CWE-22 Path Traversal in load_prompt() — .txt file read without authentication

Moderate **oobabooga** published **GHSA-mfgg-vvc6-vqq7** 5 days ago

Package

text-generation-webui (pip)

Affected versions	Patched versions
< 4.3	4.3

Description

Summary

An unauthenticated path traversal vulnerability in `load_prompt()` allows reading any `.txt` file on the server filesystem.
The file content is returned verbatim in the API response.

Details

The vulnerable code is in `modules/prompts.py` at lines 7-24:

```
def load_prompt(fname):
    file_path = shared.user_data_dir / 'logs' / 'notebook' / f'{fname}.txt'
    if file_path.exists():
        with open(file_path, 'r', encoding='utf-8') as f:
            text = f.read()
            return text.rstrip()
```

poc.zip

The `fname` parameter comes from a Gradio Dropdown.
Gradio does not server-side validate dropdown values, so an attacker can POST `fname="../../secret/api_keys"` via the API.
The path resolves to `logs/notebook/../../secret/api_keys.txt`, escaping the intended directory.
No `os.path.basename()` or `sanitize_filename()` is applied.
The `.txt` extension is always appended, limiting reads to text files.

PoC

1. Clone the repository and start the server.
2. Send a crafted API request with a traversal payload as the prompt filename.
3. The server opens the target `.txt` file and returns its content verbatim in the Gradio Textbox response.

I verified this by cloning the repository, running the verbatim `load_prompt()` function with `fname="../../secret/api_keys"`, and confirming that `.txt` file content from outside the logs directory is returned verbatim.

Impact

Any `.txt` file readable by the server process can be exfiltrated.
Many sensitive files use `.txt` extension: API key files, environment notes, deployment logs, password lists, license keys.
No authentication required by default.

Remediation: apply `os.path.basename(fname)` before path construction.

We believe this qualifies as a valid security issue.
If you agree, we'd appreciate the following credit on the CVE:
Reported by Woohyun Choi, Sunwoo Lee, and Seunghyun Yoon (Korea Institute of Energy Technology, KENTECH)

Severity

Moderate 5.3 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	Low

Integrity	None
Availability	None

Learn more about base metrics

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

CVE ID

CVE-2026-35487

Weaknesses

▶ CWE-22

Credits

programsurf	Reporter
woohyunchoi-kentech	Reporter
yoonsh	Reporter