée l'image sur le serveur :



Si nous accédons directement à ce chemin depuis le navigateur, on retrouve bien la même image :



Rendons-nous directement au chemin `http://local/hackable/users/` :



---

Nous avons désormais la liste de tous les utilisateurs enregistrés dans le système !

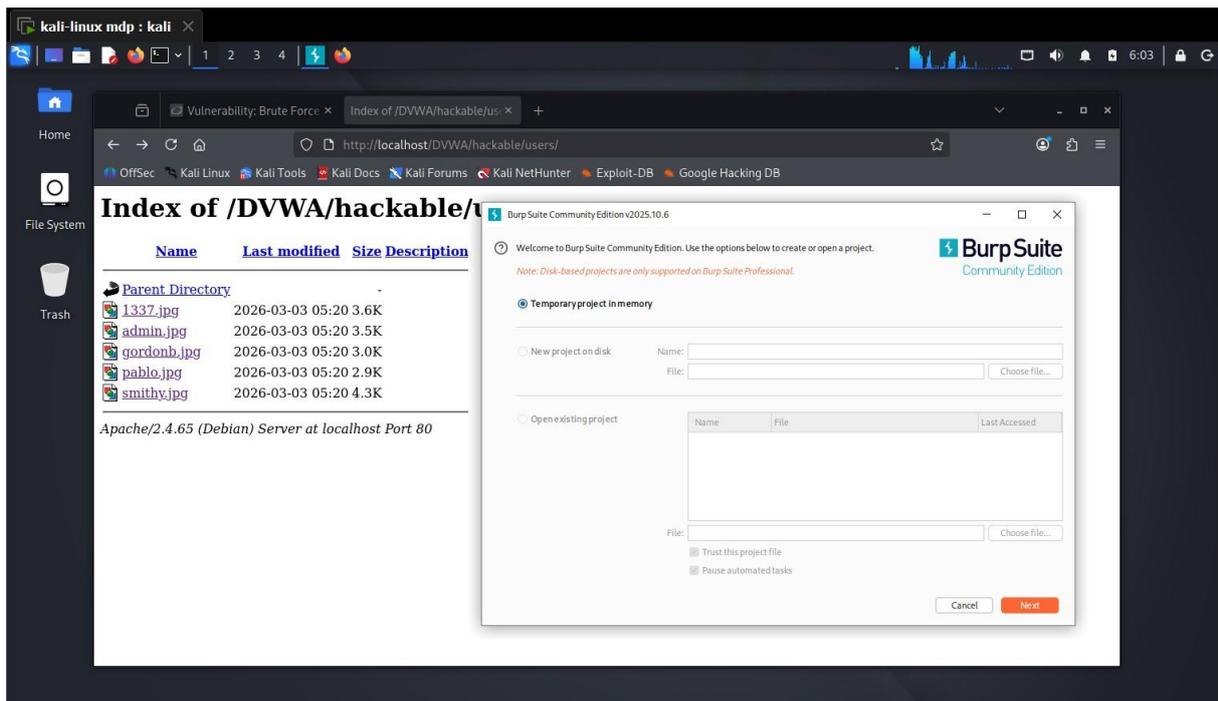
Il y a donc 5 utilisateurs.

On apprend également qu'il s'agit de la technologie Apache/2.4.65 Debian

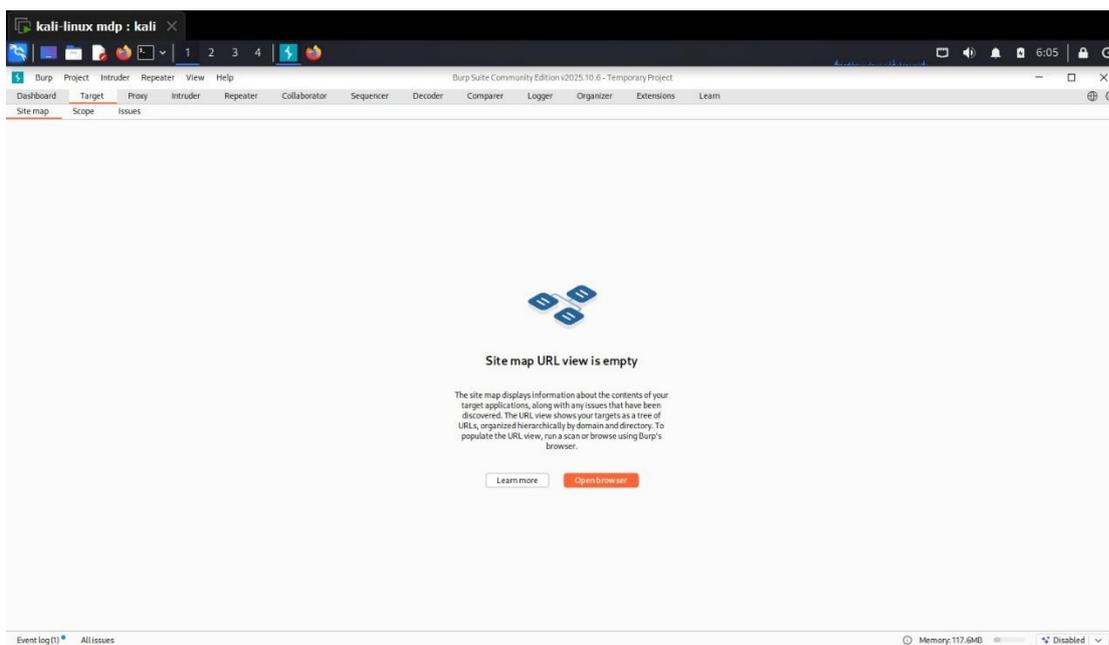
## 3-Attaque avec Burp Suite

Burp Suite est un proxy situé entre le navigateur et le serveur web. Il permet de filtrer le trafic web et d'envoyer des requêtes

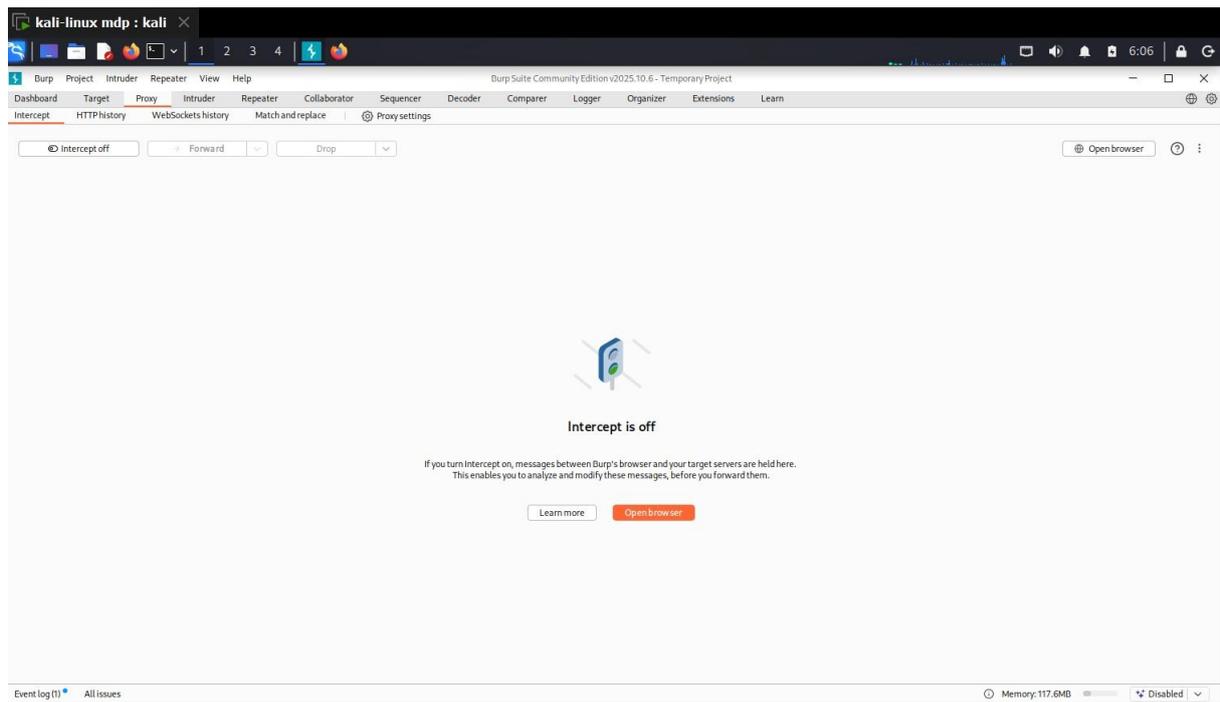
Nous allons donc utiliser le logiciel Burp Suite :



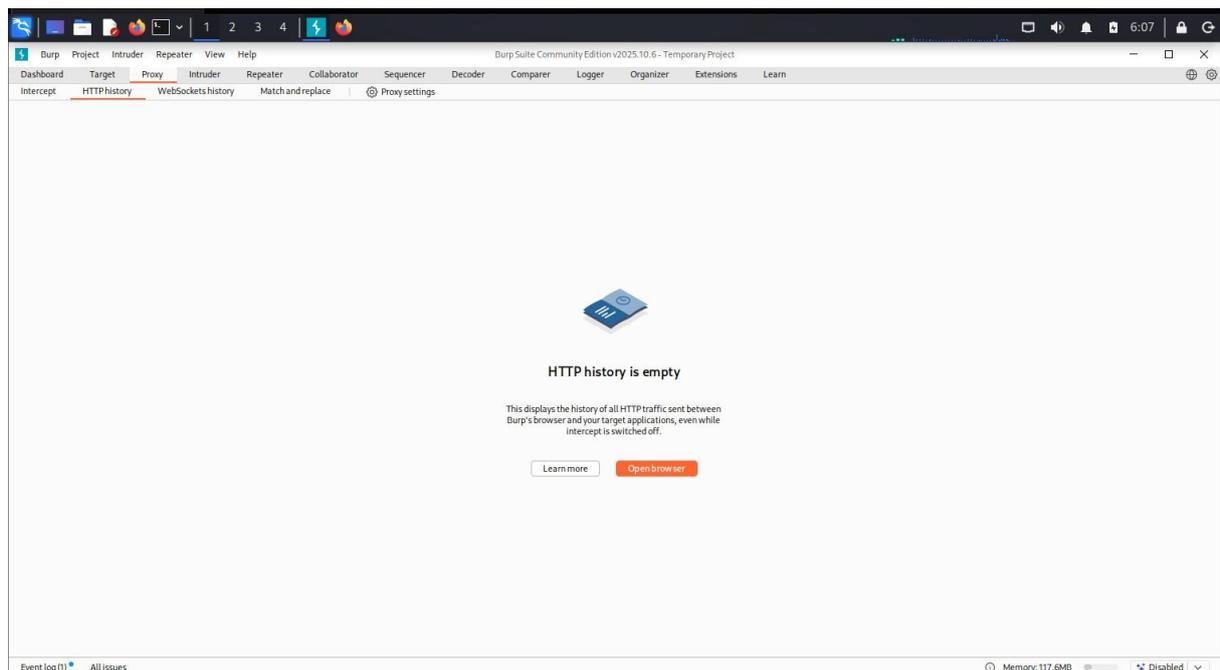
Dans Burp Suite, nous avons l'onglet « Target », qui permet de définir un site web victime, et de limiter notre vue, notre scope envers la victime



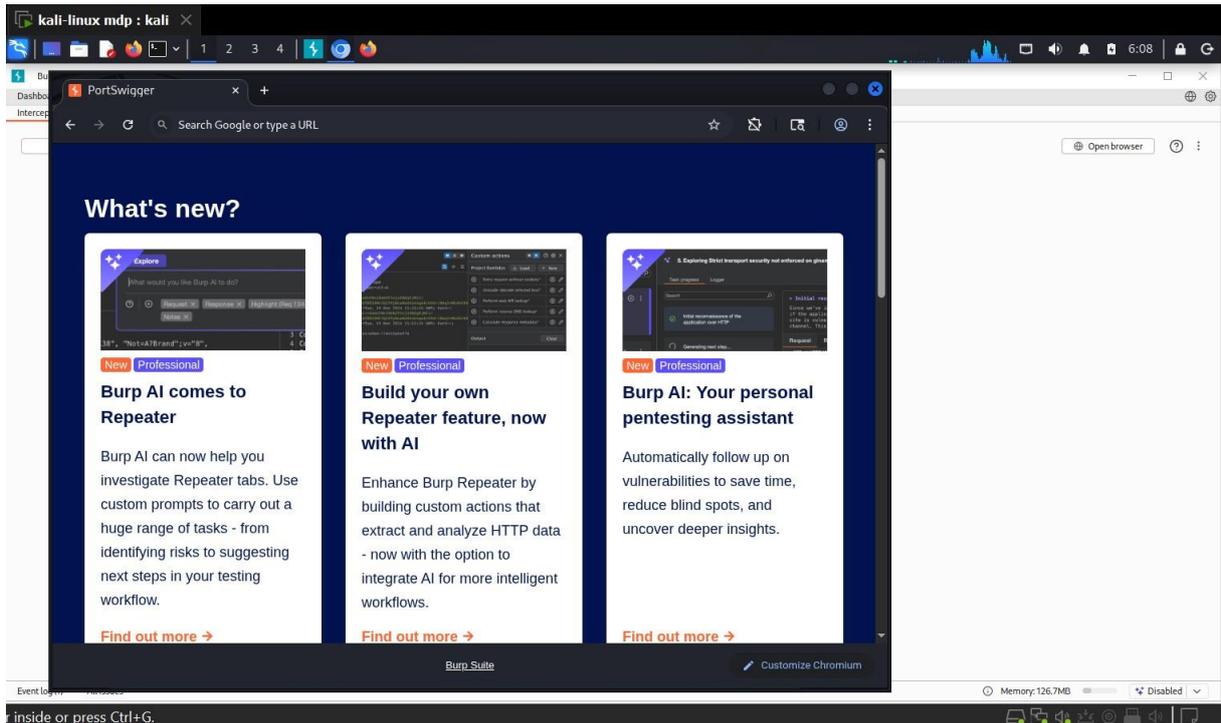
Nous avons également l'onglet « Proxy » qui permet d'intercepter et analyser le trafic web :



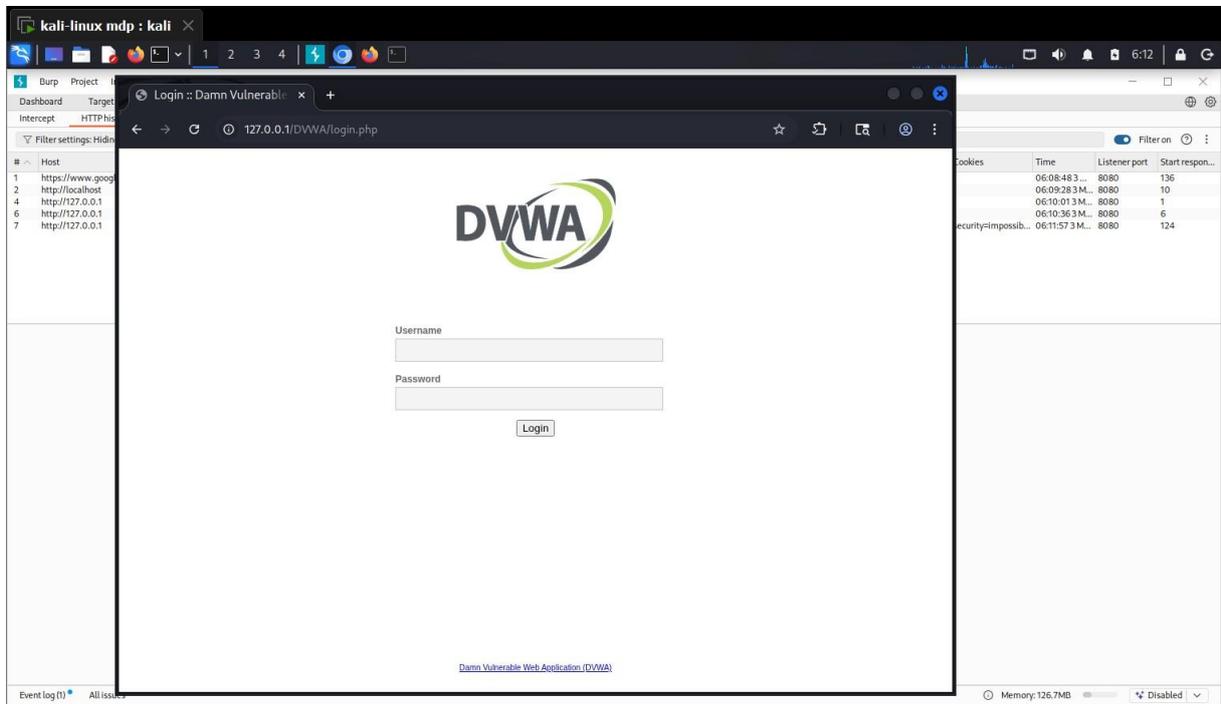
Dans l'option « http history », on peut voir le trafic, et choisir les paquets importants :



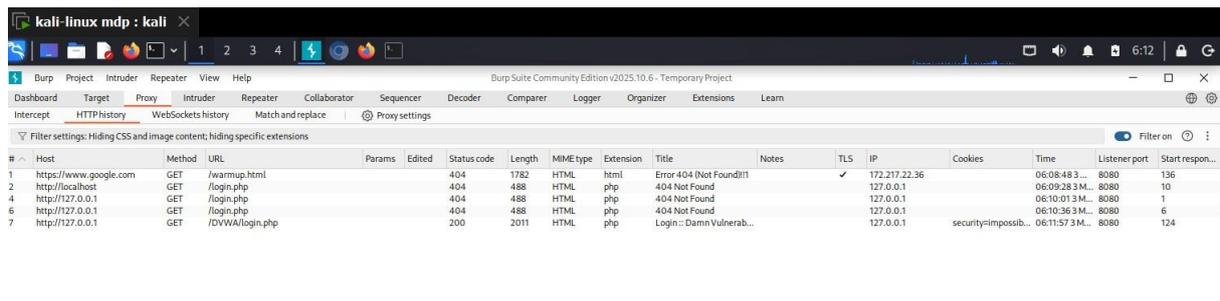
Dans l'option « Proxy », « Intercept » on clique sur « Open Browser ». Cela ouvre un navigateur qui possède Burp en tant que Proxy :



Connectons-nous sur DVWA avec ce navigateur :

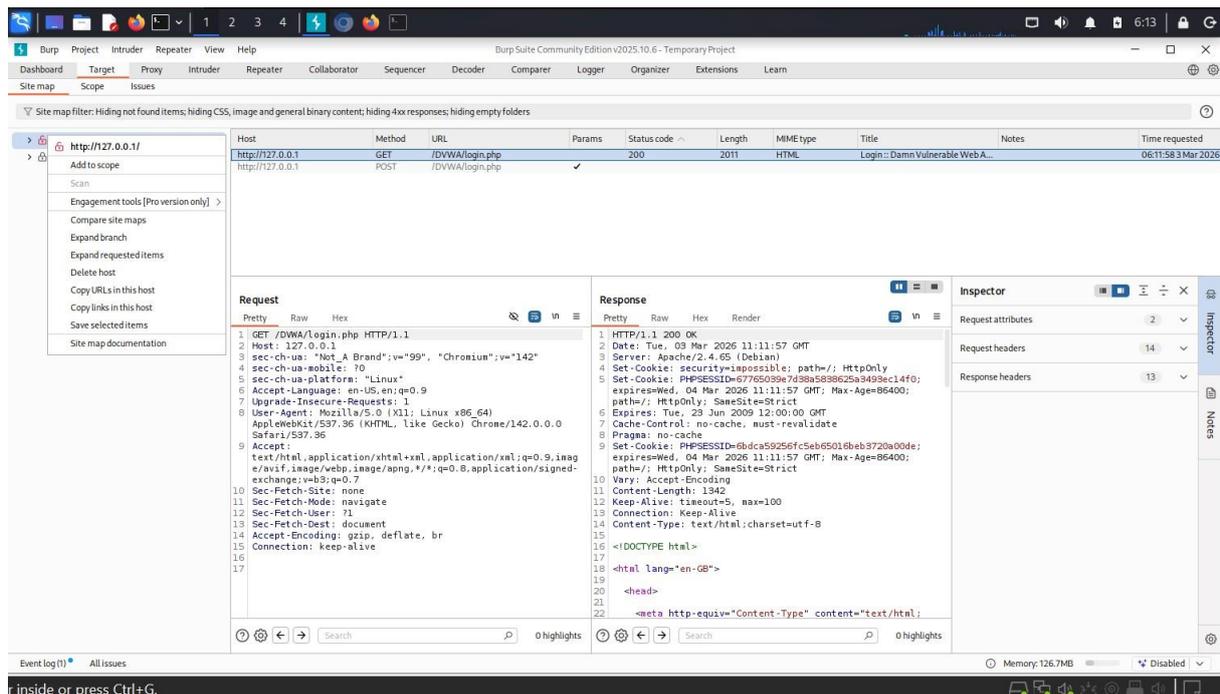


Analysons les requêtes http lors de notre connexion :



#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start respon...
1	https://www.google.com	GET	/warmup.html			404	1782	HTML	html	Error 404 (Not Found)!!!		✓	172.217.22.36		06:08:48.3...	8080	136
2	http://localhost	GET	/login.php			404	488	HTML	php	404 Not Found			127.0.0.1		06:09:28.3M...	8080	10
4	http://127.0.0.1	GET	/login.php			404	488	HTML	php	404 Not Found			127.0.0.1		06:10:01.3M...	8080	1
6	http://127.0.0.1	GET	/login.php			404	488	HTML	php	404 Not Found			127.0.0.1		06:10:36.3M...	8080	6
7	http://127.0.0.1	GET	/DVWA/login.php			200	2011	HTML	php	Login: Damn Vulnerab...			127.0.0.1	security=impossib...	06:11:57.3M...	8080	124

Dans l'onglet « Target », on sélectionne l'IP 127.0.0.1 et on clique sur « Add to scope » :



The screenshot shows the Burp Suite interface with the 'Target' tab selected. The 'Site map' section on the left shows a tree view with 'http://127.0.0.1/' expanded, and a context menu open with 'Add to scope' selected. The main area displays the details of a request and response for the selected item.

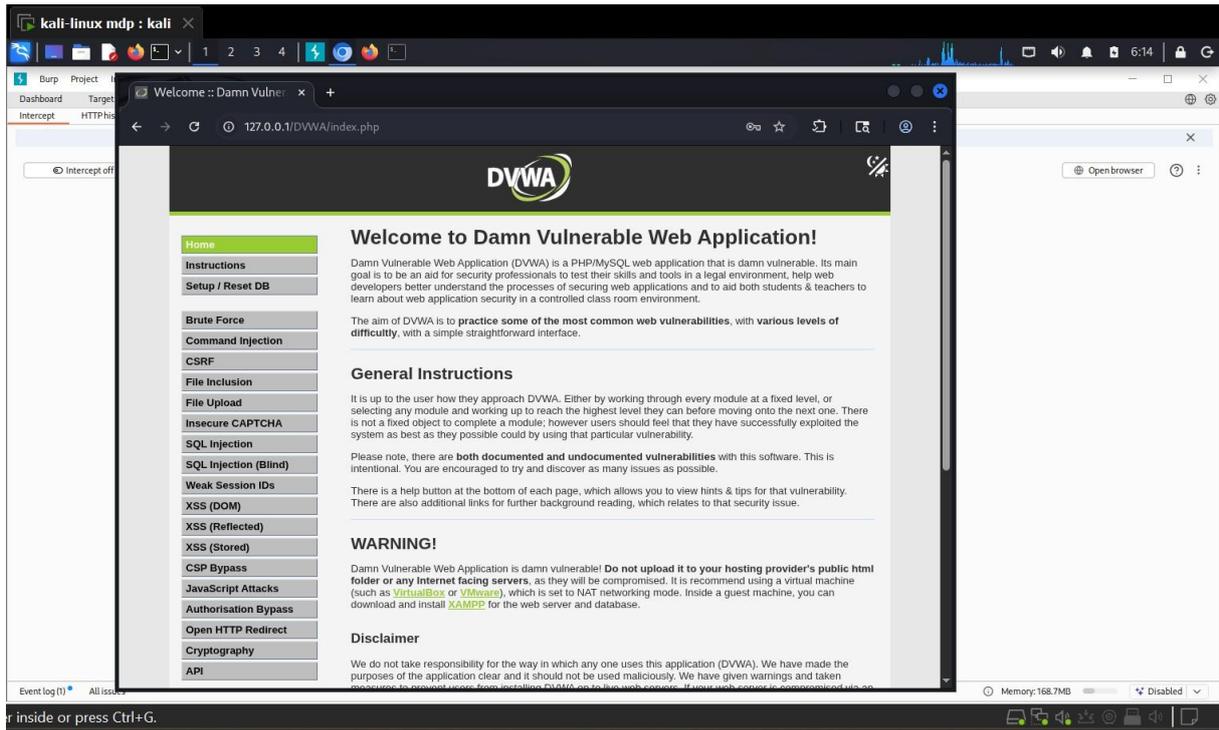
**Request**

```
1 GET /DVWA/login.php HTTP/1.1
2 Host: 127.0.0.1
3 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="142"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Connection: keep-alive
```

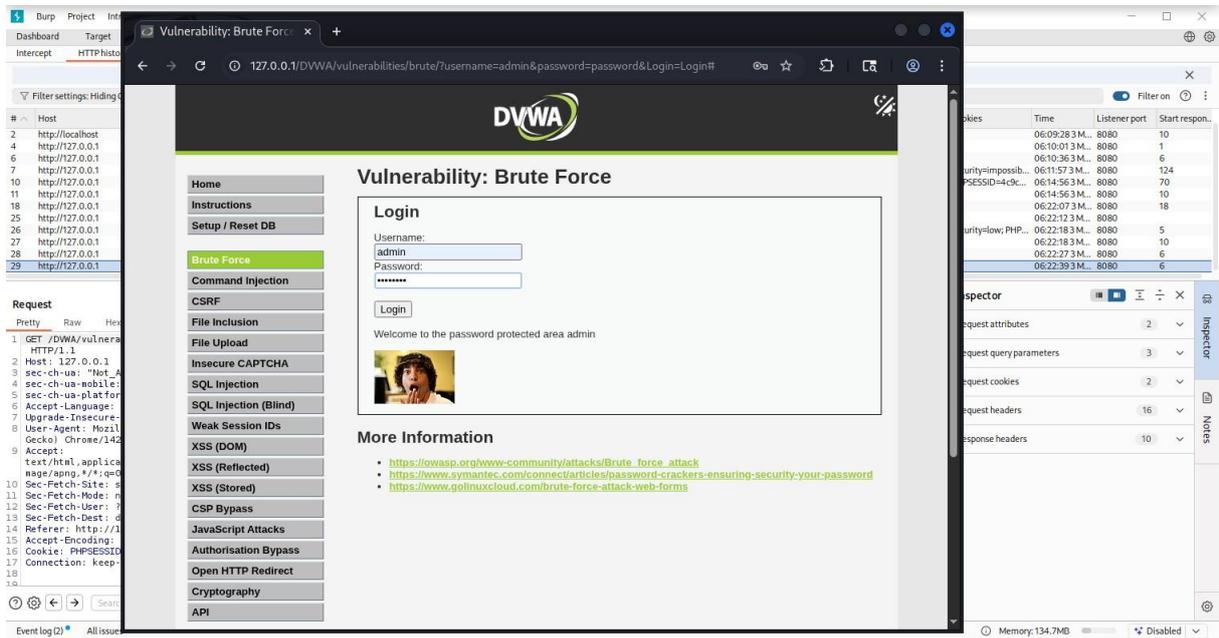
**Response**

```
1 HTTP/1.1 200 OK
2 Date: Tue, 03 Mar 2026 11:11:57 GMT
3 Server: Apache/2.4.18 (Debian)
4 Set-Cookie: security=impossible; path=/; HttpOnly
5 Set-Cookie: PHPSESSID=67765039e7d98a5838625a3499ec14f0; expires=Wed, 04 Mar 2026 11:11:57 GMT; Max-Age=86400; path=/; HttpOnly; SameSite=Strict
6 Expires: Tue, 23 Jun 2009 12:00:00 GMT
7 Cache-Control: no-cache, must-revalidate
8 Pragma: no-cache
9 Set-Cookie: PHPSESSID=6bdca59256f5eb65016eb3720a00de; expires=Wed, 04 Mar 2026 11:11:57 GMT; Max-Age=86400; path=/; HttpOnly; SameSite=Strict
10 Vary: Accept-Encoding
11 Content-Length: 1342
12 Keep-Alive: timeout=5, max=100
13 Connection: Keep-Alive
14 Content-Type: text/html; charset=utf-8
15
16 <!DOCTYPE html>
17
18 <html lang="en-GB">
19
20 <head>
21
22 <meta http-equiv="Content-Type" content="text/html;
```

Nous sommes désormais connectés sur notre machine DVWA, avec Burp en tant que Proxy :



Utilisons désormais nos logins dans l'atelier Brute Force :



## Analysons les requêtes http suite à notre login :

The screenshot shows the Burp Suite interface with the HTTP history table and the request/response inspector. The selected request is:

```
1 GET /DVWA/vulnerabilities/brute/?username=admin&password=password HTTP/1.1
2 Host: 127.0.0.1
3 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="142"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
9 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
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://127.0.0.1/DVWA/vulnerabilities/brute/
15 Accept-Encoding: gzip, deflate, br
16 Cookie: PHPSESSID=4c9c1f54af3695c8741944aee765f5; security=low
17 Connection: keep-alive
```

The response is:

```
1 HTTP/1.1 200 OK
2 Date: Tue, 03 Mar 2026 11:22:39 GMT
3 Server: Apache/2.4.18 (Debian)
4 Expires: Tue, 23 Jun 2009 12:00:00 GMT
5 Cache-Control: no-cache, must-revalidate
6 Pragma: no-cache
7 Vary: Accept-Encoding
8 Content-Length: 4778
9 Keep-Alive: timeout=5, max=100
10 Connection: Keep-Alive
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: Brute Force :: Damn Vulnerable Web Application (DVWA)
22 </title>
```

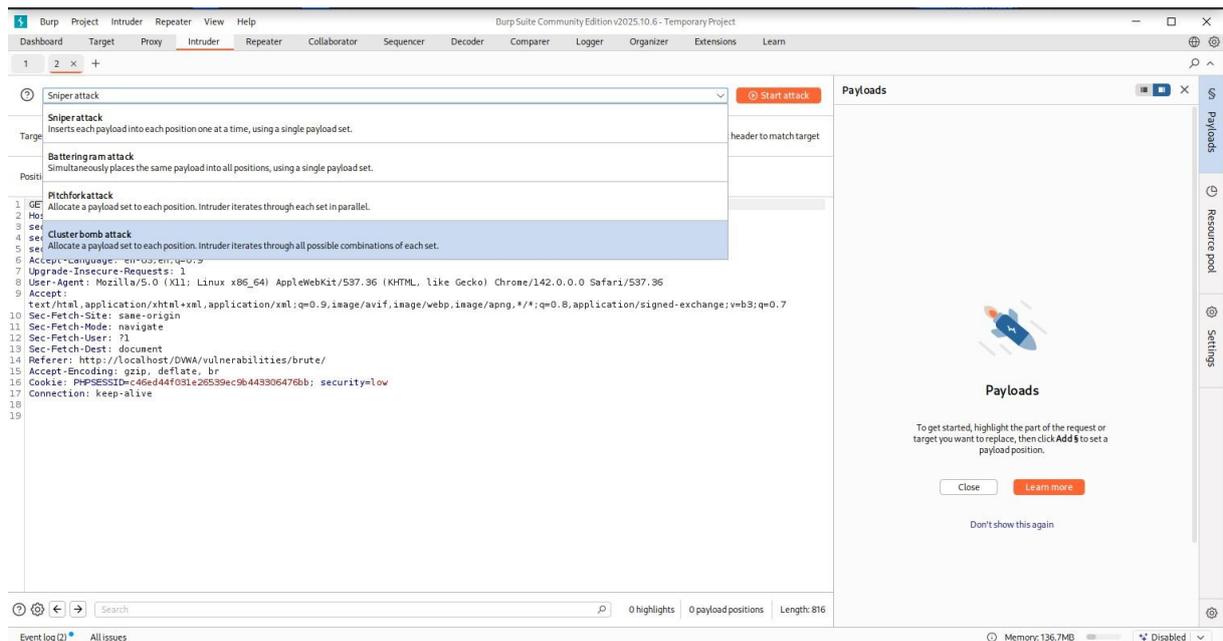
On retrouve bien nos informations de connexion dans les requêtes http.

Sélectionnons la requête http contenant les informations de connexion. On utilise l'option « Send to Intruder » :

The screenshot shows the Burp Suite interface with the context menu open for the selected request. The 'Send to Intruder' option is highlighted, indicating the process of preparing an attack.

« Send to intruder » nous permet de modifier la requête http qui contient le login et le password afin de préparer une attaque

## On sélectionne « Cluster Bomb attack » :

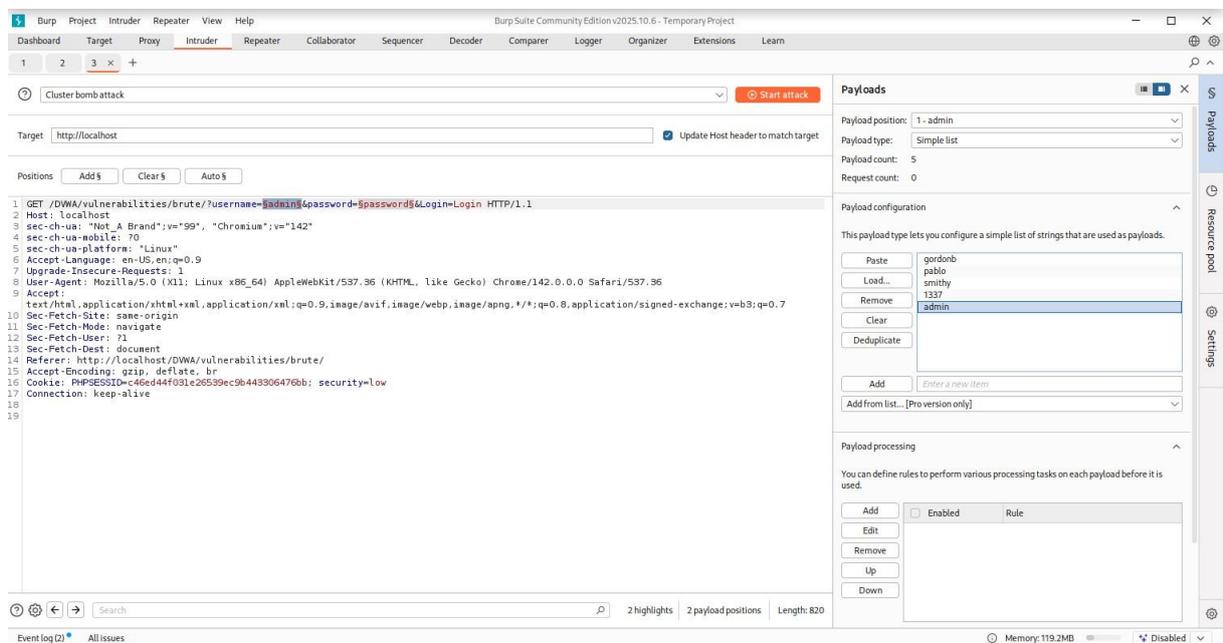


The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. The 'Cluster bomb attack' is highlighted in the list of attack types. The main window displays the request details for a GET request to `http://localhost/DWA/vulnerabilities/brute/`. The request includes headers like `Accept-Language`, `Upgrade-Insecure-Requests`, `User-Agent`, `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`, `Sec-Fetch-Mode`, `Sec-Fetch-User`, `Sec-Fetch-Dest`, `Referer`, `Accept-Encoding`, `Cookie`, and `Connection`.

On utilise l'option « add \$ » afin de définir des variables pour « username » et « password »

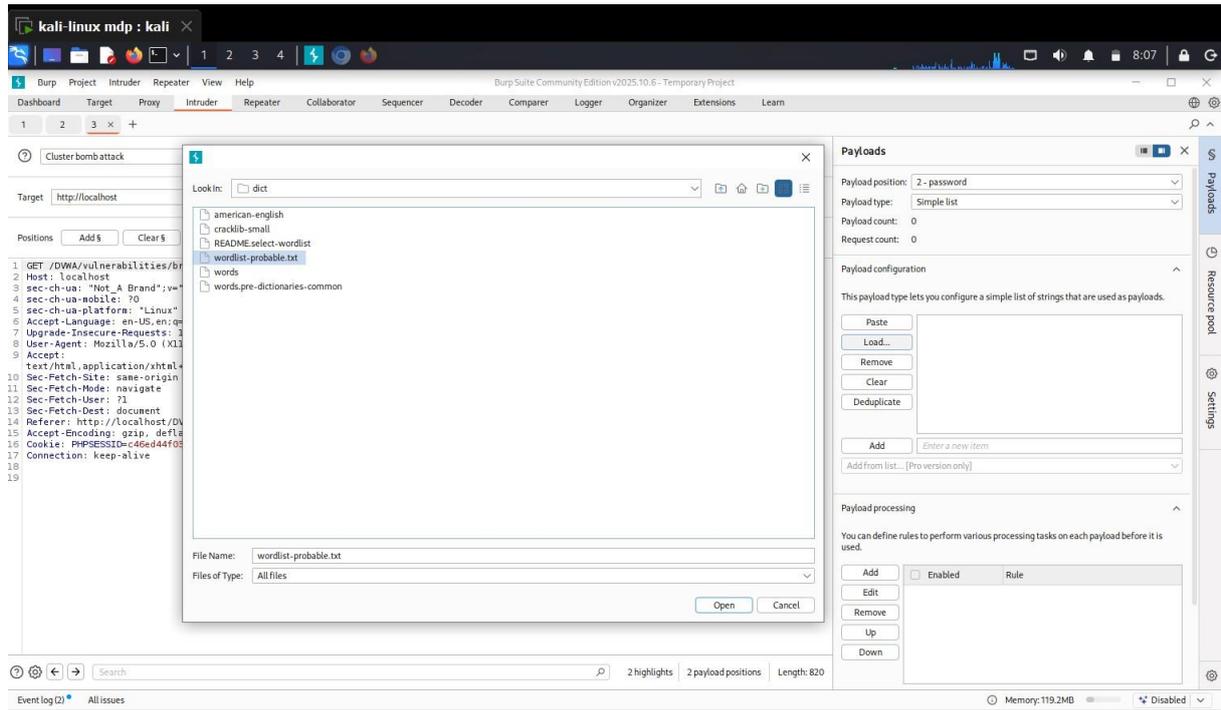
Payload nous permet d'attribuer des valeurs pour les variables \$

Dans Payload, on renseigne les différents username trouvés précédemment :

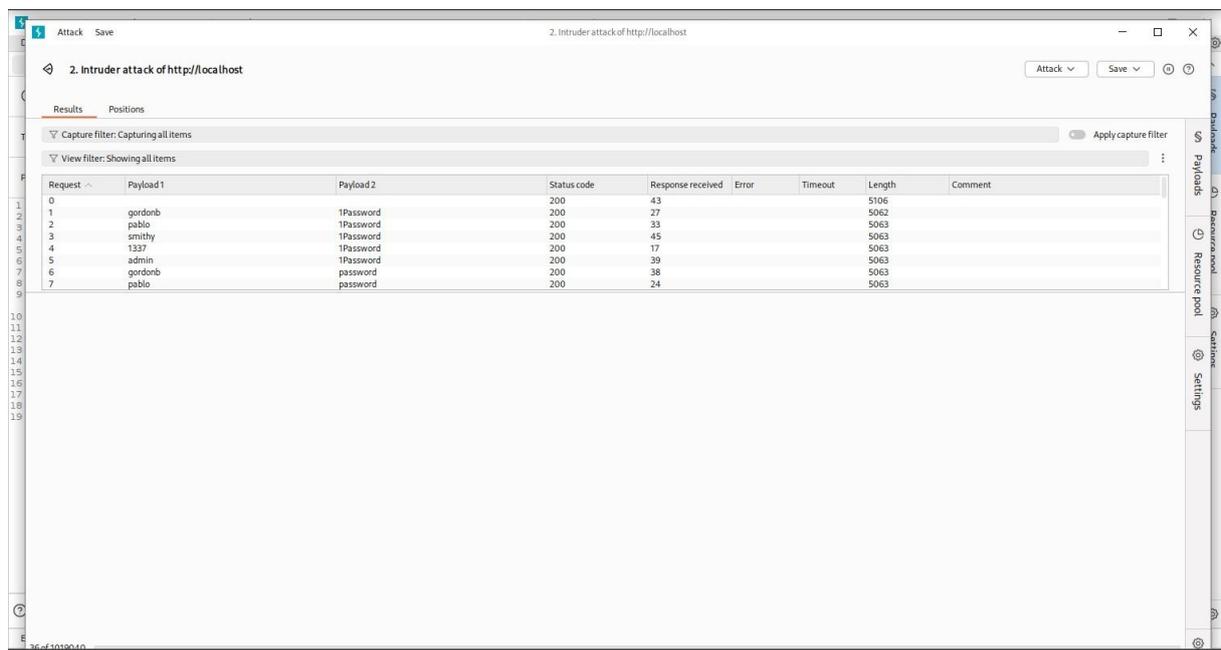


The screenshot shows the Burp Suite interface with the 'Cluster bomb attack' selected. The 'Target' is set to `http://localhost`. The 'Payloads' panel on the right is open, showing a list of usernames: `gordonb`, `pablo`, `smithy`, and `1337`. The 'Payload configuration' section is visible, and the 'Payload processing' section is also open, showing a table with columns for 'Enabled' and 'Rule'.

Pour la variable « password », on utilise le fichier « wordlist-probable.txt » qui est un fichier comprenant une liste de mot de passe :



On clique ensuite sur « Start Attack » :



L'attaque est désormais lancée. Le logiciel va désormais envoyer une série de requêtes http avec les différents username et password renseignés précédemment.

Ici nous retrouvons dans cette requête nos informations de connexion :

The screenshot shows the Burp Suite interface during an intruder attack. The main window displays a table of captured requests and responses. The selected request (Request 10) is shown in detail below the table.

Request #	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
5	admin	!Password	200	39			5063	
6	gordonb	password	200	38			5063	
7	pablo	password	200	24			5063	
8	smithy	password	200	30			5108	
9	1337	password	200	12			5063	
10	admin	password	200	12			5105	
11	gordonb	123456789	200	7			5062	
12	pablo	123456789	200	7			5062	

**Request 10 Details:**

```
1 GET /DWA/vulnerabilities/brute/?username=admin&password=password&login=Login HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Not A Brand";v="99", "Chromium";v="142"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
9 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
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://localhost/DWA/vulnerabilities/brute/
15 Accept-Encoding: gzip, deflate, br
16 Cookie: PHPSESSID=c66e44f031cc399cc9b443306476bb; security=low
17 Connection: keep-alive
18
19
```