

Презентация по пятому этапу индивидуального проекта

Информационная безопасность

Коняева Марина Александровна

НФИбд-01-21

Студ. билет: 1032217044

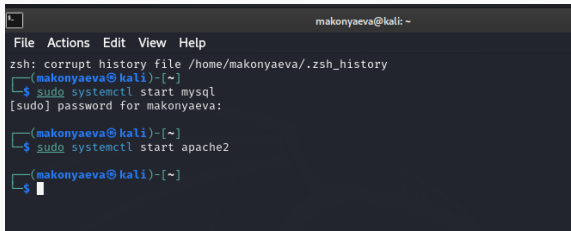
2024

RUDN

Научиться использовать Burp Suite.

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. [@parasram].

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



```
makonyaeva@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/makonyaeva/.zsh_history  
(makonyaeva@kali)~  
$ sudo systemctl start mysql  
[sudo] password for makonyaeva:  
(makonyaeva@kali)~  
$ sudo systemctl start apache2  
(makonyaeva@kali)~  
$
```

Рис. 1: Запуск локального сервера

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

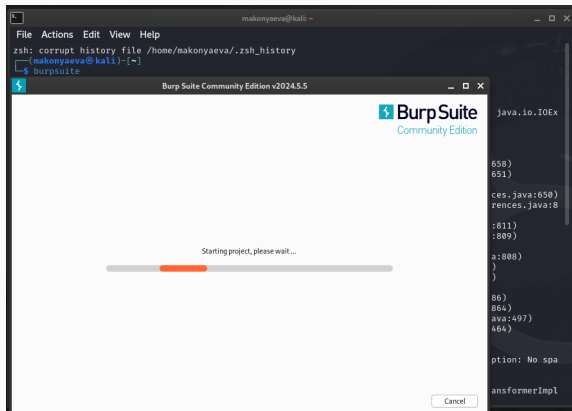


Рис. 2: Запуск приложения

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

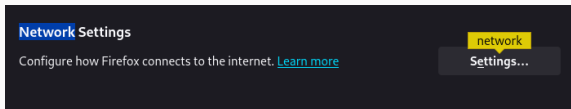
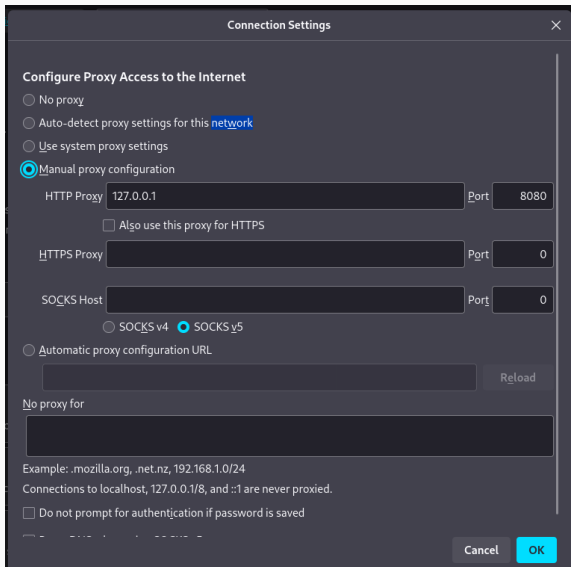


Рис. 3: Сетевые настройки браузера

Выполнение лабораторной работы

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



Выполнение лабораторной работы

Изменяю настройки Прoxy инструмента Burp Suite для дальнейшей работы (рис. [-@fig:005]).

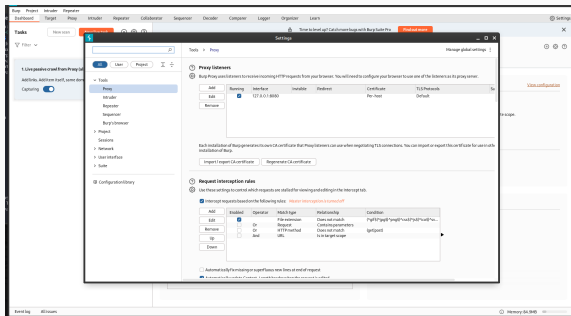


Рис. 5: Настройки Burp Suite

Во вкладке Proxu устанавливаю “Intercept is on” (рис. [-@fig:006]).

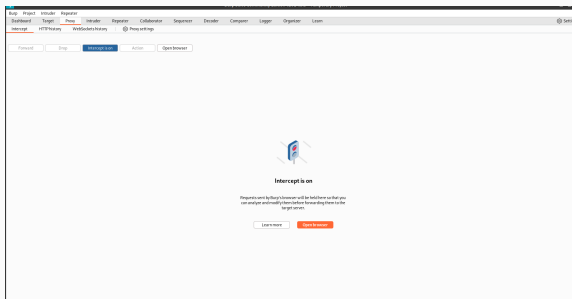


Рис. 6: Настройки Proxu

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

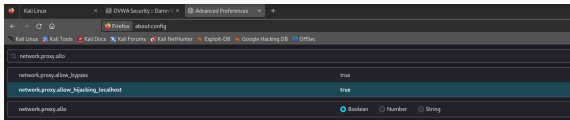


Рис. 7: Настройки параметров

Выполнение лабораторной работы

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

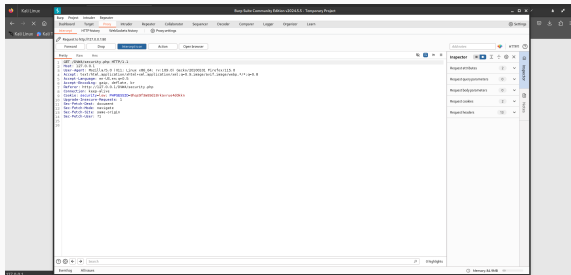


Рис. 8: Получаемые запросы сервера

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



Рис. 9: Страница авторизации

Выполнение лабораторной работы

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

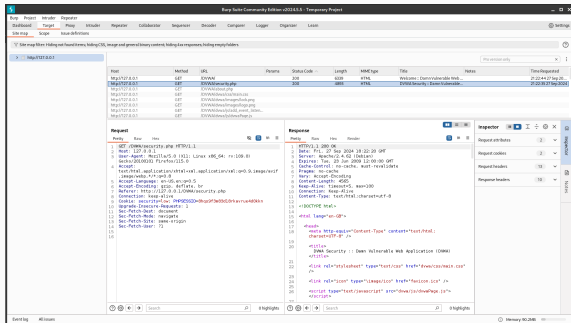


Рис. 10: История запросов

Выполнение лабораторной работы

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]).

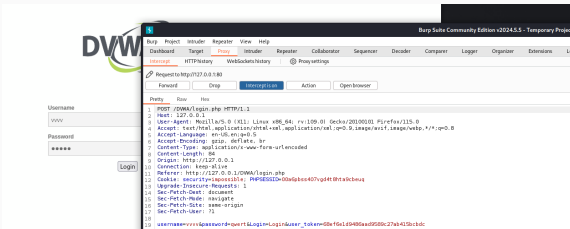


Рис. 11: Ввод случайных данных

Выполнение лабораторной работы

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]).

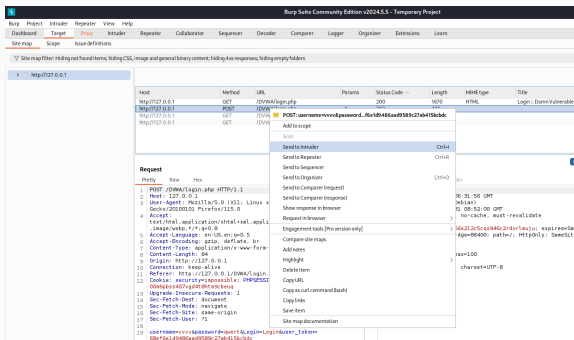


Рис. 12: POST-запрос с вводом пароля и логина

Выполнение лабораторной работы

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

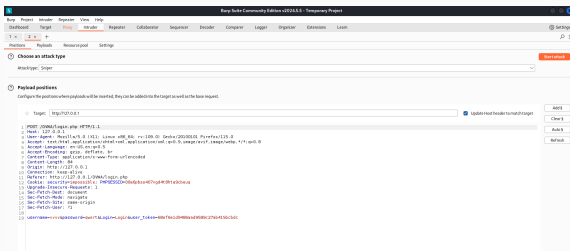


Рис. 13: Вкладка Intruder

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]).

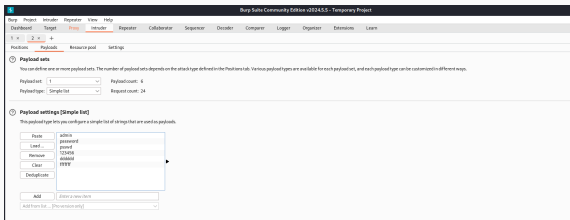


Рис. 15: Первый Simple list

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]).

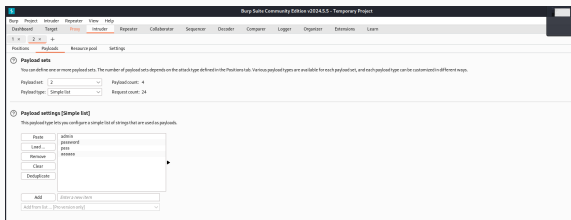
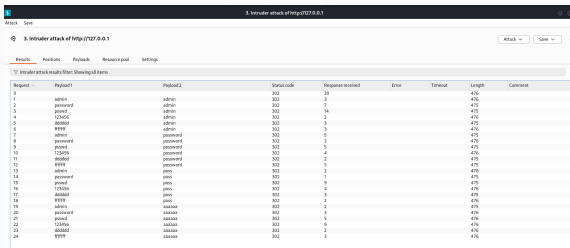


Рис. 16: Второй Simple list

Выполнение лабораторной работы

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



Request	Payload1	Payload2	Status code	Response received	Error	Timeout	Length	Content
0			302	28			476	
1	admin	admin	302	5			476	
2	password	admin	302	7			475	
3	pownd	admin	302	14			475	
4	123456	admin	302	2			476	
5	ababab	admin	302	3			475	
6	9999	admin	302	3			476	
7	admin	password	302	5			475	
8	password	password	302	5			476	
9	pownd	password	302	5			475	
10	123456	password	302	4			476	
11	ababab	password	302	2			475	
12	9999	password	302	5			475	
13	admin	perm	302	2			476	
14	password	perm	302	1			475	
15	pownd	perm	302	9			475	
16	123456	perm	302	4			476	
17	ababab	perm	302	5			475	
18	9999	perm	302	2			476	
19	admin	aaaaaa	302	2			475	
20	password	aaaaaa	302	9			476	
21	pownd	aaaaaa	302	6			476	
22	123456	aaaaaa	302	9			476	
23	ababab	aaaaaa	302	2			476	
24	9999	aaaaaa	302	3			476	

Рис. 17: 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 admin-admin нас перенаправило на login.php, это значит, что пара не подходит (рис. [-@fig:018]).

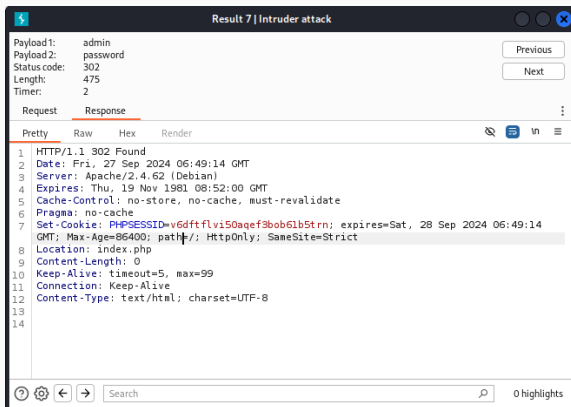


Рис. 18: Результат запроса

Выполнение лабораторной работы

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

Request	Payload1	Payload2	Status code	Response received	Error	Timeout	Length	Comment
0			302	4			476	
1	admin	admin	302	24			476	
2	password	admin	302	8			476	
3	password	admin	302	7			476	
4	123456	admin	302	2			476	
5	000000	admin	302	3			476	
6	99999	admin	302	1			476	
7	admin	password	302	2			476	
8	password	password	302	2			476	
9	123456	password	302	1			476	
10	000000	password	302	1			476	
11	99999	password	302	4			476	
12	admin	pass	302	2			476	
13	password	pass	302	3			476	
14	123456	pass	302	3			476	
15	000000	pass	302	3			476	
16	99999	pass	302	3			476	
17	admin	pass	302	2			476	
18	password	pass	302	2			476	
19	123456	pass	302	2			476	
20	000000	pass	302	2			476	
21	99999	pass	302	2			476	
22	admin	pass	302	4			476	
23	password	pass	302	2			476	
24	123456	pass	302	6			476	

Рис. 19: Результат запроса

Выполнение лабораторной работы

Дополнительная проверка с использованием Repeater, нажимаем на нужный нам запрос правой кнопкой мыши и жмем “Send to Repeater” (рис. [-@fig:020]).

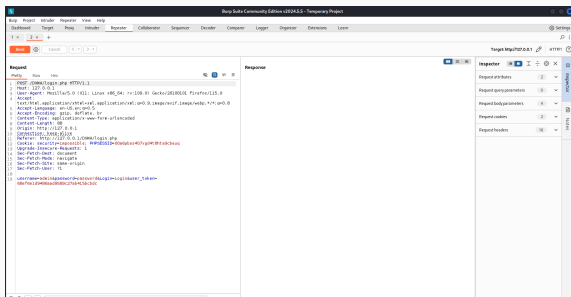


Рис. 20: Дополнительная проверка результата

Выполнение лабораторной работы

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

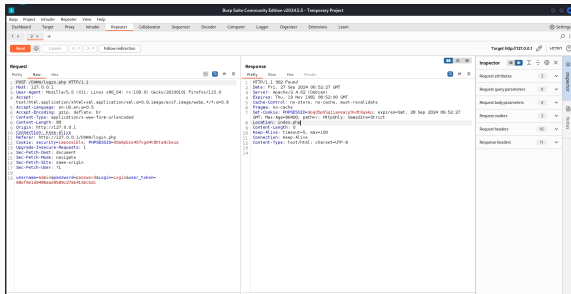


Рис. 21: Вкладка Repeater

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

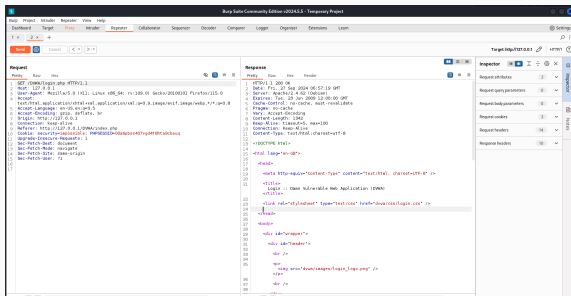


Рис. 22: Окно Response

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

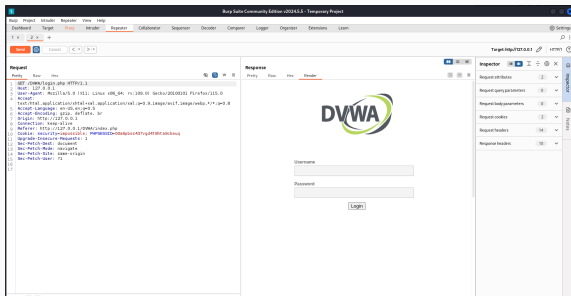


Рис. 23: Изменение в окне Response

Выполнение лабораторной работы

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

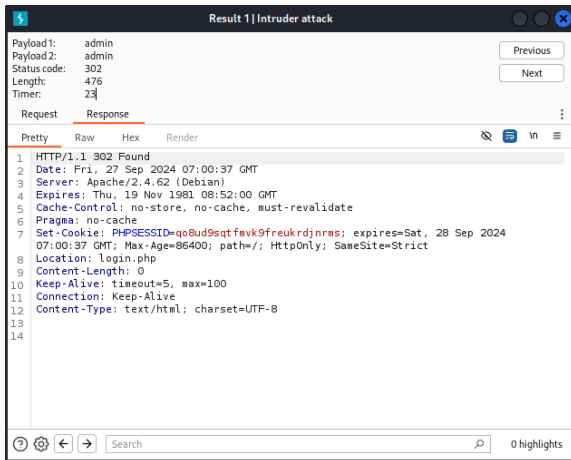


Рис. 24: Полученная страница

При выполнении лабораторной работы были приобретены навыки использования инструмента Burp Suite.

[1] Методические материалы курса.

[2] Linux Tool Documentation: Burp Suite (URL:
<https://portswigger.net/burp/documentation>)