

Отчет по выполнению индивидуального проекта. Этап №5

Основы информационной безопасности

Назармамадов Умед Джамshedович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	19
	Список литературы	20

Список иллюстраций

3.1	Запуск локального сервера	7
3.2	Запуск приложения	7
3.3	Сетевые настройки браузера	8
3.4	Настройки сервера	8
3.5	Настройки Burp Suite	9
3.6	Настройки Proxu	9
3.7	Настройки параметров	10
3.8	Получаемые запросы сервера	10
3.9	Страница авторизации	10
3.10	История запросов	11
3.11	Ввод случайных данных	11
3.12	POST-запрос с вводом пароля и логина	12
3.13	Вкладка Intruder	12
3.14	Изменение типа атаки	13
3.15	Первый Simple list	13
3.16	Второй Simple list	14
3.17	Запуск атаки	14
3.18	Результат запроса	15
3.19	Результат запроса	15
3.20	Дополнительная проверка результата	16
3.21	Вкладка Repeater	16
3.22	Окно Response	17
3.23	Изменение в окне Response	17
3.24	Полученная страница	18

Список таблиц

1 Цель работы

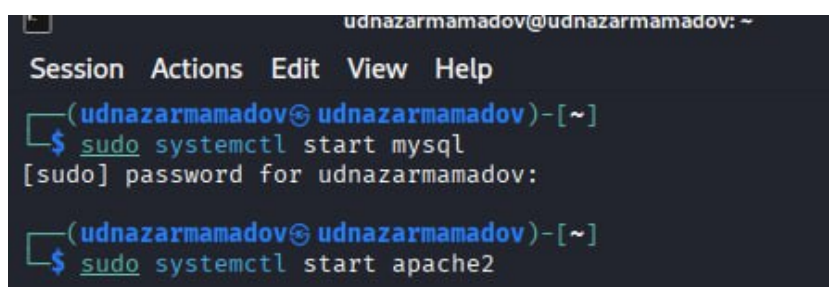
Научиться использовать Burp Suite.

2 Теоретическое введение

Burp Suite представляет собой набор мощных инструментов безопасности веб-приложений, которые демонстрируют реальные возможности злоумышленника, проникающего в веб-приложения. Эти инструменты позволяют сканировать, анализировать и использовать веб-приложения с помощью ручных и автоматических методов. Интеграция интерфейсов этих инструментов обеспечивает полную платформу атаки для обмена информацией между одним или несколькими инструментами, что делает Burp Suite очень эффективной и простой в использовании платформой для атаки веб-приложений. [1].

3 Выполнение лабораторной работы

Запускаю локальный сервер, на котором открою веб-приложение DVWA для тестирования инструмента Burp Suite (рис. [fig:001?]).



```
udnazarmamadov@udnazarmamadov: ~  
Session Actions Edit View Help  
(udnazarmamadov@udnazarmamadov)-[~]  
$ sudo systemctl start mysql  
[sudo] password for udnazarmamadov:  
(udnazarmamadov@udnazarmamadov)-[~]  
$ sudo systemctl start apache2
```

Рис. 3.1: Запуск локального сервера

Запускаю инструмент Burp Suite (рис. [fig:002?]).

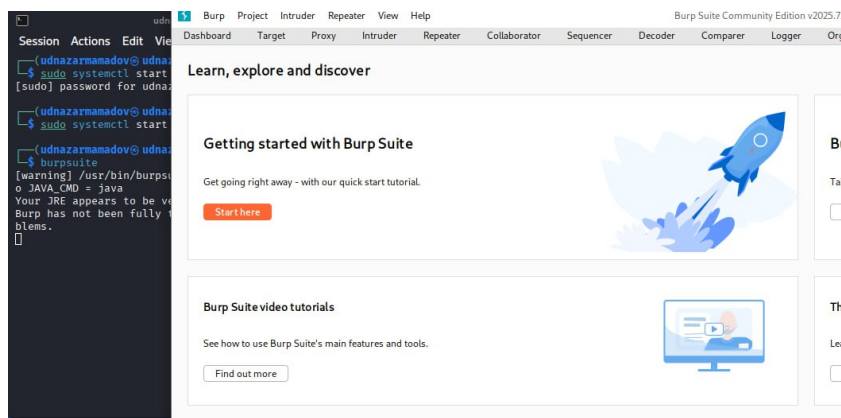


Рис. 3.2: Запуск приложения

Открываю сетевые настройки браузера, для подготовке к работе (рис. [fig:003?]).

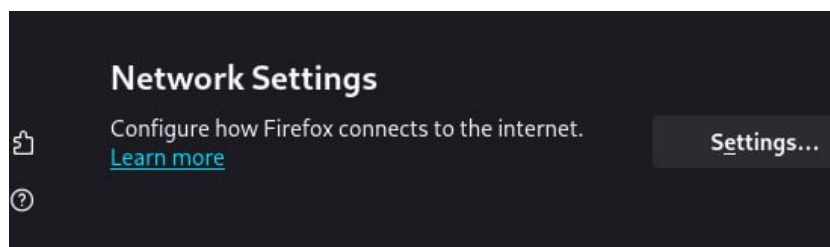


Рис. 3.3: Сетевые настройки браузера

Изменение настроек сервера для работы с проху и захватом данных с помощью Burp Suite (рис. [fig:004?]).

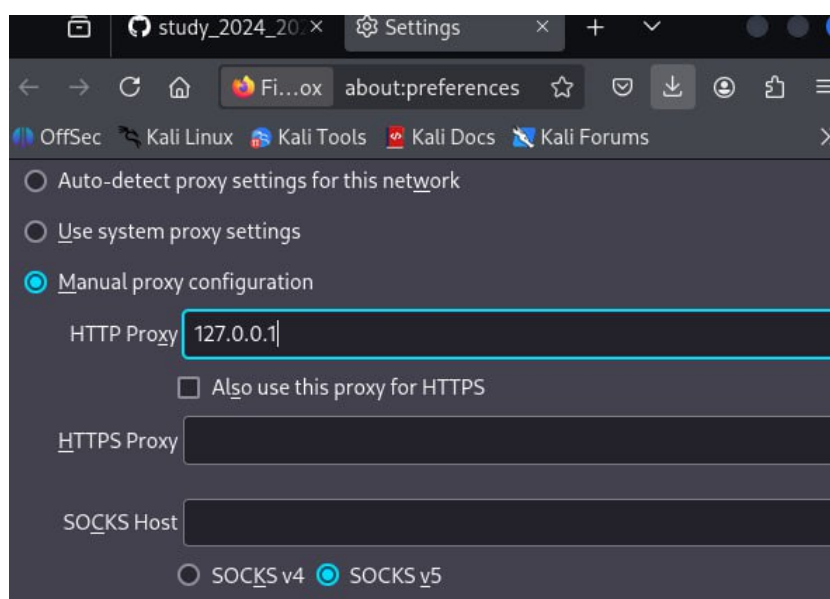


Рис. 3.4: Настройки сервера

Изменяю настройки Проху инструмента Burp Suite для дальнейшей работы (рис. [fig:005?]).

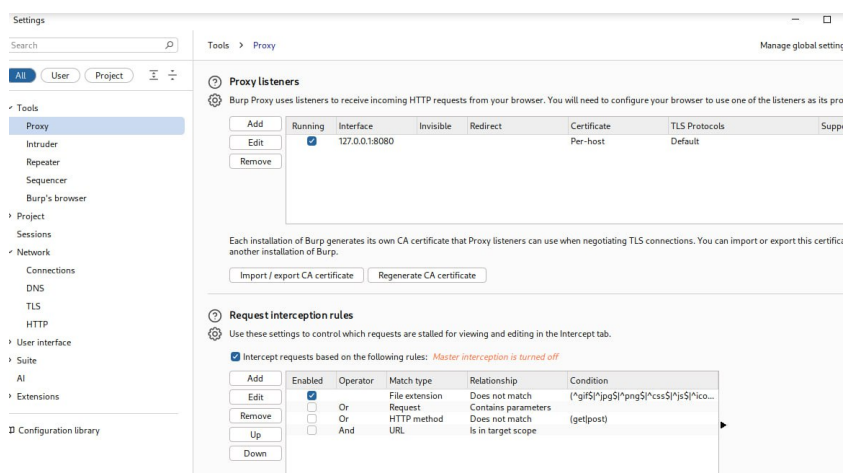


Рис. 3.5: Настройки Burp Suite

Во вкладке Proxy устанавливаю “Intercept is on” (рис. [fig:006?]).

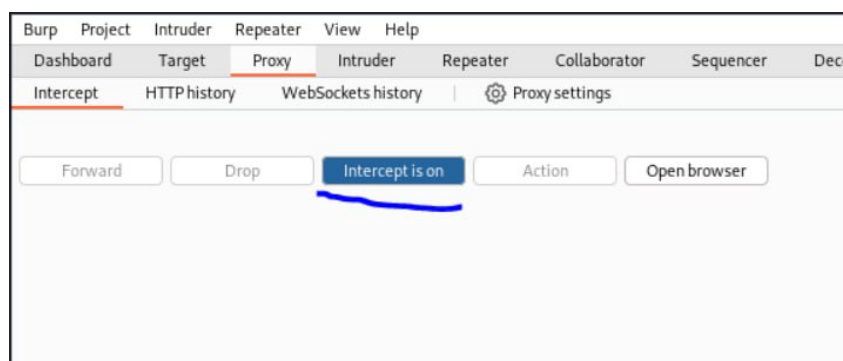


Рис. 3.6: Настройки Proxy

Чтобы Burp Suite исправно работал с локальным сервером, необходимо установить параметр `network_allow_hijacking_localhost` на `true` (рис. [fig:007?]).

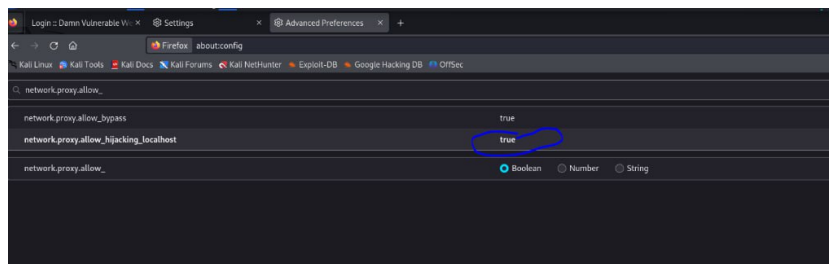


Рис. 3.7: Настройки параметров

Пытаюсь зайти в браузере на DVWA, тут же во вкладки Проху появляется захваченный запрос. Нажимаем “Forward”, чтобы загрузить страницу (рис. [fig:008?]).

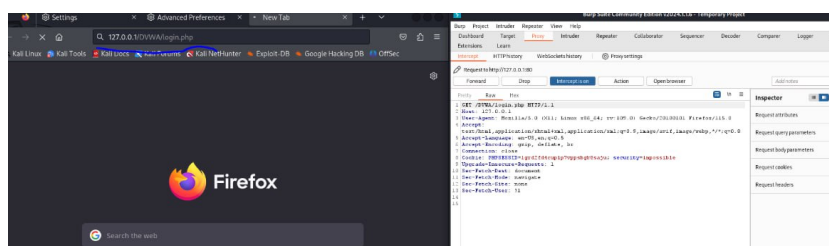


Рис. 3.8: Получаемые запросы сервера

Загрузилась страница авторизации, текст запроса поменялся (рис. [fig:009?]).

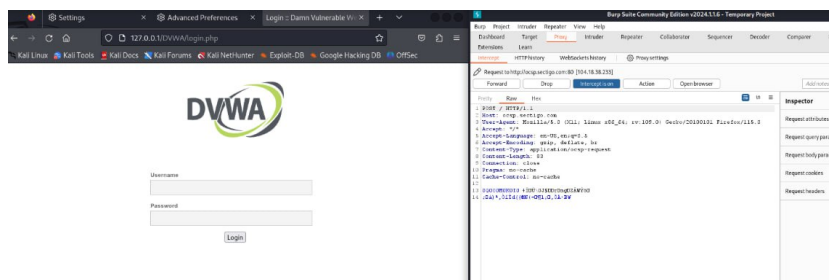


Рис. 3.9: Страница авторизации

История запросов хранится во вкладке Target (рис. [fig:010?]).

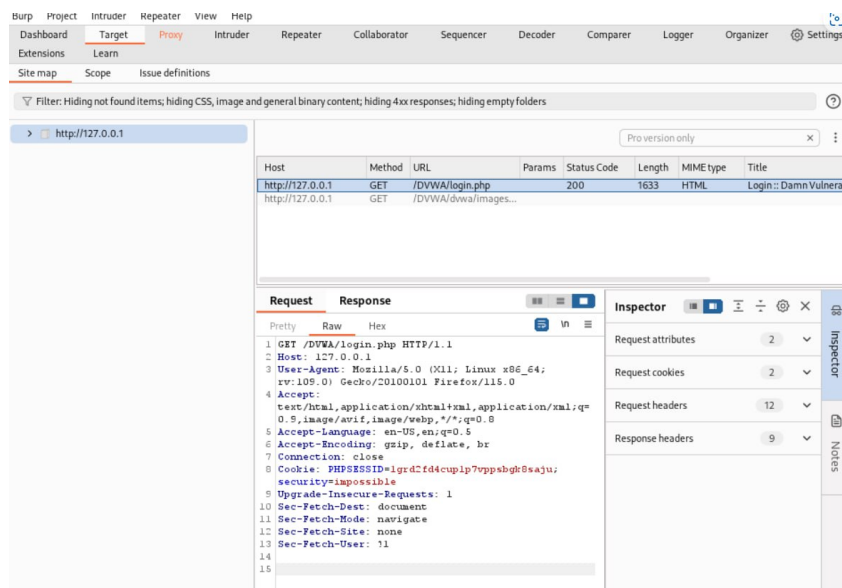


Рис. 3.10: История запросов

Попробуем 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. [fig:011?]).

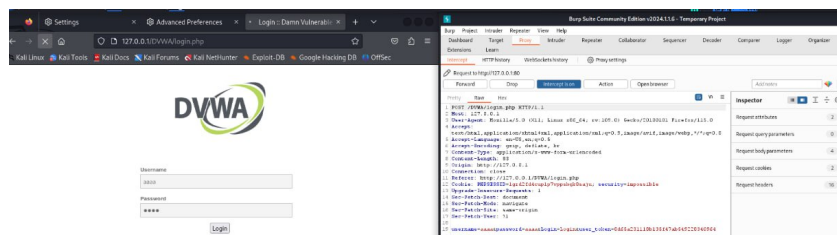


Рис. 3.11: Ввод случайных данных

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. [fig:012?]).

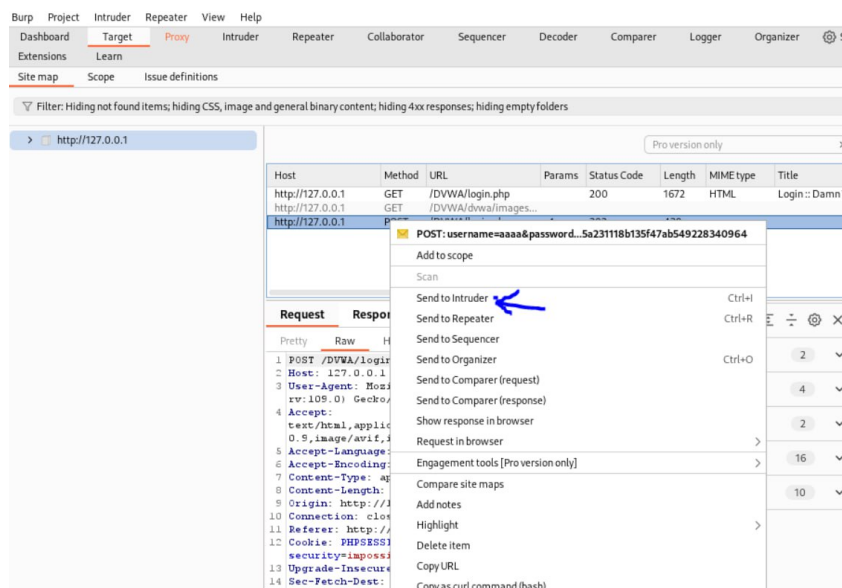


Рис. 3.12: POST-запрос с вводом пароля и логина

Попадаем на вкладку Intruder, видим значения по умолчанию у типа атаки и наш запрос (рис. [fig:013?]).

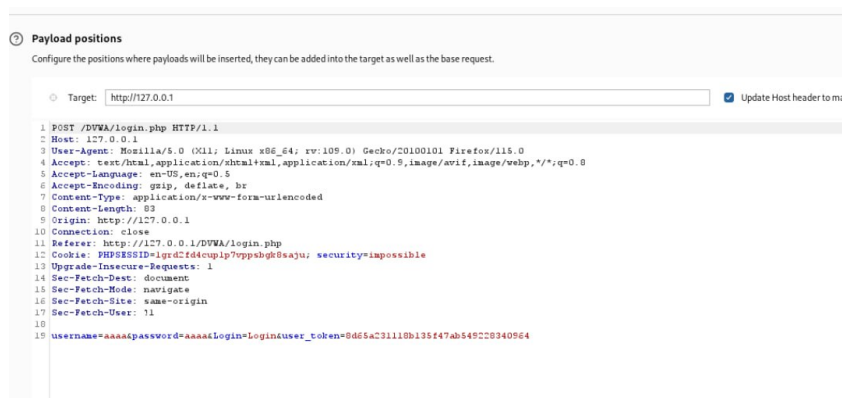


Рис. 3.13: Вкладка Intruder

Изменяем значение типа атаки на Cluster bomb и проставляем специальные символы у тех данных в форме для ввода, которые будем пробивать, то есть у имени пользователя и пароля (рис. [fig:014?]).



Рис. 3.14: Изменение типа атаки

Так как мы отметили два параметра для подбора, то нам нужно два списка со значениями для подбора. Заполняем первый список в Payload setting (рис. [fig:015?]).

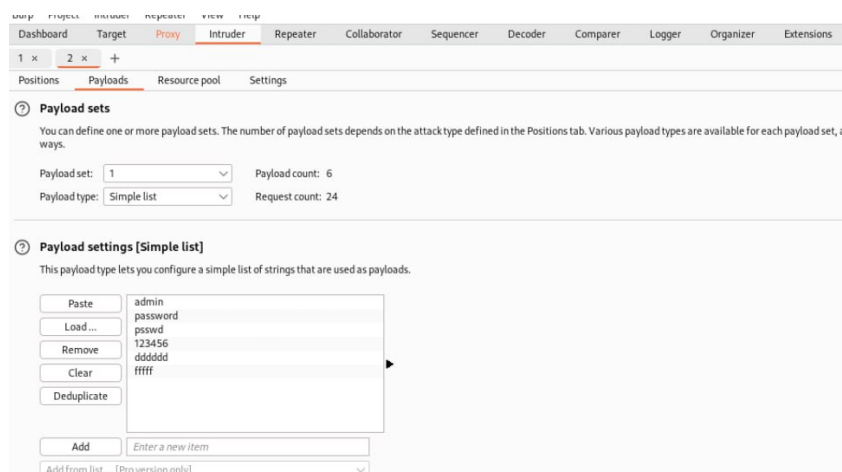


Рис. 3.15: Первый Simple list

Переключаемся на второй список и добавляем значения в него. В строке request count видим нужное количество запросов, чтобы проверить все возможные пары пользователь-пароль (рис. [fig:016?]).

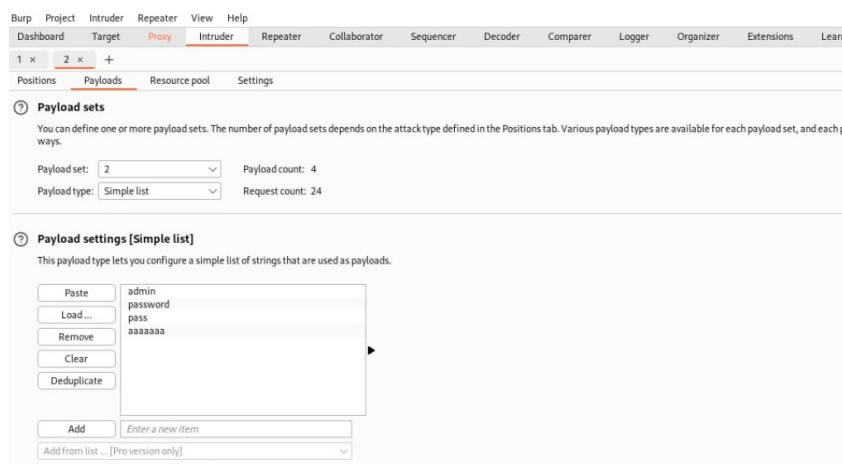


Рис. 3.16: Второй Simple list

Запускаю атаку и начинаю подбор (рис. [fig:017?]).

Attack Save							
2. Intruder attack of http://127.0.0.1							
Results Positions Payloads Resource pool Settings							
Filter: Showing all items							
Request	Payload1	Payload2	Status code	Response received	Error	Timeout	Length
0			302	7			476
1	admin	admin	302	18			475
2	password	admin	302	3			476
3	pass	admin	302	4			475
4	123456	admin	302	2			476
5	000000	admin	302	5			475
6	!!!!	admin	302	5			476
7	admin	password	302	3			475
8	password	password	302	6			476
9	pass	password	302	6			475
10	123456	password	302	2			476
11	000000	password	302	3			475
12	!!!!	password	302	3			476
13	admin	pass	302	2			475
14	password	pass	302	2			476
15	pass	pass	302	3			475
16	123456	pass	302	3			476
17	000000	pass	302	2			475
18	!!!!	pass	302	2			476
19	admin	aaaaaaa	302	10			475
20	password	aaaaaaa	302	9			476
21	pass	aaaaaaa	302	7			475
22	123456	aaaaaaa	302	5			476
23	000000	aaaaaaa	302	4			475
24	!!!!	aaaaaaa	302	7			476

Рис. 3.17: Запуск атаки

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 admin-admin нас перенаправило на login.php, это значит, что пара не подходит (рис. [fig:018?]).

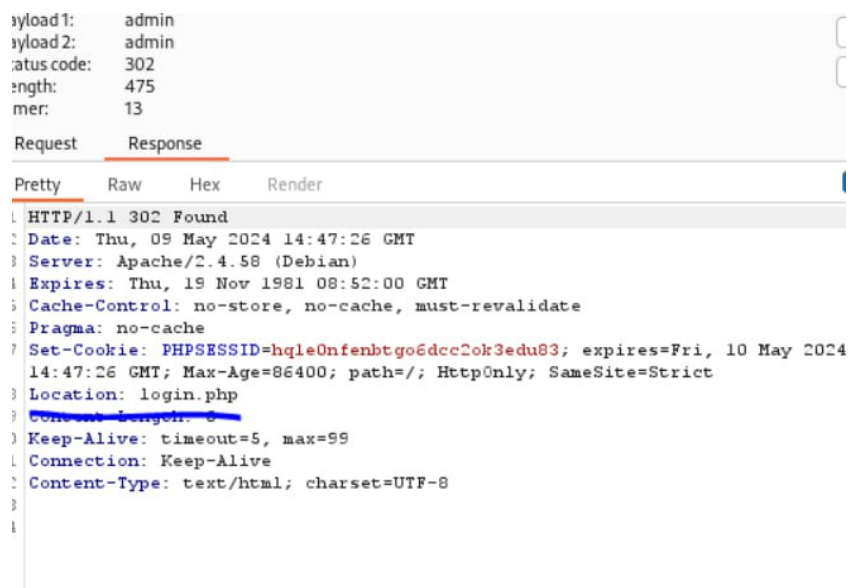


Рис. 3.18: Результат запроса

Проверим результат пары admin-password во вкладке Response, теперь нас перенаправляет на страницу index.php, значит пара должна быть верной (рис. [fig:019?]).

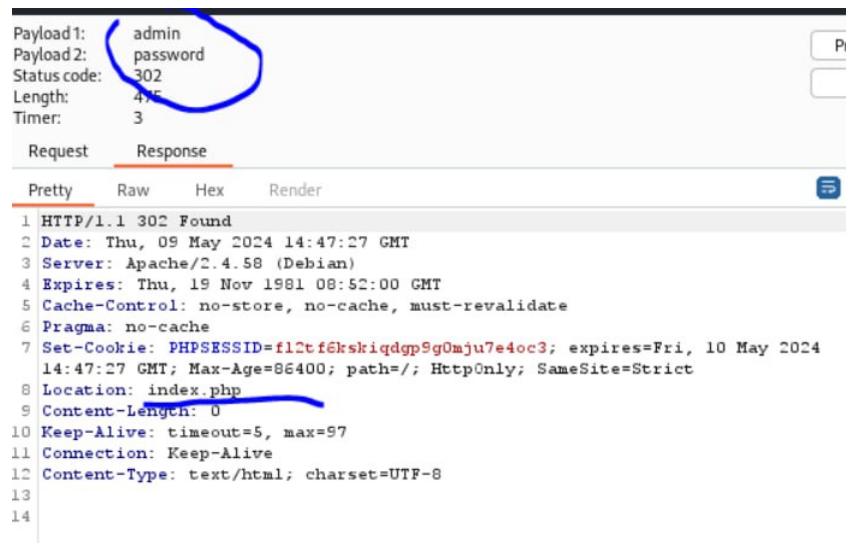
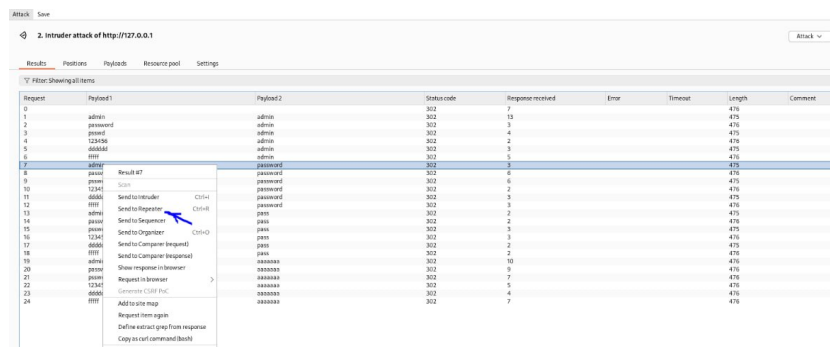


Рис. 3.19: Результат запроса

Дополнительная проверка с использованием Repeater, нажимаем на нужный

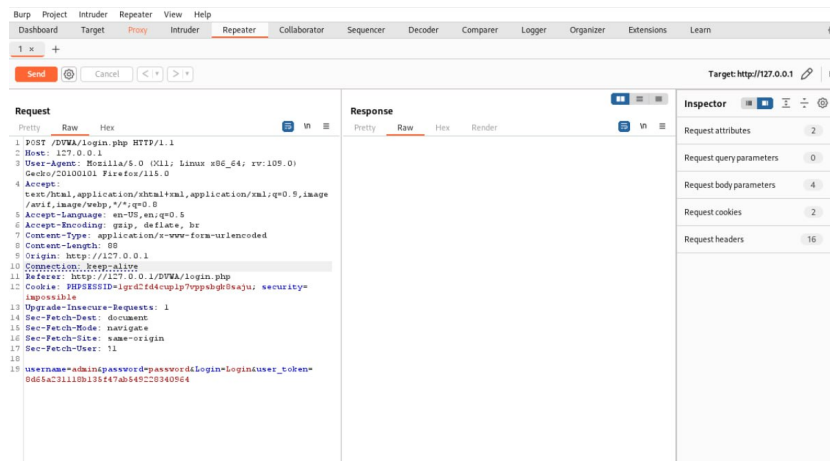
нам запрос правой кнопкой мыши и жмем “Send to Repeater” (рис. [fig:020?]).



Request	Payload1	Payload2	Status code	Response received	Error	Timeout	Length	Comment
0			302	7			476	
1	admin	admin	302	13			476	
2	password	admin	302	3			476	
3	passwd	admin	302	4			476	
4	123456	admin	302	2			476	
5	daaaaa	admin	302	3			476	
6	!!!!	admin	302	5			476	
7	admin	password	302	3			476	
8	passwd	password	302	6			476	
9	12345	password	302	2			476	
10	!!!!	password	302	3			476	
11	daaaa	Send to Intruder	302	3			476	
12	!!!!	Send to Repeater	302	3			476	
13	admin	Send to Sequencer	302	2			476	
14	passwd	Send to Sequencer	302	2			476	
15	passwd	Send to Organizer	302	3			476	
16	12345	Send to Sequencer	302	3			476	
17	daaaa	Send to Comparer (request)	302	2			476	
18	!!!!	Send to Comparer (request)	302	3			476	
19	admin	Show response in browser	302	10			476	
20	passwd	Show response in browser	302	9			476	
21	passwd	Request in browser	302	7			476	
22	12345	Compare HTTP Post	302	5			476	
23	daaaa	Add to map	302	4			476	
24	!!!!	Request item again	302	7			476	

Рис. 3.20: Дополнительная проверка результата

Переходим во вкладку “Repeater” (рис. [fig:021?]).



Request	Response
1 POST /DVWA/login.php HTTP/1.1 2 Host: 127.0.0.1 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: 80 9 Origin: http://127.0.0.1 10 Connection: keep-alive 11 Referer: http://127.0.0.1/DVWA/login.php 12 Cookie: PHPSESSID=lgdCf64cuplp7ppabgh8aju; security=Impossible 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: 11 18 19 username=admin&password=password&login=loginuser_token=0d65a23118b136f47ab449228340564	2020 OK 21 Location: http://127.0.0.1/index.php 22 Content-Type: text/html; charset=UTF-8 23 Content-Length: 1433 24 Date: Mon, 15 May 2022 12:00:00 GMT 25 Server: Apache/2.4.18 (Ubuntu)

Рис. 3.21: Вкладка Repeater

Нажимаем “send”, получаем в Response в результат перенаправление на index.php (рис. [fig:022?]).

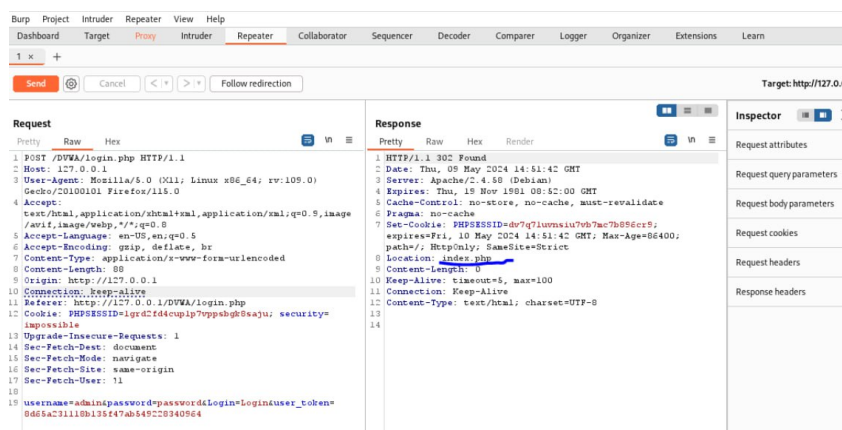


Рис. 3.22: Окно Response

После нажатия на Follow redirection, получим неcompiled html код в окне Response (рис. [fig:023?]).

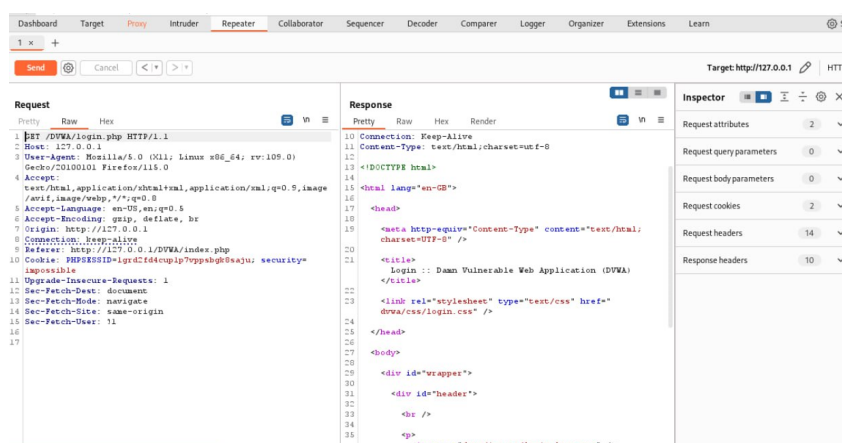


Рис. 3.23: Изменение в окне Response

Далее в подокне Render получим то, как выглядит полученная страница (рис. [fig:024?]).

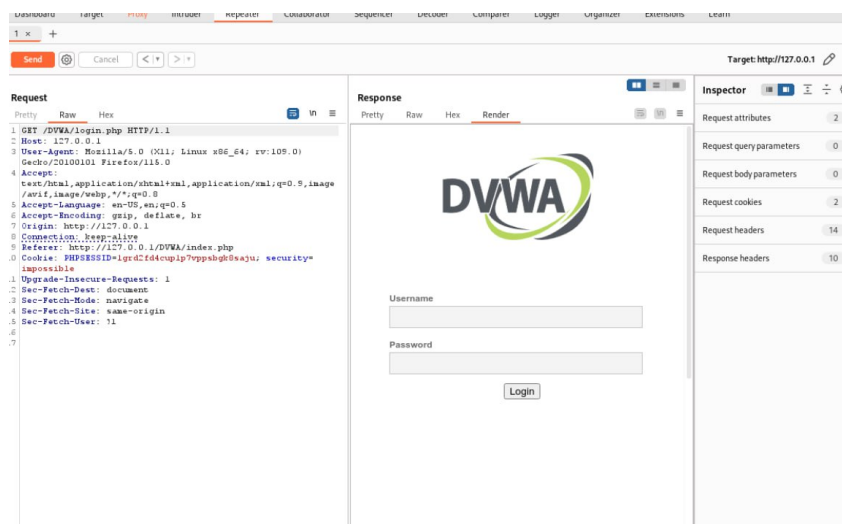


Рис. 3.24: Полученная страница

4 Выводы

При выполнении лабораторной работы я научился использовать инструмент Burp Suite.

Список литературы

1. Ш. Парасрам Т.Х.и.др. А. Замм. Kali Linux: Тестирование на проникновение и безопасность: для профессионалов. Питер, 2022. 448 с.