

# Индивидуальный проект - Этап 5

## Основы информационной безопасности

---

Чистов Д. М.

10 мая 2025

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

## Цель работы

---

Получение навыков пользования Burp Suite.

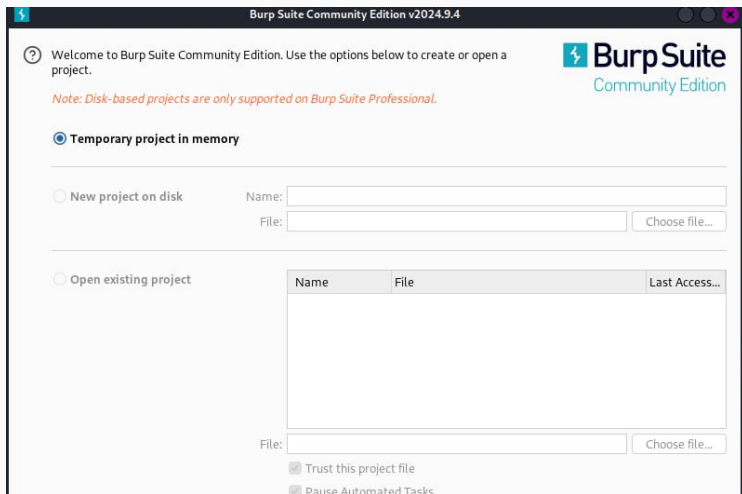
Burp Suite - инструмент для тестирования безопасности веб-приложений, позволяющий множеством функций перехватывать, анализировать, модифицировать разные HTTP-запросы между клиентом и сервером.

## Выполнение лабораторной работы

---

## Выполнение лабораторной работы

Запускаю Burp Suite, прохожу через пару диалоговых окон, где спрашивают, как будет устроен проект, над которым мы будем работать.



Открываю встроенный в Burp Suite браузер и открываю в нём DVWA - всё как обычно.

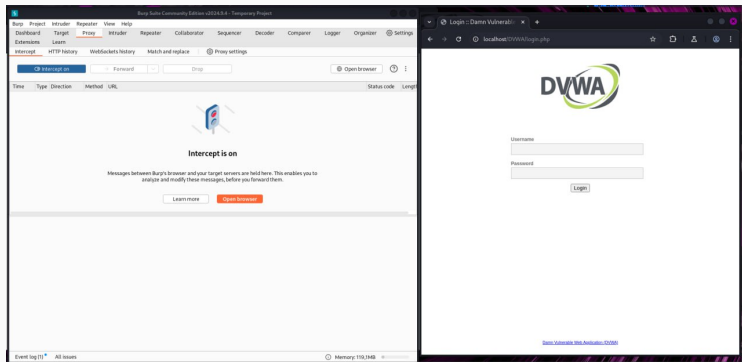
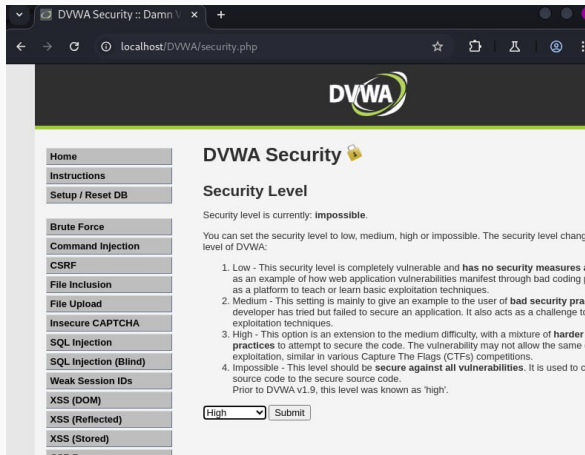


Рис. 2: DVWA через встроенный браузер

# Выполнение лабораторной работы

Перед работой надо запустить apache2 и mysql, буду тестировать Burp Suite на dvwa - брут форс пароля, как в этапе про Hydra, только в этот раз у DVWA будет уровень защиты “Высокий”.





# Выполнение лабораторной работы

Перехожу на страничку Brute Force DVWA, там есть кнопку view source, которая позволяет посмотреть код данной странички. Такая страничка различается на разных уровнях сложности - на уровне сложности High появляется user\_token, который совсем чуток усложняет брут форс.

## vulnerabilities/brute/source/high.php

```
<?php

if( isset( $ GET[ 'Login' ] ) ) {
    // Check Anti-CSRF token
    checkToken( $ REQUEST[ 'user token' ], $ SESSION[ 'session token' ], 'index.php' );

    // Sanitise username input
    $user = $ GET[ 'username' ];
    $user = stripslashes( $user );
    $user = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["_
MySQLConverterToo"] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : "");

    // Sanitise password input
    $pass = $ GET[ 'password' ];
    $pass = stripslashes( $pass );
    $pass = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["_
MySQLConverterToo"] Fix the mysql_escape_string() call! This code does not work.", E_USER_ERROR)) ? "" : "");
    $pass = md5( $pass );

    // Check database
    $query = "SELECT * FROM 'users' WHERE user = '$user' AND password = '$pass'";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query ) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])) ? mys

if( $result && mysqli_num_rows( $result ) == 1 ) {
    // Get users details
    $row = mysqli_fetch_assoc( $result );
    $avatar = $row["avatar"];

    // Login successful
    echo "<p>Welcome to the password protected area {user}</p>";
    echo "<img src=\"{$avatar}\" />";
}
else {
    // Login failed
```

При каждом обновлении страницы меняется и user\_token (а страница будет много обновляться при множестве неудачных попыток брут форса), сервер в свою очередь не пропускает реквесты, у которых уже устарел user\_token, т.е. взломщику нужно придумать способ, как этот user\_token получать автоматически при каждой попытке брут форса.

Идём далее, с помощью Burp Suite мы можем автоматизировать процесс нахождения user\_token (он вшит в страничку). Захожу в настройки и во вкладке Sessions создаю новое правило.

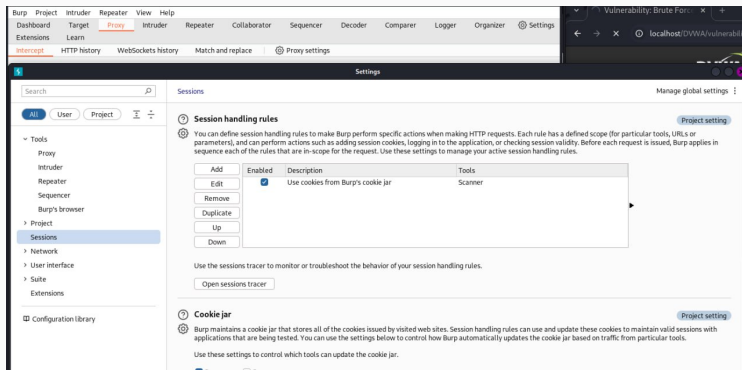


Рис. 5: Новое правило в Burp Suite

# Выполнение лабораторной работы

В новом правиле мы добавляем новое “макро действие”, и затем настраиваем его - открывается Macro Recorder, где мы выбираем наш последний реквест - попытку входа в DVWA, откуда мы можем посмотреть на наш реквест в виде кода и найти строки с user\_token.

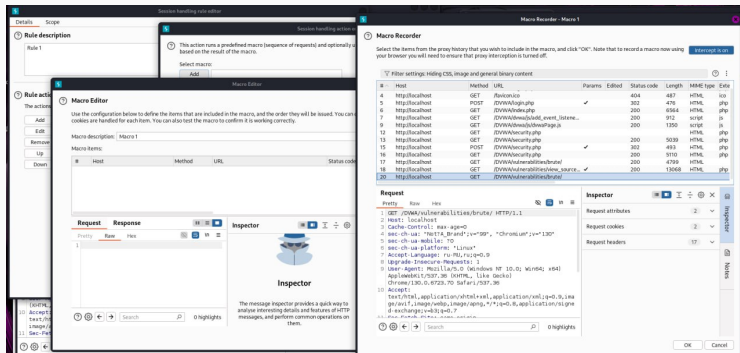


Рис. 6: Новое макро действие в Burp Suite

# Выполнение лабораторной работы

В открытом коде реквеста находим нужный параметр, за которым мы будем следить и запоминать - `user_token`.

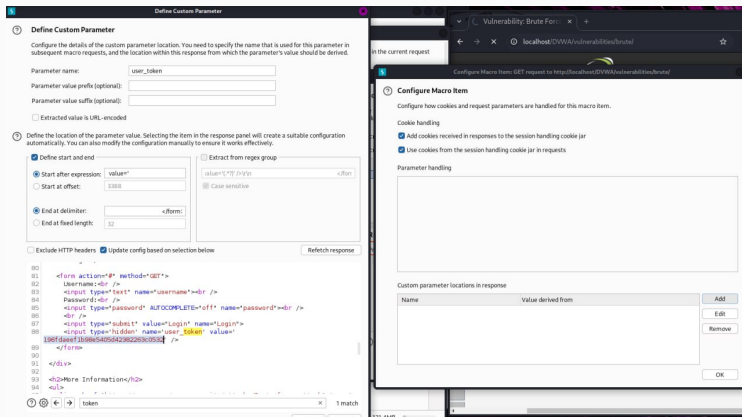
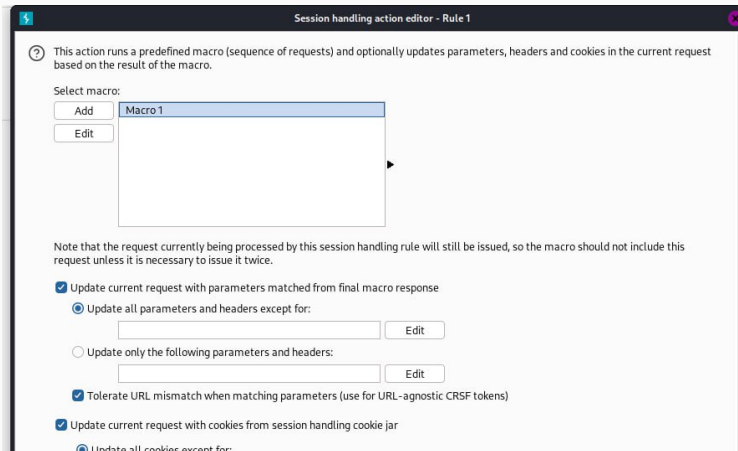


Рис. 7: Отслеживание параметра `user_token` в каждом реквесте

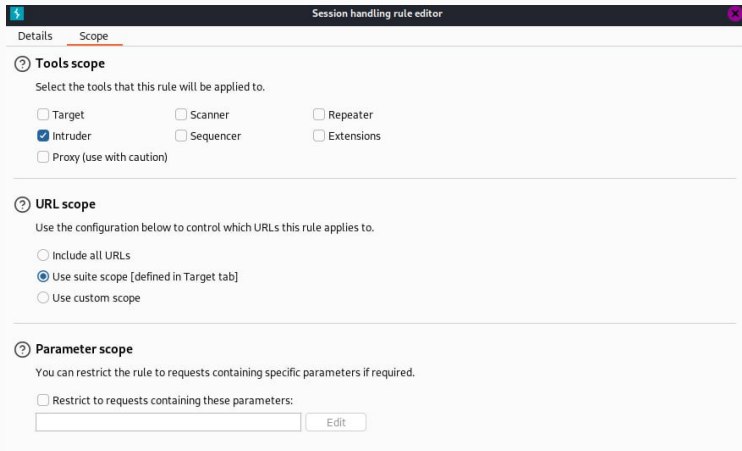
## Выполнение лабораторной работы

Сохраняем наше макро действие, ставим галочку “Tolerate URL mismatch when matching parameters (Use for URL-agnostic CSRF tokens)” - тут написано ставить, если мы имеем дело с юзер токенами.



## Выполнение лабораторной работы

Возвращаемся в настройку правила, выставляем галочки так, чтобы это правило применялось исключительно к инструменту Intruder - им мы будем пользоваться для брут форса приложения.



The screenshot shows the 'Session handling rule editor' window with the 'Scope' tab selected. The window has a dark title bar with a lightning bolt icon on the left and a close button on the right. Below the title bar, there are two tabs: 'Details' and 'Scope', with 'Scope' being the active tab. The main content area is divided into three sections, each with a question mark icon and a title:

- Tools scope**  
Select the tools that this rule will be applied to.  
This section contains six checkboxes arranged in two rows:
  - Target (unchecked)
  - Intruder (checked)
  - Proxy (use with caution) (unchecked)
  - Scanner (unchecked)
  - Sequencer (unchecked)
  - Repeater (unchecked)
  - Extensions (unchecked)
- URL scope**  
Use the configuration below to control which URLs this rule applies to.  
This section contains three radio buttons:
  - Include all URLs (unchecked)
  - Use suite scope [defined in Target tab] (checked)
  - Use custom scope (unchecked)
- Parameter scope**  
You can restrict the rule to requests containing specific parameters if required.  
This section contains one checkbox labeled 'Restrict to requests containing these parameters:' which is unchecked. Below it is a text input field and an 'Edit' button.

# Выполнение лабораторной работы

Начнём. Включаем Interceptor - перехватываем реквест с попыткой входа.

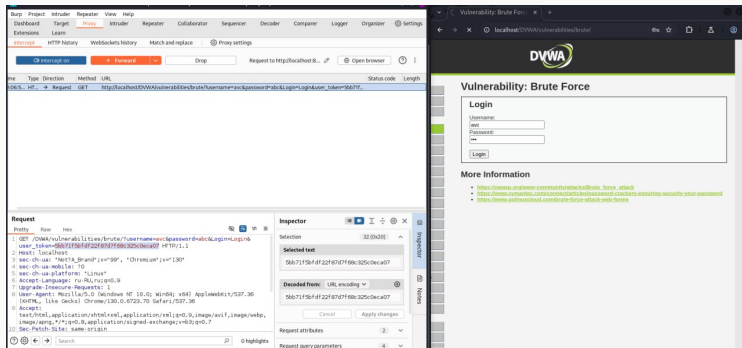
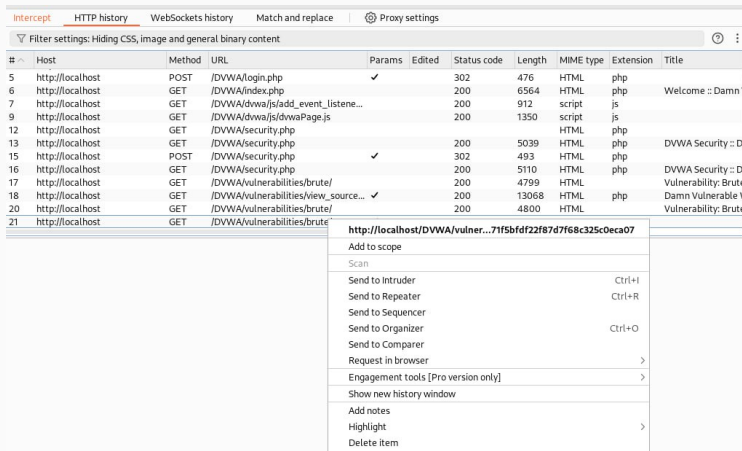


Рис. 10: Перехват реквеста с попыткой входа



# Выполнение лабораторной работы

Открываем вкладку HTTP-history и находим перехваченный реквест, нажимаем на него правой кнопкой и “Send to Intruder” (отправляем в инструмент взломщика), а затем “Add to scope”.



The screenshot shows the Burp Suite interface with the HTTP history tab selected. The table below lists the intercepted requests. The context menu is open for the selected request (row 21).

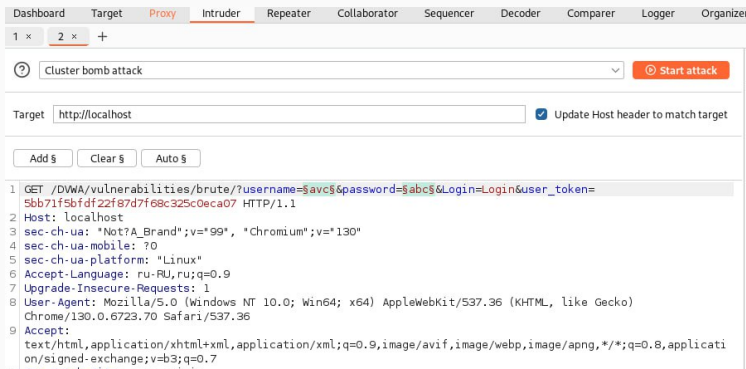
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
5	http://localhost	POST	/DVWA/login.php	✓		302	476	HTML	php	
6	http://localhost	GET	/DVWA/index.php			200	6564	HTML	php	Welcome :: Damn
7	http://localhost	GET	/DVWA/dvwa/js/add_event_listene...			200	912	script	js	
9	http://localhost	GET	/DVWA/dvwa/js/dvwaPage.js			200	1350	script	js	
12	http://localhost	GET	/DVWA/security.php					HTML	php	
13	http://localhost	GET	/DVWA/security.php			200	5039	HTML	php	DVWA Security :: D
15	http://localhost	POST	/DVWA/security.php	✓		302	493	HTML	php	
16	http://localhost	GET	/DVWA/security.php			200	5110	HTML	php	DVWA Security :: D
17	http://localhost	GET	/DVWA/vulnerabilities/brute/			200	4799	HTML		Vulnerability: Brute
18	http://localhost	GET	/DVWA/vulnerabilities/view_source...	✓		200	13068	HTML	php	Damn Vulnerable \
20	http://localhost	GET	/DVWA/vulnerabilities/brute/			200	4800	HTML		Vulnerability: Brute
21	http://localhost	GET	/DVWA/vulnerabilities/brute/							

Context menu options for the selected request:

- http://localhost/DVWA/vulner...71f5bdfd22f87d7f68c325c0eca07
- Add to scope
- Scan
- Send to Intruder (Ctrl+I)
- Send to Repeater (Ctrl+R)
- Send to Sequencer
- Send to Organizer (Ctrl+O)
- Send to Comparer
- Request in browser
- Engagement tools [Pro version only]
- Show new history window
- Add notes
- Highlight
- Delete item

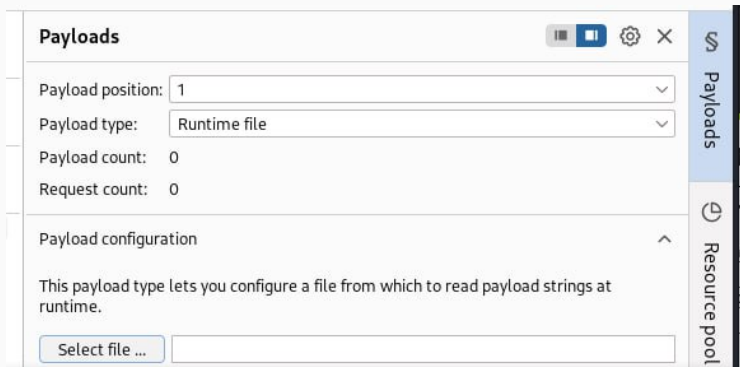
## Выполнение лабораторной работы

Теперь открываем вкладку “Intruder” - находим посланный нами реквест, выбираем тип атаки “Cluster Bomb” - стандартный брут форс - постоянный перебор и отправка реквестов, также выделяем значения параметра username и нажимаем “Add \$”, так мы выделили первый параметр, который мы будем перебирать и посылать каждый реквест - аналогично делаем и со значением переменной password.



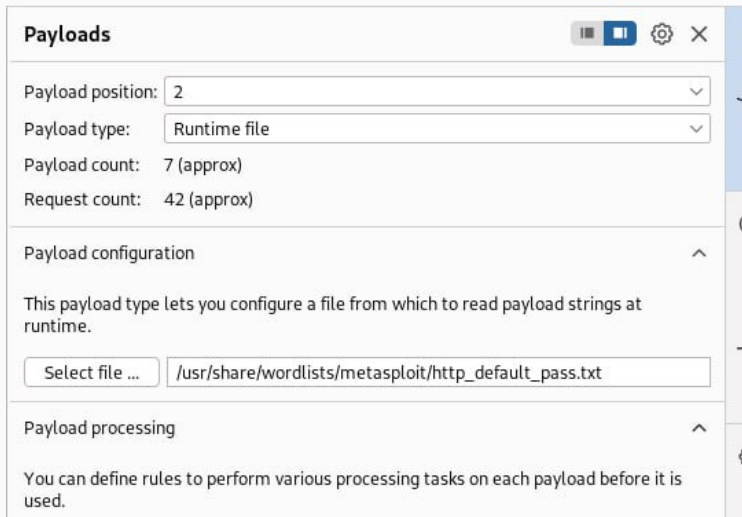
## Выполнение лабораторной работы

Открываем Payloads (тут мы настраиваем переменные, которые будем перебирать, т.к. мы перебираем логин и пароль, у нас их 2). Первый пейлоуд - выбираем, что будем перебирать: значения из файла, выбираем файл - в Kali есть стандартный список дефолтных логинов и паролей - они лежат в `/usr/share/wordlists/metasploit`. Для списка логинов выбираем `http_default_users.txt`.



## Выполнение лабораторной работы

Аналогично делаем и для второго пейлоуда - перебор паролей - http\_default\_pass.txt.



The screenshot shows the 'Payloads' window in Metasploit. It contains the following configuration:

- Payload position:** 2
- Payload type:** Runtime file
- Payload count:** 7 (approx)
- Request count:** 42 (approx)

**Payload configuration**

This payload type lets you configure a file from which to read payload strings at runtime.

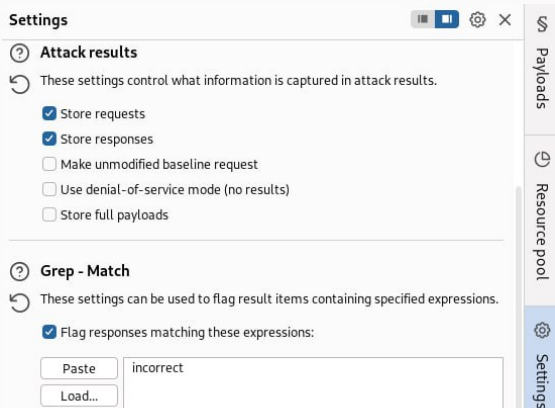
**Select file ...** /usr/share/wordlists/metasploit/http\_default\_pass.txt

**Payload processing**

You can define rules to perform various processing tasks on each payload before it is used.

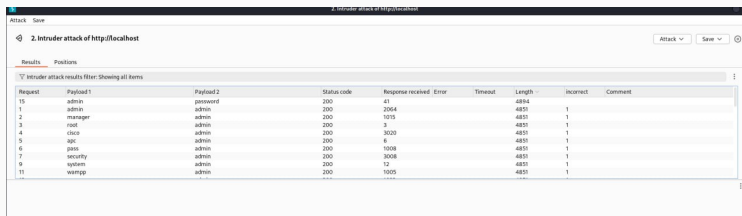
## Выполнение лабораторной работы

Открываем настройки Intruder, для наглядности добавим слово, за которым мы будем следить, и если оно появляется в коде странички - то мы ставим нашему реквесту флажок. Выбираем слово “incorrect”, тогда мы обратим внимание, что при правильном наборе логина и пароля флажка не будет.



# Выполнение лабораторной работы

Запускаем нашего атакующего - начинаем брут форс. Наглядно видно, как посылается много реквестов. На фото я также их отсортировал по длине кода в страничке. Обратим внимание, что тут в первой строке при логине admin и пароле password мало того, нету флажка Incorrect, так ещё и длина кода страничке значительно отличается от всех остальных - явно что-то особенное случилось при таком наборе логина и пароля. Обычно, взломщик в таком случае сам попробует такой набор логина и пароля.



Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Incorrect	Comment
15	admin	password	200	41			4894		
1	admin	admin	200	2064			4851	1	
2	manager	admin	200	1015			4851	1	
3	root	admin	200	3			4851	1	
4	cisco	admin	200	3020			4851	1	
5	apc	admin	200	6			4851	1	
6	pass	admin	200	1008			4851	1	
7	security	admin	200	3008			4851	1	
9	system	admin	200	12			4851	1	
11	wampp	admin	200	1005			4851	1	

Рис. 16: Задocumented брут форс атака

Вставляем такую комбинацию логина и пароля в страничку входа и видим, что мы успешно прорвались в чужок аккаунт.

# Vulnerability: Brute Force


## Login

Username:

Password:

Login

Welcome to the password protected area admin



## Выводы

---



При выполнении данной работы я успешно получил навыки работы с Burp Suite.

## Список литературы

---

Индивидуальный проект

Brute Force DVWA разной сложности с использованием Burp Suite (На английском)