SECURITY ANALYSIS REPORT

CVE ID: CVE-1999-0276
Analysis Date: November 07, 2025
Total Files Analyzed: 8
Total Matches: 8

Vulnerability Summary:
mSQL v2.0.1 and below allows remote execution through a buffer overflow.

Executive Summary

This report presents the results of an automated security analysis performed to identify potential vulnerabilities related to **CVE-1999-0276** in the target codebase.

The analysis utilized advanced query decomposition and semantic search techniques to scan **8** unique files, generating **3** specialized search queries that resulted in **8** potential matches.

Each identified file has been analyzed for patterns and code structures that may be related to the vulnerability, providing actionable insights for security remediation.

Key Metrics

Metric	Value
CVE ID	CVE-1999-0276
Total Queries Generated	3
Files Scanned	8
Total Matches Found	8
Analysis Date	2025-11-07 17:56:51

Vulnerability Details

CVE Identifier: CVE-1999-0276

Description:
mSQL v2.0.1 and below allows remote execution through a buffer overflow.

Analysis Methodology

The analysis employed a multi-stage approach:

1. Query Decomposition (Hype): The CVE description was decomposed into multiple specialized search queries using hypothetical answer generation techniques.

2. Semantic Search: Each query was executed against a FAISS vector database containing embedded representations of the codebase files.

3. File Retrieval: Matched files were retrieved and their full contents analyzed.

4. Consolidation: Results were aggregated and ranked by relevance score.

Generated Search Queries

Query 1: mSQL v2.0.1 and below allows remote execution through a buffer overflow.

Query 2: mSQL v2.0.1 and below allows remote execution through a buffer overrun.

Query 3: mSQL v2.0.1 and below allows remote execution through a stack overflow.

Findings Overview

Rank	File Path	Relevance	Size
1	examples\tutorial\tests\data.sql	N/A	394 chars
2	examples\tutorial\flaskr\schema.sql	N/A	498 chars
3	src\flask__main__.py	N/A	30 chars
4	docs\deploying\asgi.rst	N/A	500 chars
5	tests\test_apps\helloworld\wsgi.py	N/A	36 chars
6	examples\tutorial\flaskr\db.py	N/A	500 chars
7	tests\static\index.html	N/A	22 chars
8	tests\test_apps\cliapp\multiapp.py	N/A	67 chars

Detailed File Analysis

File 1: examples\tutorial\tests\data.sql

Matched Query: mSQL v2.0.1 and below allows remote execution through a buffer overflow.
Relevance Score: 0.0000
File Size: 394 characters

Content Preview:

```
INSERT INTO user (username, password)
VALUES
('test', 'pbkdf2:sha256:50000$TCI4GzcX$0de171a4f4dac32e3364c7ddc7c14f3e2fa61f2d1757448
3f7ffbb431b4acb2f'),
('other', 'pbkdf2:sha256:50000$kJPKsz6N$d2d4784f1b030a9761f5ccaeeaca413f27f2ecb76d6168
407af962ddce849f79');

INSERT INTO post (title, body, author_id, created)
VALUES
('test title', 'test' || x'0a' || 'body', 1, '2018-01-01 00:00:00');
```

File 2: examples\tutorial\flaskr\schema.sql

Matched Query: mSQL v2.0.1 and below allows remote execution through a buffer overflow.
Relevance Score: 0.0000
File Size: 498 characters

Content Preview:

```
-- Initialize the database.
-- Drop any existing data and create empty tables.

DROP TABLE IF EXISTS user;
DROP TABLE IF EXISTS post;

CREATE TABLE user (
id INTEGER PRIMARY KEY AUTOINCREMENT,
username TEXT UNIQUE NOT NULL,
password TEXT NOT NULL
);

CREATE TABLE post (
id INTEGER PRIMARY KEY AUTOINCREMENT,
author_id INTEGER NOT NULL,
created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
title TEXT NOT NULL,
body TEXT NOT NULL,
FOREIGN KEY (author_id) REFERENCES user (id)
);
```

File 3: src\flask__main__.py

Matched Query: mSQL v2.0.1 and below allows remote execution through a buffer overflow.
Relevance Score: 0.0000

File Size: 30 characters

Content Preview:

```
from .cli import main

main()
```

File 4: docs\deploying\asgi.rst

Matched Query: mSQL v2.0.1 and below allows remote execution through a buffer overflow.
Relevance Score: 0.0000
File Size: 673 characters

Content Preview:

```
ASGI
====

If you'd like to use an ASGI server you will need to utilise WSGI to
ASGI middleware. The asgiref
`WsgiToAsgi <https://github.com/django/asgiref#wsgi-to-asgi-adapter>`_
adapter is recommended as it integrates with the event loop used for
Flask's :ref:`async_await` support. You can use the adapter by
wrapping the Flask app,

.. code-block:: python

from asgiref.wsgi import WsgiToAsgi
from flask import Flask

app = Flask(__name__)

...

asgi_app = WsgiToAsgi(app)

and then serving the ``asgi_app`` with the ASGI server, e.g. using
`Hypercorn <https://github.com/pgjones/hypercorn>`_,

.. sourcecode:: text

$ hypercorn module:asgi_app
```

File 5: tests\test_apps\helloworld\wsgi.py

Matched Query: mSQL v2.0.1 and below allows remote execution through a buffer overflow.
Relevance Score: 0.0000
File Size: 36 characters

Content Preview:

```
from hello import app # noqa: F401
```

File 6: examples\tutorial\flaskr\db.py

Matched Query: mSQL v2.0.1 and below allows remote execution through a buffer overrun.
Relevance Score: 0.0000
File Size: 1317 characters

Content Preview:

```
import sqlite3
from datetime import datetime

import click
from flask import current_app
from flask import g


def get_db():
"""Connect to the application's configured database. The connection
is unique for each request and will be reused if this is called
again.
"""
if "db" not in g:
g.db = sqlite3.connect(
current_app.config["DATABASE"], detect_types=sqlite3.PARSE_DECLTYPES
)
g.db.row_factory = sqlite3.Row

return g.db


def close_db(e=None):
"""If this request connected to the database, close the
connection.
"""
db = g.pop("db", None)

if db is not None:
db.close()


def init_db():
"""Clear existing data and create new tables."""
db = get_db()

with current_app.open_resource("schema.sql") as f:
db.executescript(f.read().decode("utf8"))


@click.command("init-db")
def init_db_command():
"""Clear existing data and create new tables."""
init_db()
click.echo("Initializ
```

... truncated (317 more characters)

File 7: tests\static\index.html

Matched Query: mSQL v2.0.1 and below allows remote execution through a stack overflow.
Relevance Score: 0.0000
File Size: 22 characters

Content Preview:

```
<h1>Hello World!</h1>
```

File 8: tests\test_apps\cliapp\multiapp.py

Matched Query: mSQL v2.0.1 and below allows remote execution through a stack overflow.
Relevance Score: 0.0000
File Size: 67 characters

Content Preview:

```
from flask import Flask

app1 = Flask("app1")
app2 = Flask("app2")
```

Appendix

A. Analysis Configuration

Base Query: mSQL v2.0.1 and below allows remote execution through a buffer overflow.
Total Queries: 3
Total Matches: 8
Unique Files: 8
Generated: 2025-11-07 17:56:51

B. Disclaimer

This report is generated by an automated security analysis tool and should be reviewed by qualified security professionals. The matches identified may include false positives and require manual verification. The analysis is based on semantic similarity and does not constitute a comprehensive security audit.