

Отчет по выполнению индивидуального проекта. Этап №5

Основы информационной безопасности

Закиров Н.Д.

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	18
	Список литературы	19

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	9
3.8	Получаемые запросы сервера	10
3.9	Страница авторизации	10
3.10	История запросов	10
3.11	Ввод случайных данных	11
3.12	POST-запрос с вводом пароля и логина	11
3.13	Вкладка Intruder	12
3.14	Изменение типа атаки	12
3.15	Первый Simple list	13
3.16	Второй Simple list	13
3.17	Запуск атаки	14
3.18	Результат запроса	14
3.19	Результат запроса	15
3.20	Дополнительная проверка результата	15
3.21	Вкладка Repeater	16
3.22	Окно Response	16
3.23	Изменение в окне Response	17
3.24	Полученная страница	17

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.

3 Выполнение лабораторной работы

Запускаю локальный сервер, на котором открою веб-приложение DVWA для тестирования инструмента Burp Suite (рис. 3.1).

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

Рис. 3.1: Запуск локального сервера

Запускаю инструмент Burp Suite (рис. 3.2).

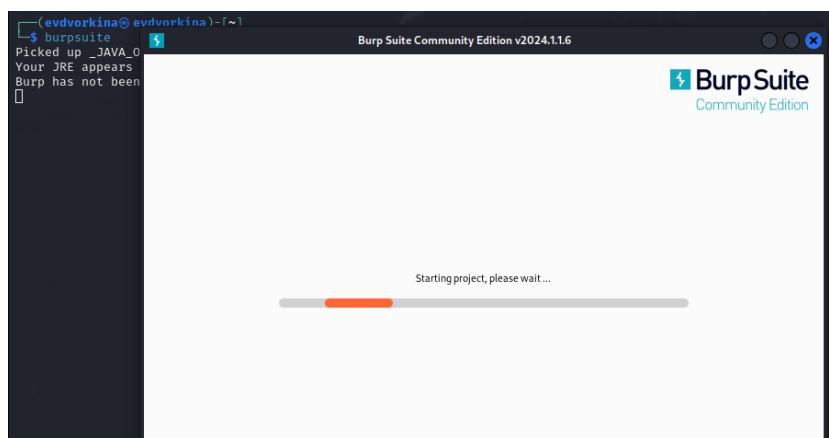


Рис. 3.2: Запуск приложения

Открываю сетевые настройки браузера, для подготовке к работе (рис. 3.3).

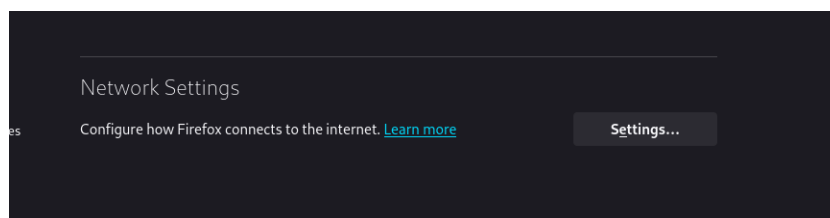


Рис. 3.3: Сетевые настройки браузера

Изменение настроек сервера для работы с проху и захватом данных с помощью Burp Suite (рис. 3.4).

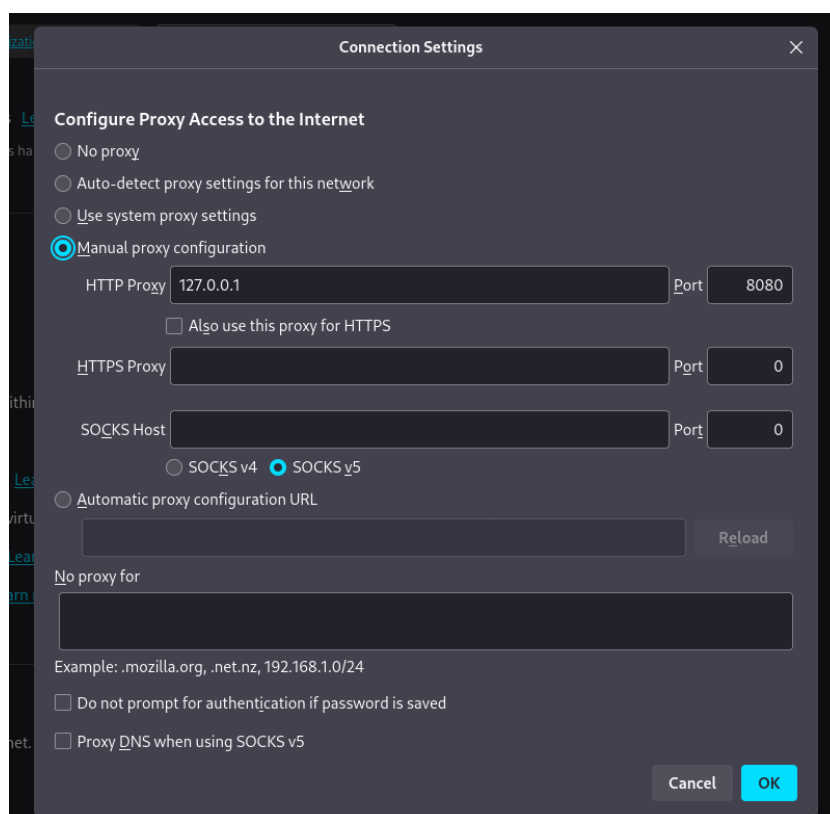


Рис. 3.4: Настройки сервера

Изменяю настройки Проху инструмента Burp Suite для дальнейшей работы (рис. 3.5).

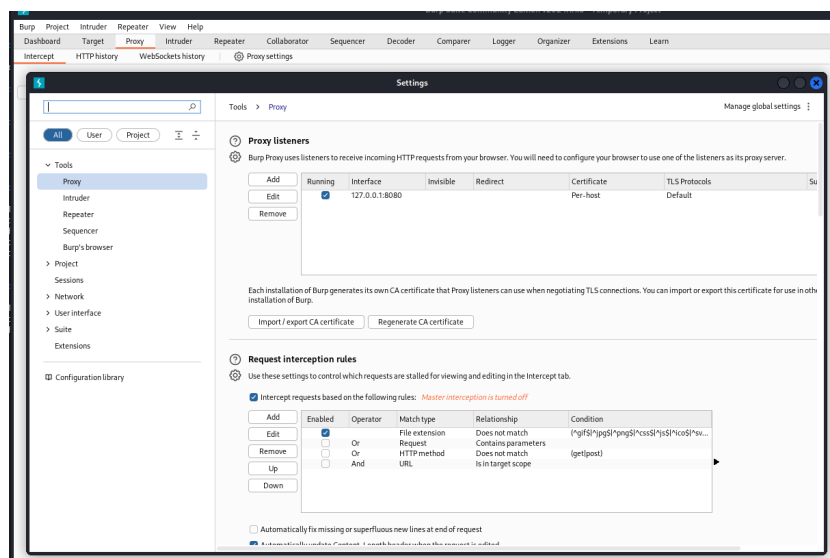


Рис. 3.5: Настройки Burp Suite

Во вкладке Проxy устанавливаю “Intercept is on” (рис. 3.6).

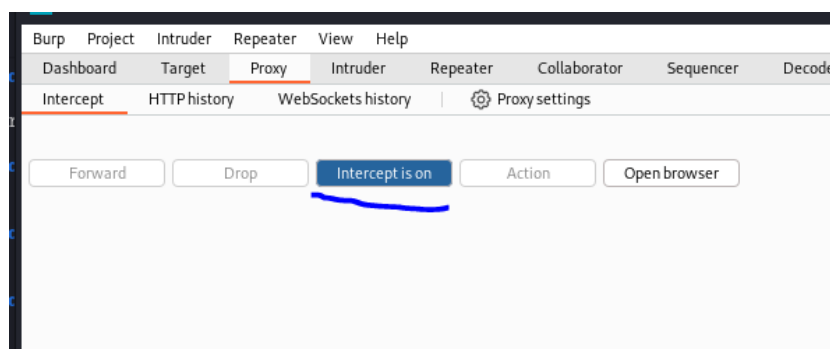


Рис. 3.6: Настройки Proxy

Чтобы Burp Suite исправно работал с локальным сервером, необходимо установить параметр `network_allow_hijacking_localhost` на `true` (рис. 3.7).

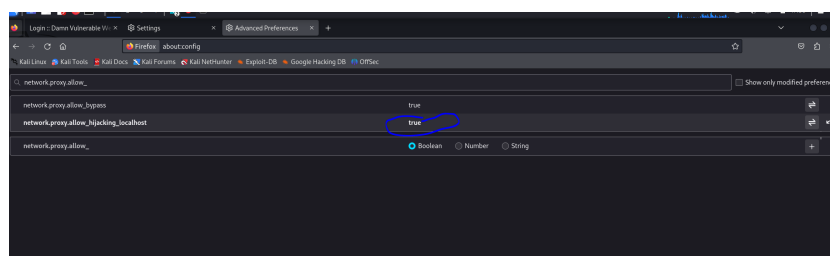


Рис. 3.7: Настройки параметров

Пытаюсь зайти в браузере на DVWA, тут же во вкладки Проху появляется захваченный запрос. Нажимаем “Forward”, чтобы загрузить страницу (рис. 3.8).

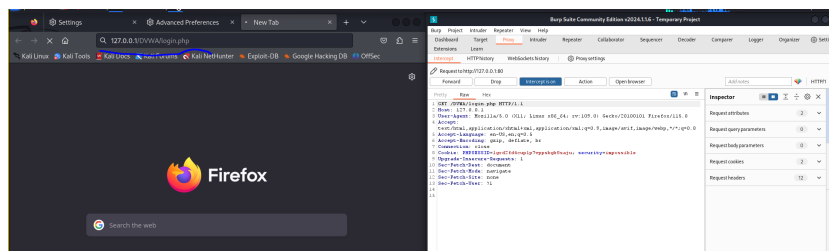


Рис. 3.8: Получаемые запросы сервера

Загрузилась страница авторизации, текст запроса поменялся (рис. 3.9).

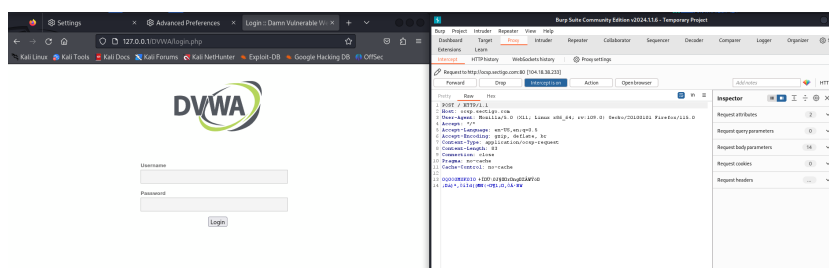


Рис. 3.9: Страница авторизации

История запросов хранится во вкладке Target (рис. 3.10).

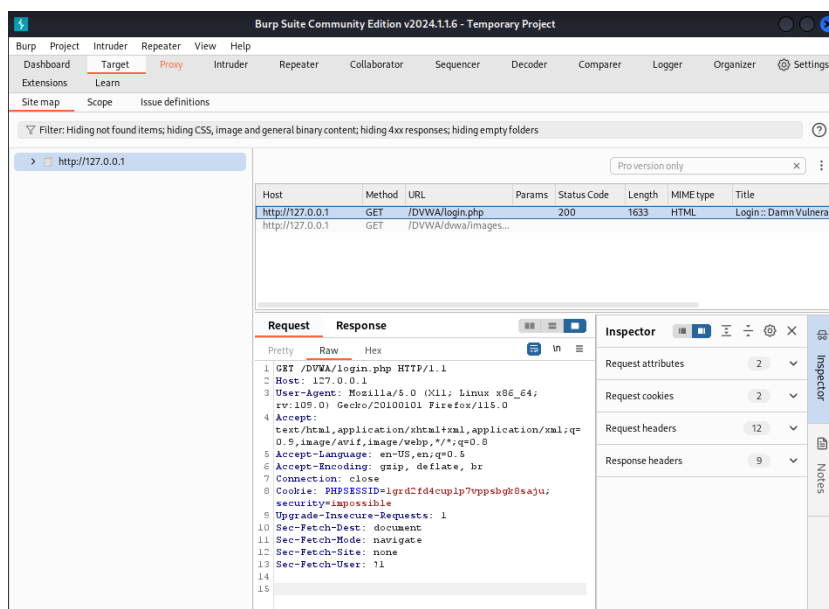


Рис. 3.10: История запросов

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. 3.11).

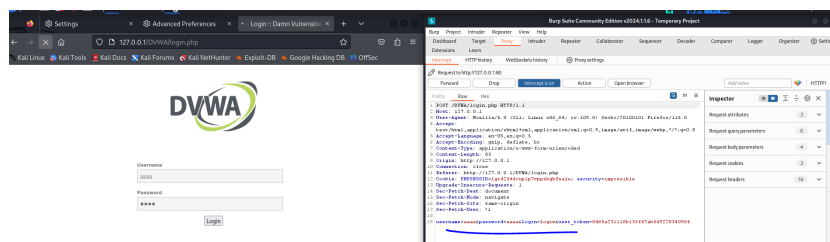


Рис. 3.11: Ввод случайных данных

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. 3.12).

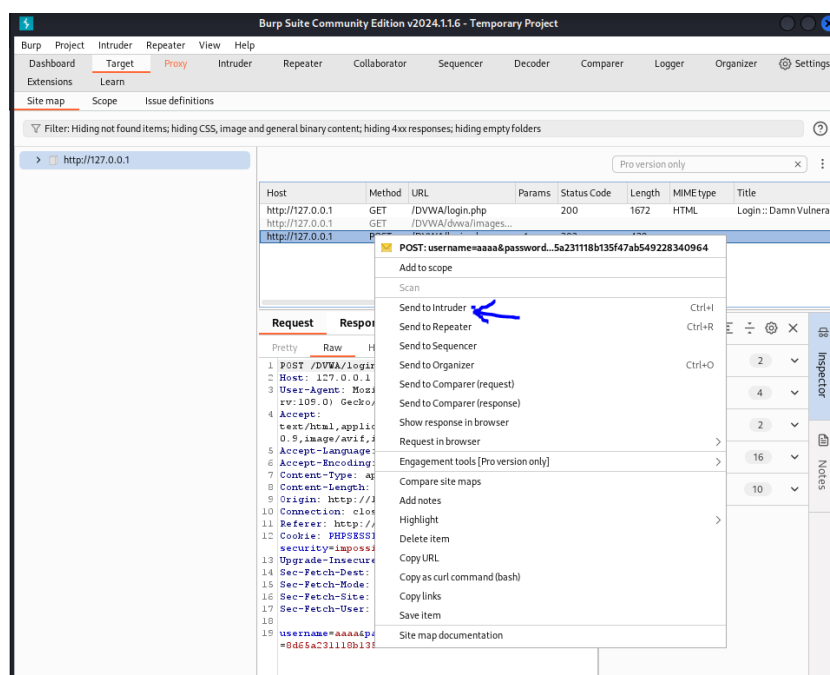


Рис. 3.12: POST-запрос с вводом пароля и логина

Попадаем на вкладку Intruder, видим значения по умолчанию у типа атаки и наш запрос (рис. 3.13).

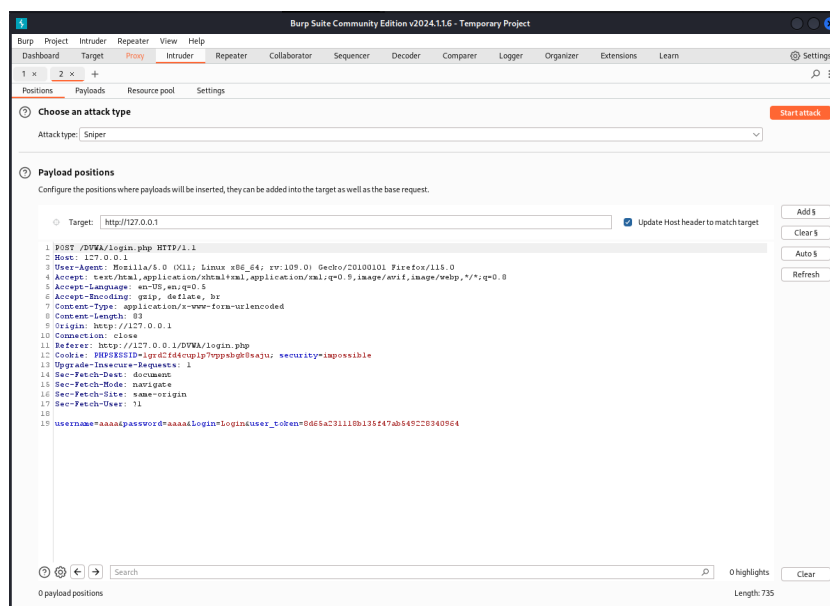


Рис. 3.13: Вкладка Intruder

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. 3.14).

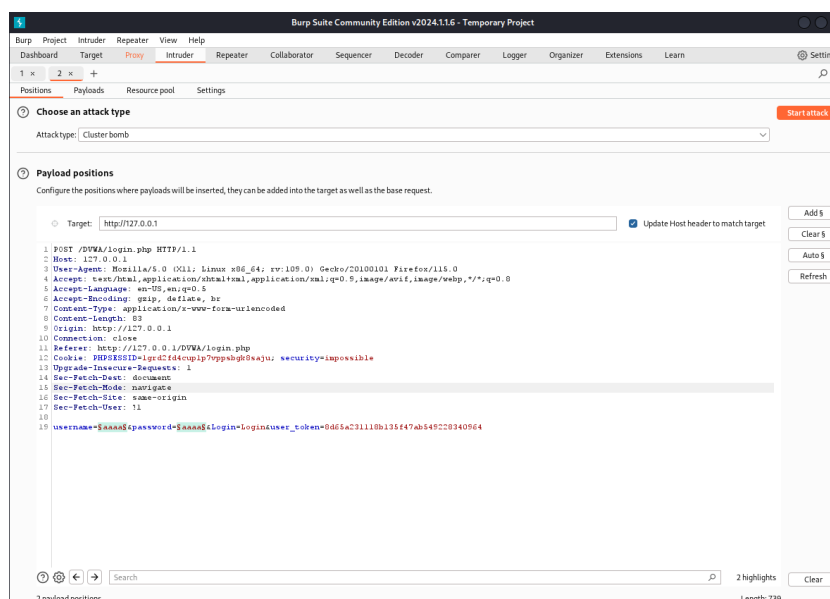


Рис. 3.14: Изменение типа атаки

Так как мы отметили два параметра для подбора, то нам нужно два списка

со значениями для подбора. Заполняем первый список в Payload setting (рис. 3.15).

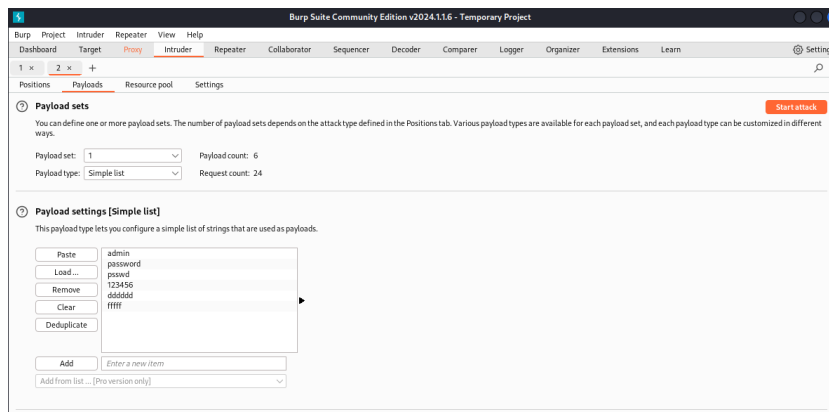


Рис. 3.15: Первый Simple list

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. 3.16).

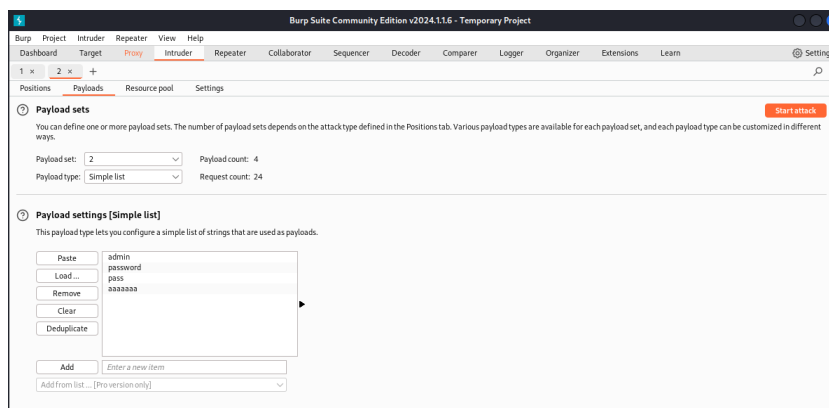


Рис. 3.16: Второй Simple list

Запускаю атаку и начинаю подбор (рис. 3.17).

Request	Payload1	Payload2	Status code	Response received	Error	Timeout	Length	Comment
0			302	7			475	
1	admin	admin	302	13			475	
2	password	password	302	3			475	
3	password	admin	302	4			475	
4	123456	admin	302	3			475	
5	password	password	302	3			475	
6	HTTP	admin	302	5			475	
7	admin	password	302	5			475	
8	password	password	302	6			475	
9	password	password	302	2			475	
10	123456	password	302	3			475	
11	password	password	302	3			475	
12	HTTP	password	302	3			475	
13	admin	password	302	2			475	
14	password	password	302	2			475	
15	password	password	302	3			475	
16	123456	password	302	3			475	
17	password	password	302	3			475	
18	HTTP	password	302	2			475	
19	admin	password	302	9			475	
20	password	password	302	7			475	
21	password	password	302	5			475	
22	123456	password	302	4			475	
23	password	password	302	7			475	
24	HTTP	password	302				475	

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. 3.18).

Request	Response
Payload 1: admin Payload 2: admin Status code: 302 Length: 475 Timer: 13	<pre> 1 HTTP/1.1 302 Found 2 Date: Thu, 09 May 2024 14:47:26 GMT 3 Server: Apache/2.4.58 (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=hq1e0nfenbtgo6dccc2ok3edu83; expires=Fri, 10 May 2024 14:47:26 GMT; Max-Age=86400; path=/; HttpOnly; SameSite=Strict 8 Location: login.php 9 Content-Type: text/html; charset=UTF-8 10 Keep-Alive: timeout=5, max=99 11 Connection: Keep-Alive 12 Content-Type: text/html; charset=UTF-8 13 14 </pre>

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.

3.19).

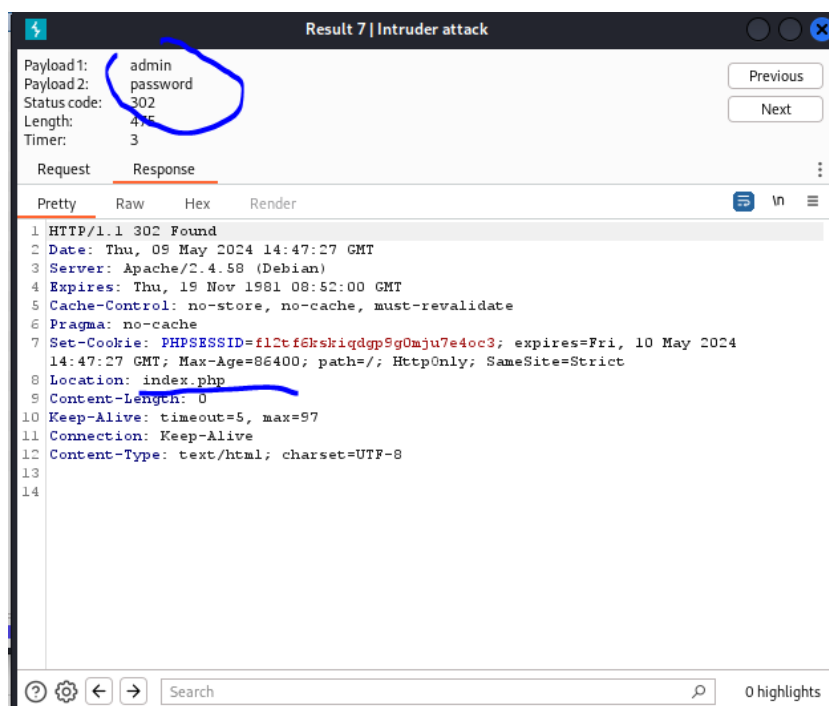


Рис. 3.19: Результат запроса

Дополнительная проверка с использованием Repeater, нажимаем на нужный нам запрос правой кнопкой мыши и жмем “Send to Repeater” (рис. 3.20).

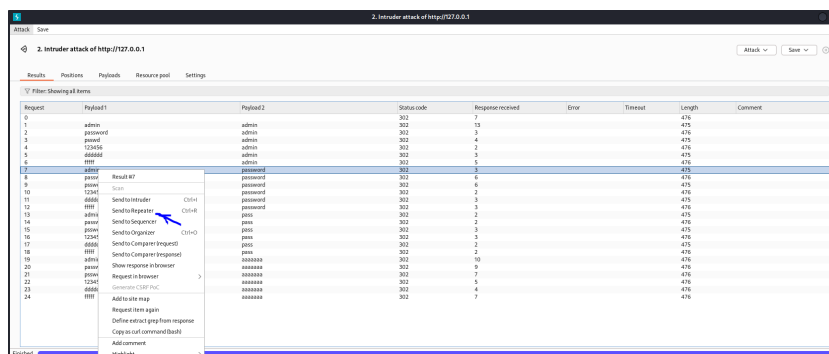


Рис. 3.20: Дополнительная проверка результата

Переходим во вкладку “Repeater” (рис. 3.21).

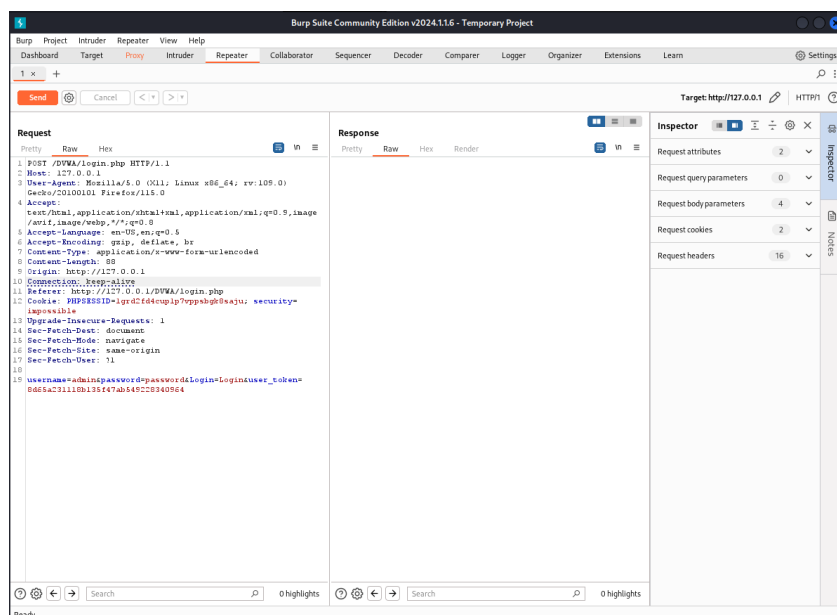


Рис. 3.21: Вкладка Repeater

Нажимаем “send”, получаем в Response в результат перенаправление на index.php (рис. 3.22).

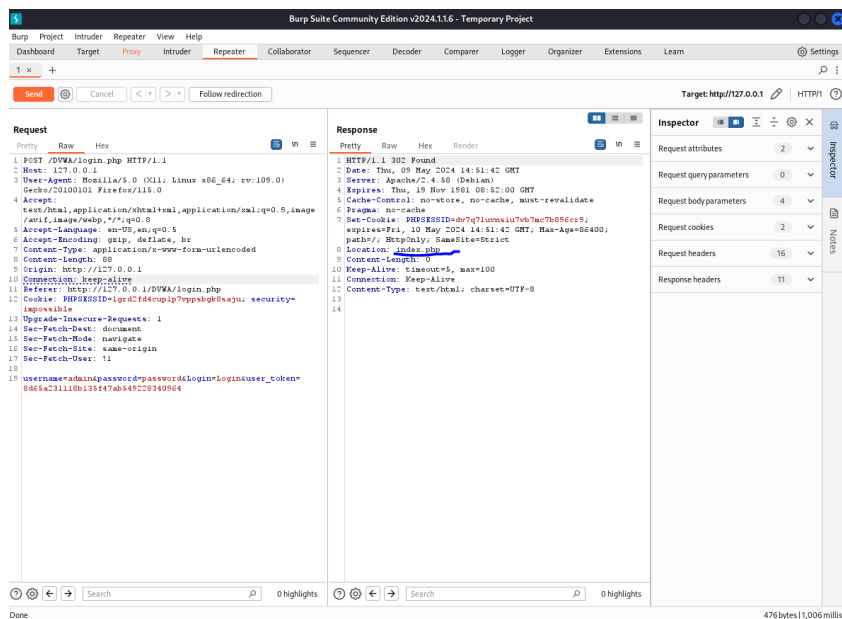


Рис. 3.22: Окно Response

После нажатия на Follow redirection, получим неcompiled html код в окне Response (рис. 3.23).

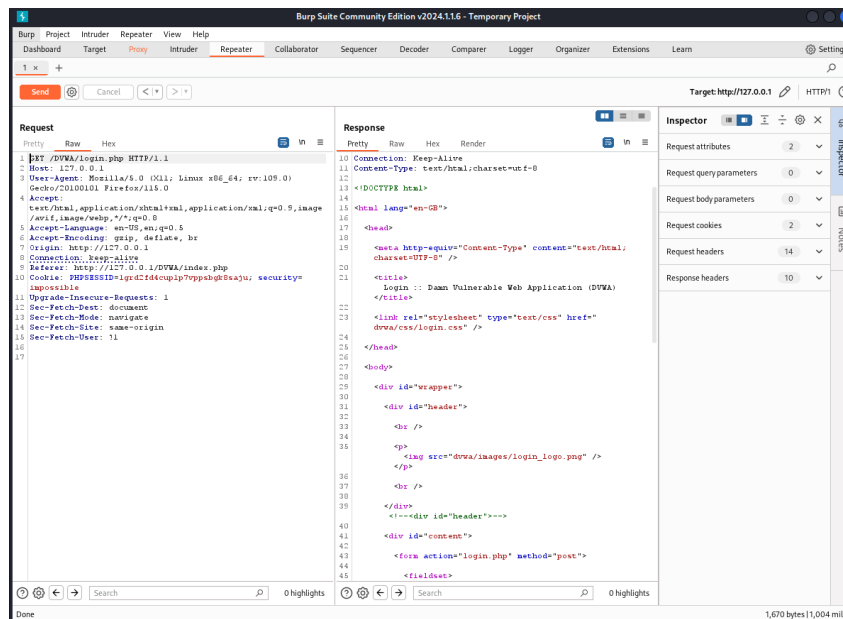


Рис. 3.23: Изменение в окне Response

Далее в подокне Render получим то, как выглядит полученная страница (рис. 3.24).

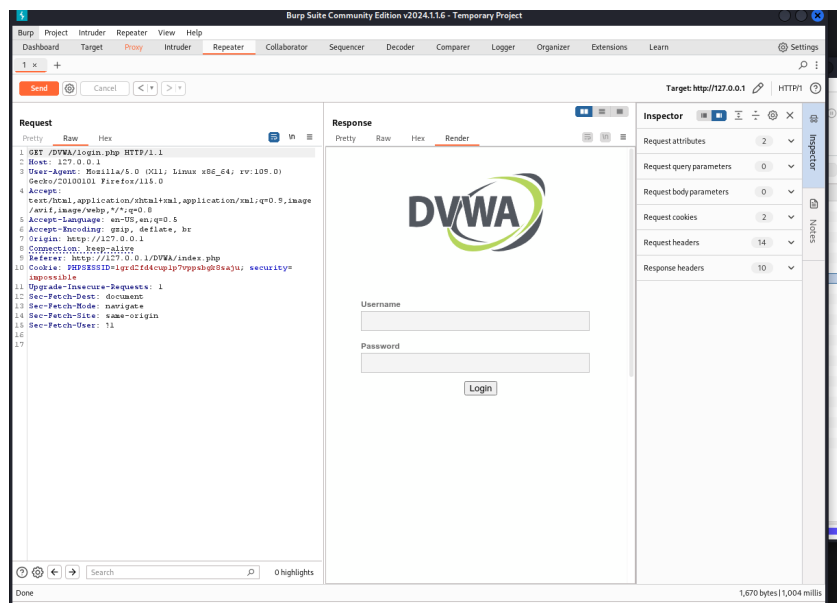


Рис. 3.24: Полученная страница

4 Выводы

При выполнении лабораторной работы научился использовать инструмент Burp Suite.

Список литературы