COMMUNITY CONTENT

Home › Submit › 791761

Submit #791761: FoundationAgents MetaGPT 0.8.1 Code Injection (CWE-94)

Title FoundationAgents MetaGPT 0.8.1 Code Injection (CWE-94)

Description

Technical Details

A Code Injection vulnerability exists in the Tree-of-Thought (ToT) solver in MetaGPT (metagpt/strategy/tot.py), where Python's eval() function is used to parse LLM responses without validation, leading to Remote Code Execution (RCE).

In the generate_thoughts() method, the system queries the LLM and extracts code block content using CodeParser.parse_code(text=rsp). The extracted content is then passed directly to eval(thoughts) on line 66. Because eval() allows the execution of arbitrary Python code, an attacker who can influence the LLM's response (e.g., via prompt injection) can execute malicious code.

Vulnerable Code

File: metagpt/strategy/tot.py
Method: generate_thoughts()
Why: Line 66 calls thoughts = eval(thoughts) directly on the untrusted string extracted from the LLM's output.

Reproduction

1. Simulate a scenario where the LLM is influenced by malicious prompt injection to return the following response:

```json
__import__('os').system('id > /tmp/tot_eval_rce_proof.txt') or [{"node_id": "1", "node_state_instruction": "legitimate thought"}]
```

2. When the ToT solver calls generate_thoughts(), it extracts the code block and evaluates the payload.
3. The os.system() command executes successfully, and the or condition returns a valid list so the program does not crash.
4. Verify /tmp/tot_eval_rce_proof.txt is created on the filesystem.

Impact

- Remote Code Execution (RCE) via LLM Prompt Injection: An attacker can achieve full system access by crafting input that causes the LLM to include Python code in its response. This can occur via user prompt injection, a compromised API endpoint, or a poisoned model.

Source ⚠ https://github.com/FoundationAgents/MetaGPT/issues/1933

User Eric-d (UID 96861)

Submission 03/28/2026 04:40 AM (15 days ago)

Community Content

Submissions are made by VulDB community users. VulDB is *not responsible* for their content nor the links to external sources.

Please use the raw information shown and the links listed *with caution*. They might contain malicious and harmful actions, code or data.

The corresponding VulDB entries contain the moderated, verified, and normalized information provided within the raw submission.

Documentation

- Submission Policy
- Data Processing
- CVE Handling

Moderation	04/11/2026 09:49 AM (14 days later)
Status	Accepted
VulDB entry	356570 [FoundationAgents MetaGPT up to 0.8.1 Tree-of-Thought Solver metagpt/strategy/tot.py generate_thoughts code injection]
Points	20