COMMUNITY CONTENT

Home › Submit › 784454

Submit #784454: openchatbi v0.2.1 SQL Injection

Title	openchatbi v0.2.1 SQL Injection
Description	OpenChatBI suffers from a critical Arbitrary SQL Run Vulnerability by prompt injection, including statements that can lead to remote code execution on the database server. The vulnerability exists in the multi-stage Text2SQL workflow where user input is processed through several LLM-driven nodes (Agent, information extraction, schema linking, and SQL generation) before being executed against the database. An attacker can craft malicious prompts that manipulate each stage of the pipeline to inject arbitrary SQL commands. The core issue is that the SQL generated by llm is executed directly without any validation or sanitization. The attack flow works as follows: 1. **Agent Call tool Stage**: The attacker demand the Agent to call text2sql tool with specific context(prompt for following llm node). 2. **Information Extraction Stage**: The attacker's prompt manipulates the LLM to return attacker-controlled JSON output for the rewrite_question and keywords fields. 3. **Schema Linking Stage**: The manipulated prompt causes the LLM to return specified table selections. we manipulate step2&3 to bypass the validation in step3 which check the tables that will be used are within the candidate tables searched by keywords generated by step2. 4. **SQL Generation Stage**: The prompt injection causes the LLM to generate malicious SQL that includes dangerous database-specific commands like PostgreSQL's `COPY FROM PROGRAM`, which can execute arbitrary system commands. 5. **SQL Execution Stage**: The malicious SQL is executed without any validation, allowing the attacker's commands to run on the database server.
Source	https://github.com/ka7arotic/cve/blob/main/openchatbi-SQL/issue.md
User	Goku (UID 804896)
Submission	03/21/2026 02:29 AM (16 days ago)
Moderation	04/04/2026 11:42 PM (15 days later)
Status	Approved
VulDB entry	[zhongyu09 openchatbi up to 0.2.1 Multi-stage Text2SQL Workflow keywords sql injection]

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

Points 20