CQL Injection in Cassandra Export Module via Unsanitized Config Values

Moderate **nicolargo** published **GHSA-grp3-h8m8-45p7** 2 days ago

Package

glances (pip)

Affected versions

< 4.4.0

Patched versions

> 4.4.1

Description

Summary

The Cassandra export module (`glances/exports/glances_cassandra/__init__.py`) interpolates `keyspace`, `table`, and `replication_factor` configuration values directly into CQL statements without validation. A user with write access to `glances.conf` can redirect all monitoring data to an attacker-controlled Cassandra keyspace.

Vulnerable Code

```
# Line 80
f"CREATE KEYSPACE {self.keyspace} WITH "
f"replication = {{ 'class': 'SimpleStrategy', 'replication_factor': '{self.replication_f

# Line 94
f"CREATE TABLE {self.table} (plugin text, time timeuuid, stat map<text,float>, PRIMARY K

# Line 112
stmt = f"INSERT INTO {self.table} (plugin, time, stat) VALUES (?, ?, ?)"
```

Steps to Reproduce

1. Configure `glances.conf` with malicious `table` value:

```
[cassandra]
host = 127.0.0.1
port = 9042
keyspace = glances
table = attacker_ks.captured_stats
```

2. Create attacker keyspace in Cassandra
3. Run `glances --export cassandra`
4. All monitoring data is written to `attacker_ks.captured_stats` instead of the legitimate table

Confirmed output:

```
INSERT stmt: INSERT INTO attacker_ks.captured_stats (plugin, time, stat) VALUES (?
?, ?)
Legitimate table row count: 0
Attacker table row count: 1
[CONFIRMED] plugin=cpu, stat={'user': 50.0}
```

Impact

All exported monitoring data (CPU, memory, network, disk I/O) is silently redirected to an attacker-controlled Cassandra keyspace — both data exfiltration and data loss.

Relationship to Previous CVEs

Same vulnerability class as CVE-2026-30930 (TimescaleDB, GHSA-x46r-mf5g-xpr6) and CVE-2026-32611 (DuckDB, GHSA-49g7-2ww7-3vf5). The Cassandra exporter was missed in those fix rounds.

Proposed Fix

```python
import re

def _validate_cql_identifier(name: str) -> str:
    if not re.match(r'^[a-zA-Z_][a-zA-Z0-9_.]*$', name):
        raise ValueError(f"Invalid CQL identifier: {name!r}")
    return name

# In __init__(): validate before use
self.keyspace = _validate_cql_identifier(self.keyspace)
self.table = _validate_cql_identifier(self.table)
```

PoC

Severity

Moderate 6.3 / 10

CVSS v3 base metrics

Attack vector	Local
Attack complexity	Low
Privileges required	High
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	Low

Learn more about base metrics

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

CVE ID

CVE-2026-35588

Weaknesses

▶ CWE-89

Credits

morimori-dev Reporter