

Презентация по индивидуальной проекте 5

Основы информационной безопасности

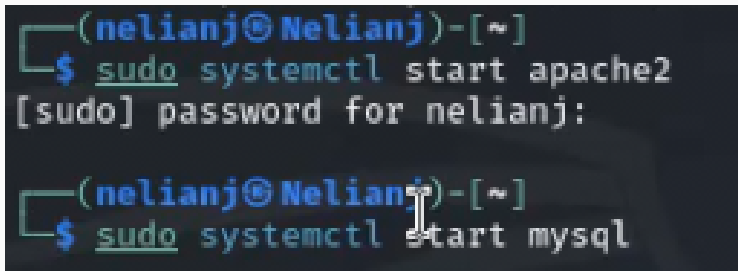
Нджову Н.

16 мая 2025

Российский университет дружбы народов, Москва, Россия

Научиться использовать Burp Suite.

Я запускаю локальный сервер, на котором открою веб-приложение DVWA для тестирования инструмента Burp Suite(рис.1).

A terminal window with a dark background and light blue text. The prompt is '(nelianj@Nelianj)-[~]'. The first command is '\$ sudo systemctl start apache2', followed by '[sudo] password for nelianj:'. The second command is '\$ sudo systemctl start mysql'.

```
(nelianj@Nelianj)-[~]  
$ sudo systemctl start apache2  
[sudo] password for nelianj:  
  
(nelianj@Nelianj)-[~]  
$ sudo systemctl start mysql
```

Рис. 1: Запуск локального сервера

Я запускаю инструмент Burp Suite. После этого я открываю сетевые настройки браузера, для подготовке к работе(рис.2).

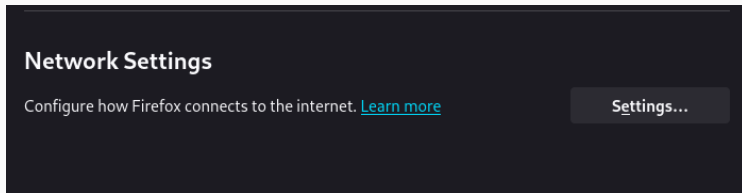


Рис. 2: Сетевые настройки браузера

Выполнение лабораторной работы

Я изменяю настройки сервера для работы с прокси и захватом данных с помощью Burp Suite(рис.3)

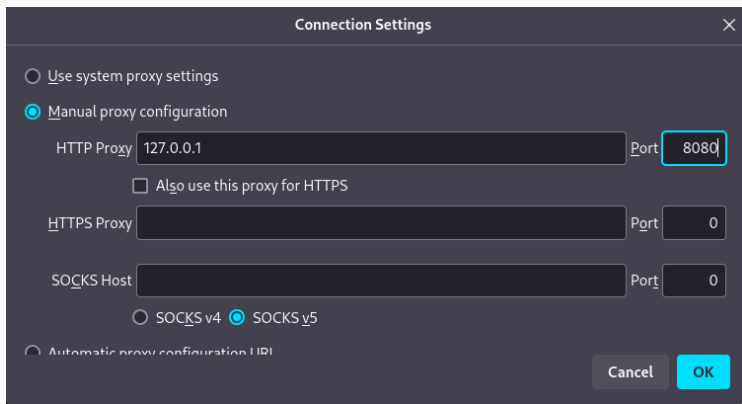


Рис. 3: Настройки сервера

Я изменяю настройки Proxy инструмента Burp Suite для дальнейшей работы(рис.4).

The screenshot displays the Burp Suite configuration window for the Proxy tool. The left sidebar shows the 'Tools' menu with 'Proxy' selected. The main panel is titled 'Tools > Proxy' and contains two sections: 'Proxy listeners' and 'Request interception rules'.

Proxy listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to config

Running	Interface	Invisible	Redirect	Certificate
<input checked="" type="checkbox"/>	127.0.0.1:8080			Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating T or another installation of Burp.

Import / export CA certificate Regenerate CA certificate

Request interception rules

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☒ Intercept requests based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="checkbox"/>	Or	File extension	Does not match	(^gif\$ ^jpg\$
<input type="checkbox"/>		Request	Contains parameters	
<input type="checkbox"/>		HTTP method	Does not match	(oetloost)

Во вкладке Proxy устанавливаю “Intercept is on”(рис.5).

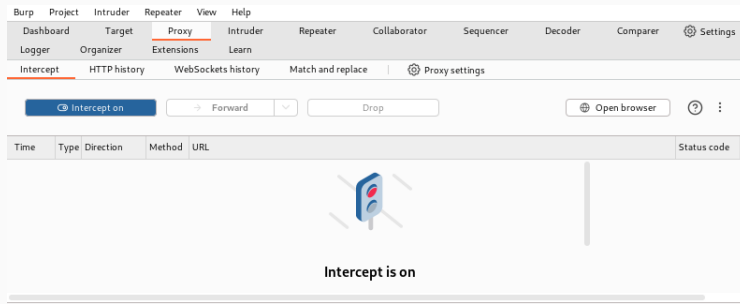


Рис. 5: Настройки Proxy

Чтобы Burp Suite исправно работал с локальным сервером, необходимо установить параметр `network.allow_hijacking_localhost` на `true` (рис.6).

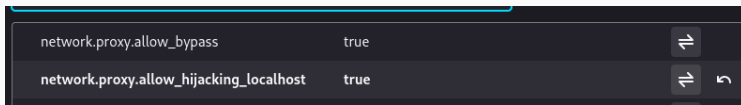


Рис. 6: Настройки параметров

Я пытаюсь зайти в браузере на DVWA, тут же во вкладки Proxy появляется захваченный запрос. Нажимаем “Forward”, чтобы загрузить страницу(рис.7 и рис.8).

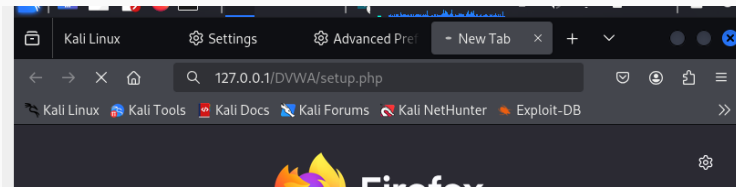


Рис. 7: Получаемые запросы сервера

Выполнение лабораторной работы

The screenshot displays the Burp Suite application interface. The top menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'View', and 'Help'. Below it, a secondary menu bar contains 'Dashboard', 'Target', 'Proxy' (highlighted), 'Intruder', 'Repeater', 'Collaborator', 'Sequencer', 'Decoder', 'Comparer', and 'Settings'. A third bar shows 'Intercept' (highlighted), 'HTTP history', 'WebSockets history', 'Match and replace', and 'Proxy settings'. The main workspace is divided into three sections: a top bar with 'Intercept on' (blue), 'Forward' (orange), and 'Drop' (white) buttons, along with a 'Request to h...' field and an 'Open browser' button; a middle table with columns 'Type', 'Direction', 'Method', 'URL', 'Status code', and 'Length'; and a bottom section split into 'Request' and 'Inspector'. The 'Request' section shows a GET request to '/DVWA/setup.php' with various headers like 'Host', 'User-Agent', 'Accept', 'Accept-Language', 'Accept-Encoding', 'Connection', 'Upgrade-Insecure-Requests', 'Sec-Fetch-Dest', 'Sec-Fetch-Mode', 'Sec-Fetch-Site', 'Sec-Fetch-User', and 'Priority'. The 'Inspector' section on the right lists 'Request attributes' (2), 'Request query parameters' (0), 'Request body parameters' (0), 'Request cookies' (0), and 'Request headers' (12).

Type	Direction	Method	URL	Status code	Length
Request		GET	/DVWA/setup.php		

Request

Pretty Raw Hex

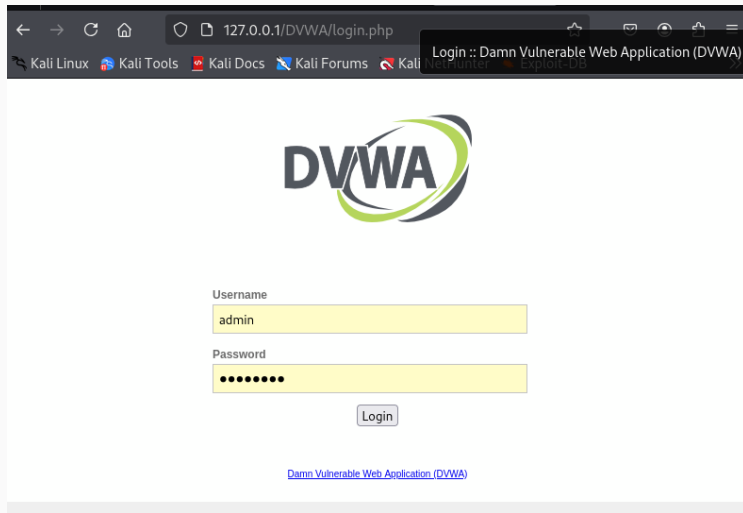
```
1 GET /DVWA/setup.php HTTP/1.1
2 Host: 127.0.0.1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
  Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
  /webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: none
12 Sec-Fetch-User: ?1
13 Priority: u=0, i
14
15
```

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 0
- Request cookies: 0
- Request headers: 12

Рис. 8: Получаемые запросы сервера

Загрузилась страница авторизации, текст запроса поменялся(рис.9 и рис.10).



Выполнение лабораторной работы

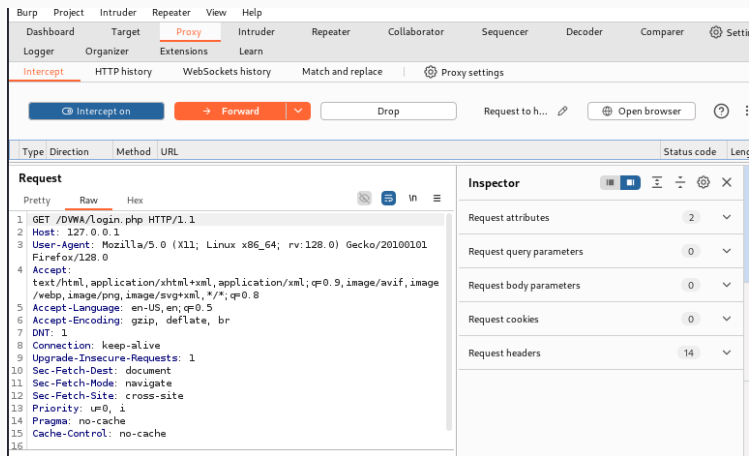


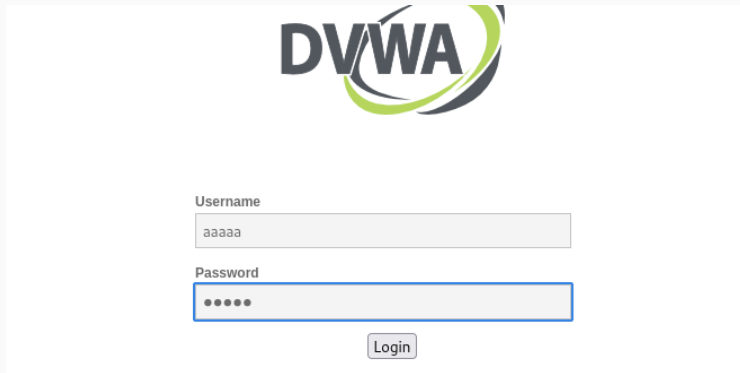
Рис. 10: Страница авторизации

История запросов хранится во вкладке Target (рис.11).

Host	Method	URL	Params	Status code
http://127.0.0.1	GET	/DVWA/dvwa/js/add_e...		200
http://127.0.0.1	GET	/DVWA/dvwa/js/dvwa...		200
http://127.0.0.1	GET	/DVWA/login.php		200
http://127.0.0.1	GET	/DVWA/setup.php		200
http://127.0.0.1	POST	/DVWA/setup.php	✓	302
http://127.0.0.1	GET	/DVWA/dvwa/images/L...		304
http://127.0.0.1	GET	/DVWA/dvwa/images/L...		304

Рис. 11: История запросов

Я попробую ввести неправильные, случайные данные в веб-приложении и нажмем **Login**. В запросе увидим строку, в которой отображаются введенные нами данные, то есть поле для ввода(рис.12 и рис.13).



The image shows the DVWA (Damn Vulnerable Web Application) login interface. At the top is the DVWA logo, which consists of the letters 'DVWA' in a bold, dark font, with a stylized green and blue swoosh graphic to the right. Below the logo are two input fields. The first field is labeled 'Username' and contains the text 'aaaaa'. The second field is labeled 'Password' and contains five black dots, indicating a masked password. Below these fields is a button labeled 'Login'.

Рис. 12: Ввод случайных данных

Request

Pretty Raw Hex

```
1 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
2 /webp,image/png,image/svg+xml,*/*;q=0.8
3
4
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: 83
9 Origin: http://127.0.0.1
10 Connection: keep-alive
11 Referer: http://127.0.0.1/DVWA/login.php
12 Cookie: security=impossible; PHPSESSID=2g1g7427mlm9ngnmm0co2f12ph
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 Priority: u=0, i
19
20 username=aaa&password=aaaaa&Login=Login&user_token=
  33adb71dfc58d5a21b5c2ebdf1023337
```

Рис. 13: Ввод случайных данных

Этот запрос так же можно найти во вкладке Target, там же жмем правой кнопкой мыши на хост нужного запроса, и далее нажимаем “Send to Intruder”(рис.14).

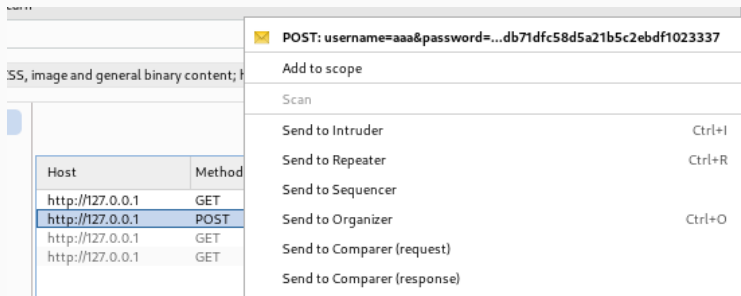
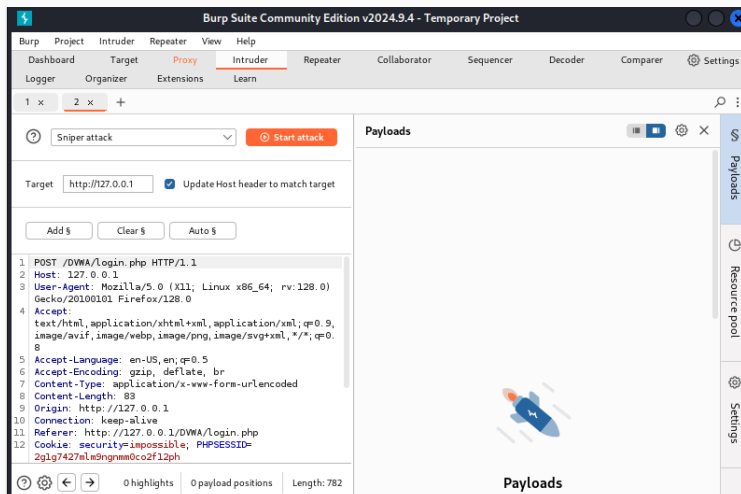


Рис. 14: POST-запрос с вводом пароля и логина

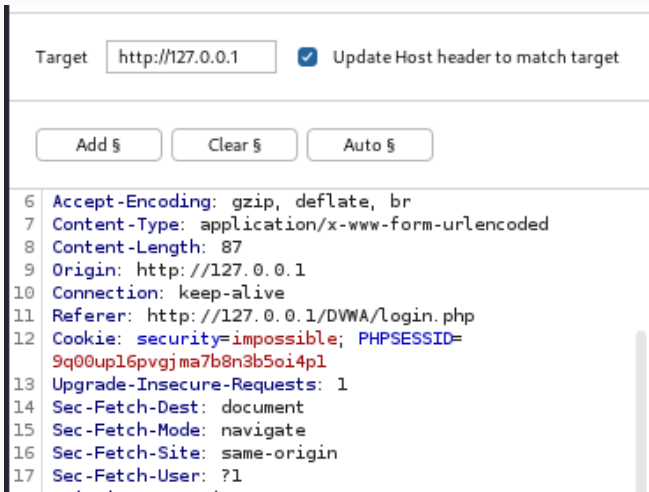
Выполнение лабораторной работы

Я попадаю на вкладку Intruder, вижу значения по умолчанию у типа атаки и наш запрос(рис.15).



Выполнение лабораторной работы

Я изменяю значение типа атаки на Cluster bomb и проставляю специальные символы у тех данных в форме для ввода, которые буду пробивать(рис.16).



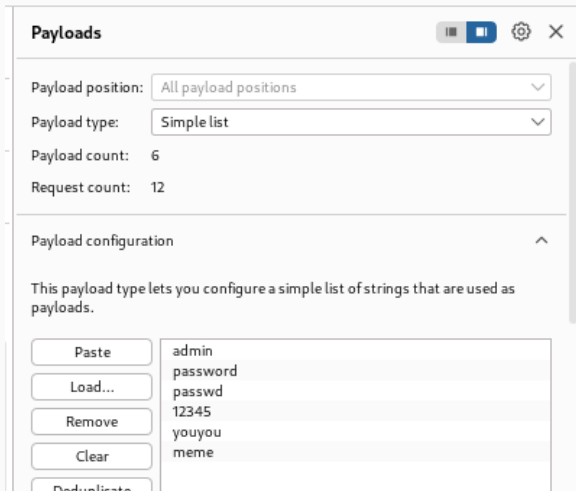
The screenshot shows a web application security tool interface. At the top, there is a 'Target' field containing 'http://127.0.0.1' and a checked checkbox labeled 'Update Host header to match target'. Below this, there are three buttons: 'Add §', 'Clear §', and 'Auto §'. The main area displays a list of HTTP headers, numbered 6 through 17. The headers are: 'Accept-Encoding: gzip, deflate, br', 'Content-Type: application/x-www-form-urlencoded', 'Content-Length: 87', 'Origin: http://127.0.0.1', 'Connection: keep-alive', 'Referer: http://127.0.0.1/DVWA/login.php', 'Cookie: security=impossible; PHPSESSID=9q00upl6pvgjma7b8n3b5oi4pl', 'Upgrade-Insecure-Requests: 1', 'Sec-Fetch-Dest: document', 'Sec-Fetch-Mode: navigate', 'Sec-Fetch-Site: same-origin', and 'Sec-Fetch-User: ?1'.

Target ☒ Update Host header to match target

```
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 87
9 Origin: http://127.0.0.1
10 Connection: keep-alive
11 Referer: http://127.0.0.1/DVWA/login.php
12 Cookie: security=impossible; PHPSESSID=
13          9q00upl6pvgjma7b8n3b5oi4pl
14 Upgrade-Insecure-Requests: 1
15 Sec-Fetch-Dest: document
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-Site: same-origin
18 Sec-Fetch-User: ?1
```

Выполнение лабораторной работы

Так как мне нужно параметра для подбора, то нужно списка со значениями для подбора(рис.17).



Payloads

Payload position: All payload positions

Payload type: Simple list

Payload count: 6

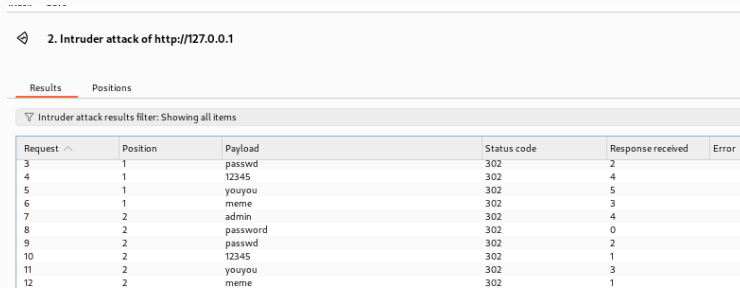
Request count: 12

Payload configuration

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

Paste	admin
Load...	password
Remove	passwd
Clear	12345
Deduplicate	youyou
	meme

Я запускаю атаку и начинаю подбор(рис.18).



The screenshot displays a web application security tool interface. At the top, there is a header "2. Intruder attack of http://127.0.0.1". Below this, there are two tabs: "Results" (selected) and "Positions". A filter bar indicates "Intruder attack results filter: Showing all items". The main content is a table with the following columns: Request, Position, Payload, Status code, Response received, and Error. The table contains 10 rows of data, showing the results of an intruder attack. The "Request" column shows request numbers 3 through 12. The "Position" column shows positions 1, 2, and 3. The "Payload" column shows various payloads including "passwd", "12345", "youyou", "meme", "admin", and "password". The "Status code" column shows "302" for all requests. The "Response received" column shows the number of responses received for each request. The "Error" column is empty for all requests.

Request ^	Position	Payload	Status code	Response received	Error
3	1	passwd	302	2	
4	1	12345	302	4	
5	1	youyou	302	5	
6	1	meme	302	3	
7	2	admin	302	4	
8	2	password	302	0	
9	2	passwd	302	2	
10	2	12345	302	1	
11	2	youyou	302	3	
12	2	meme	302	1	

Рис. 18: Запуск атаки

Выполнение лабораторной работы

При открытии результата каждого post-запроса можно увидеть полученный get-запрос, в нем видно, куда нас перенаправило после выполнения ввода пары пользователь-пароль. В представленном случае с подбором пары passwd-password нас перенаправило на login.php, это значит, что пара не подходит(рис.19).

The screenshot shows the Burp Suite interface. At the top, the 'Intruder attack results filter' shows 'Showing all items'. Below this is a table of attack results:

Requ...	Position	Payload	Status code	Response...	Error	Timeout	Length	Comment
0	0		302	1			476	
1	1	admin	302	5			475	
2	1	passwd	302	2			476	
3	1	123456	302	0			475	

Below the table, the 'Response' tab is selected, showing the HTTP response for the selected payload (index 2). The response is an HTTP 302 Found status, indicating a redirect. The 'Location' header is 'login.php'. The 'Content-Type' is 'text/html; charset=UTF-8'.

```
1 HTTP/1.1 302 Found
2 Date: Thu, 15 May 2025 20:00:32 GMT
3 Server: Apache/2.4.62 (Debian)
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT
5 Cache-Control: no-store, no-cache, must-revalidate
6 Pragma: no-cache
7 Set-Cookie: PHPSESSID=ii693lk4njldd7624ipp1qh9bo; expires=Fri, 16 May 2025 20:00:32 GMT; Max-Age=86400; path=/;
  HttpOnly; SameSite=Strict
8 Location: login.php
9 Content-Length: 0
10 Keep-Alive: timeout=5, max=100
11 Connection: Keep-Alive
12 Content-Type: text/html; charset=UTF-8
13
```

Рис. 19: Результат запроса

Я проверяю результат пары admin-password во вкладке Response, теперь нас перенаправляет на страницу index.php, значит пара должна быть верной(рис.20).

▼ Intruder attack results filter: Showing all items

Requ...	Position	Payload	Status code	Response...	Error	Timeout	Length	Comment
0	0		302	1			476	
1	1	admin	302	5			475	
2	1	paasswd	302	2			476	
3	1	123456	302	0			475	

Request Response

Pretty Raw Hex Render

```
1 HTTP/1.1 302 Found
2 Date: Thu, 15 May 2025 20:00:32 GMT
3 Server: Apache/2.4.62 (Debian)
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT
5 Cache-Control: no-store, no-cache, must-revalidate
6 Pragma: no-cache
7 Set-Cookie: PHPSESSID=cqusflseh9rt64ttjr93cfo7dr; expires=Fri, 16 May 2025 20:00:32 GMT; Max-Age=86400; path=/;
  HttpOnly; SameSite=Strict
8 Location: index.php
9 Content-Length: 0
10 Keep-Alive: timeout=5, max=99
11 Connection: Keep-Alive
12 Content-Type: text/html; charset=UTF-8
--
```

Рис. 20: Результат запроса

Дополнительная проверка с использованием Repeater, нажимаю на нужный мне запрос правой кнопкой мыши и ждем “Send to Repeater” (рис.21).

Attack Save

2. Intruder attack of http://127.0.0.1

Attack Save

Results Positions

Intruder attack results filter: Showing all items

Requ...	Position	Payload	Statu
0	0		302
1	1	admin	302
2	1	paasswd	302
3	1	123456	302
4	1	youyou	302
5	1	meme	302
6	1	bruh	302
7	2	admin	302
8	2	paasswd	302
9	2	123456	302
10	2	youyou	302
11	2	meme	302
12	2	bruh	302

Result #1

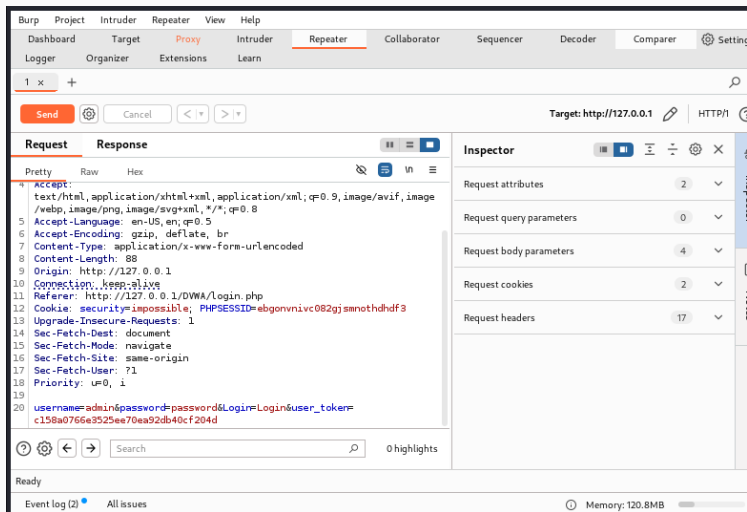
Scan

- Send to Intruder Ctrl+I
- Send to Repeater Ctrl+R
- Send to Sequencer
- Send to Organizer Ctrl+O
- Send to Comparer (request)
- Send to Comparer (response)
- Show response in browser
- Request in browser >
- Generate CSRF PoC
- Add to site map
- Request item again

Length	Comment
476	
475	
476	
475	
476	
475	
476	
475	
476	
475	
476	
475	
476	

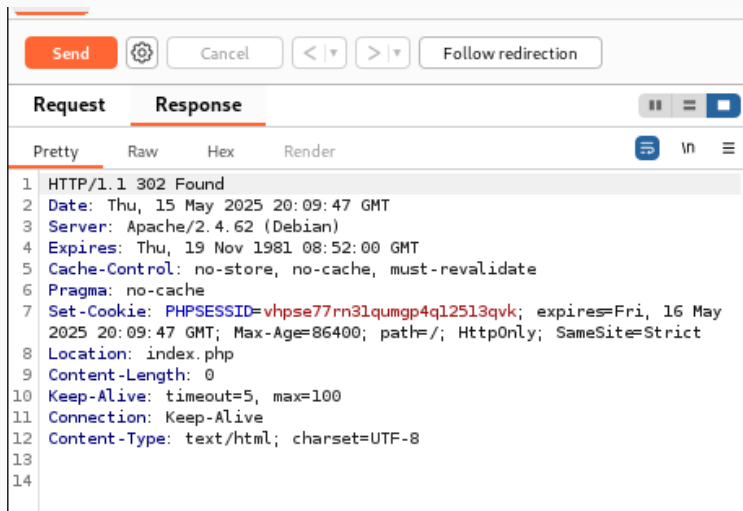
Рис. 21: Дополнительная проверка результата

Я перехожу во вкладку “Repeater”(рис.22).



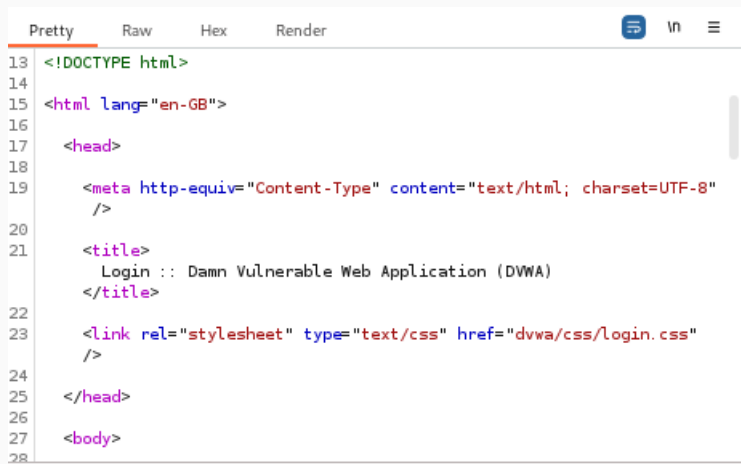
Выполнение лабораторной работы

Нажимаю "send", получаю в Response в результате перенаправление на index.php(рис.23).



Выполнение лабораторной работы

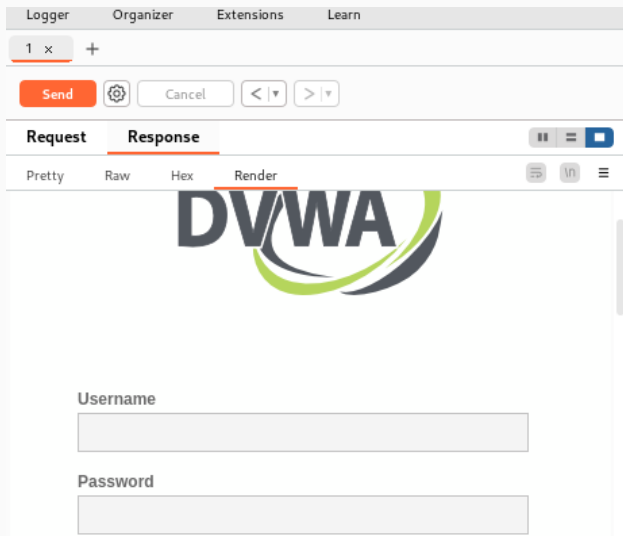
После нажатия на **Follow redirection**, получим неcompiled html код в окне Response(рис.24).



```

Pretty  Raw  Hex  Render
13 <!DOCTYPE html>
14
15 <html lang="en-GB">
16
17   <head>
18
19     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
20       />
21
22     <title>
23       Login :: Damn Vulnerable Web Application (DVWA)
24     </title>
25
26     <link rel="stylesheet" type="text/css" href="dvwa/css/login.css"
27       />
28
29   </head>
30
31   <body>
```

Далее в подокне Render получаю то, как выглядит полученная страница (рис.25).



При выполнении лабораторной работы научилась использовать инструмент Burp Suite.