магия макросов в Burp Suite / Хабр



habr.com/ru/articles/811329

artrone April 29, 2024

Лабораторная работа: Обход 2FA с помощью грубой силы





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 <u>лабораторную работу</u> "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:

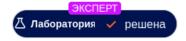
```
def get_csrf_token(text):
 soup = BeautifulSoup(text, 'html.parser')
  return soup.find('input', attrs={'name': 'csrf'})['value']
```

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





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

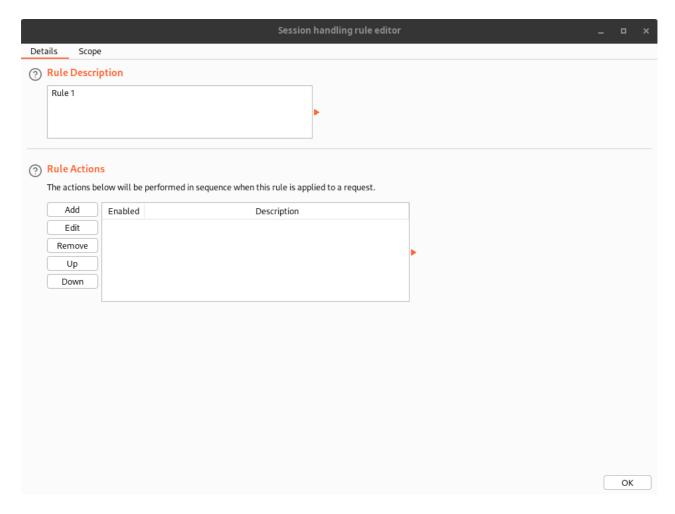
GET /login -> POST /login -> GET /login2 -> POST /login2

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:

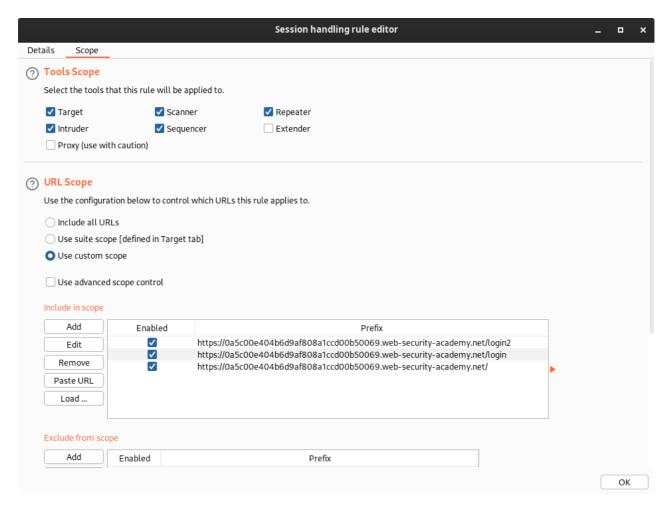
```
<form class=login-form method=POST>
<input required type="hidden" name="csrf"
value="DGgC7DB4dlFk1d4ZUdyEBoHklDP9KdNg">
<label>
```

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

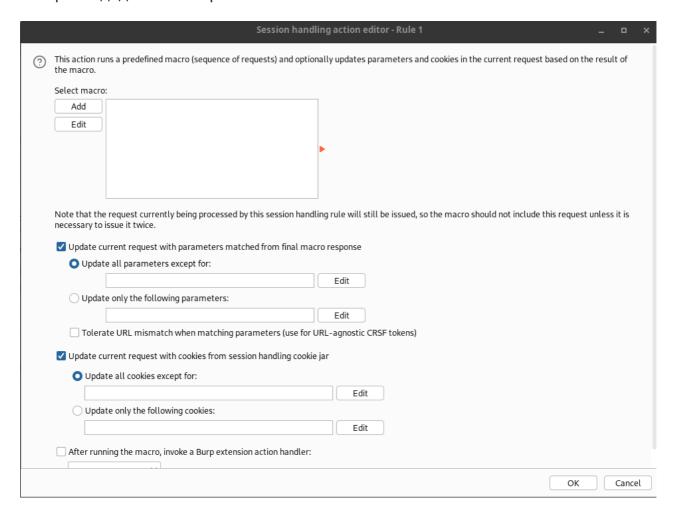
Для этого перейдем в Project options -> Sessions -> Session Handling Rules -> Add



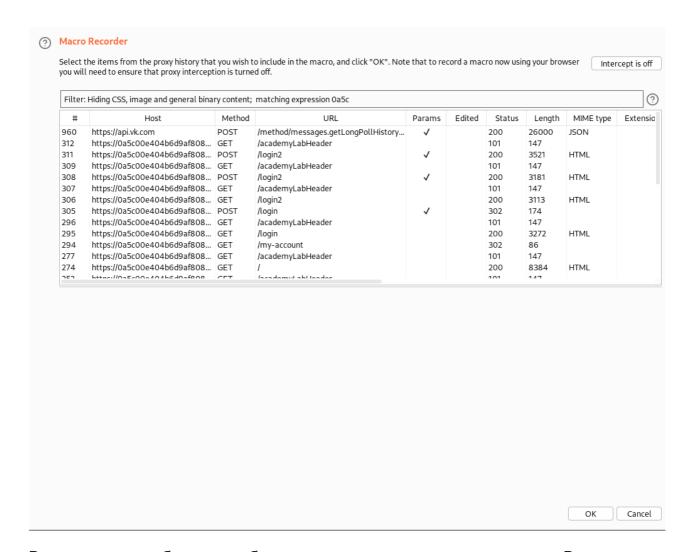
Сразу выберем скоуп:



Теперь создадим новое правило "Run a macro"



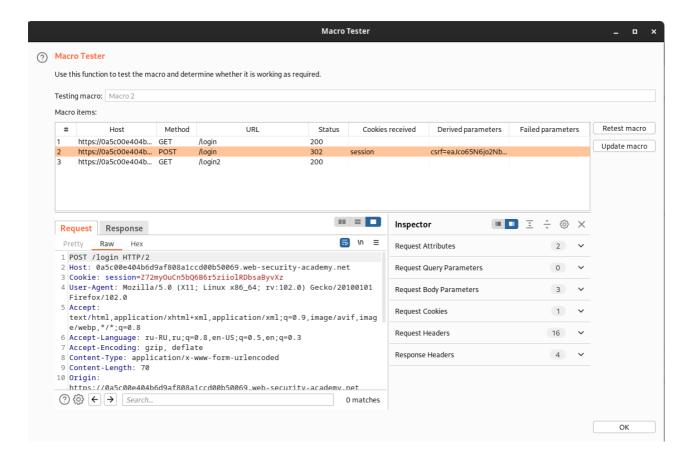
Создадим макро, нажав на "Add"



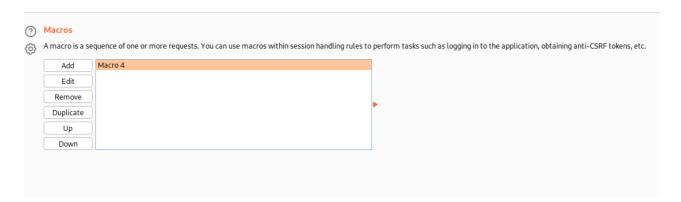
В данном окне выбираем необходимую последовательность запросов. В нашем случае,- это GET /login POST /login GET /login2



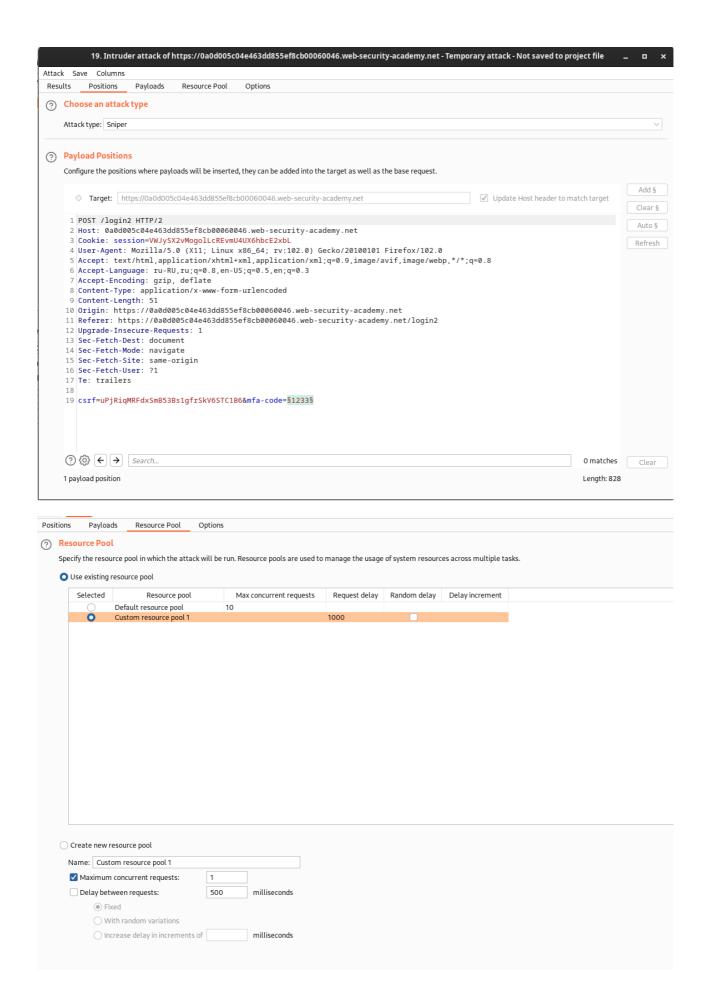
На всякий случай, лучше убедиться, что всё работает, нажав на кнопку "Test macro"

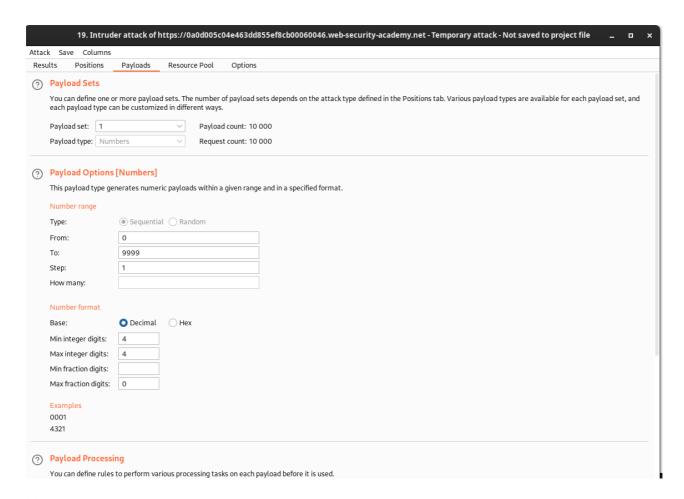


Как видим, макрос успешно создан:

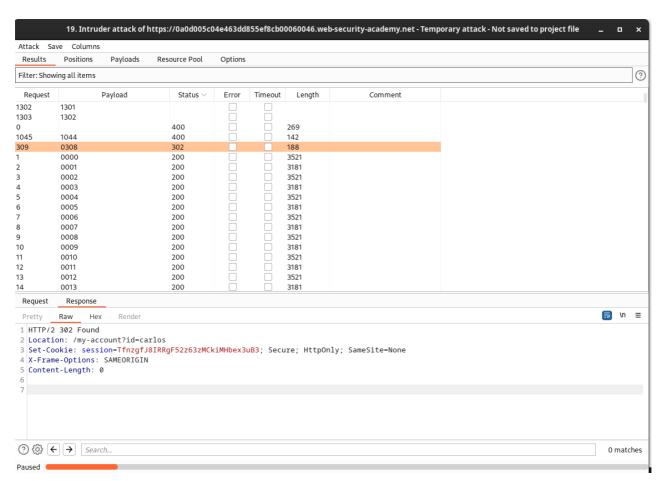


Теперь перейдем в Intruder и настроим его нужным образом





И запустим:



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 :)