

How NOT to use DuckDB



Sam Jewell
Staff Software Engineer

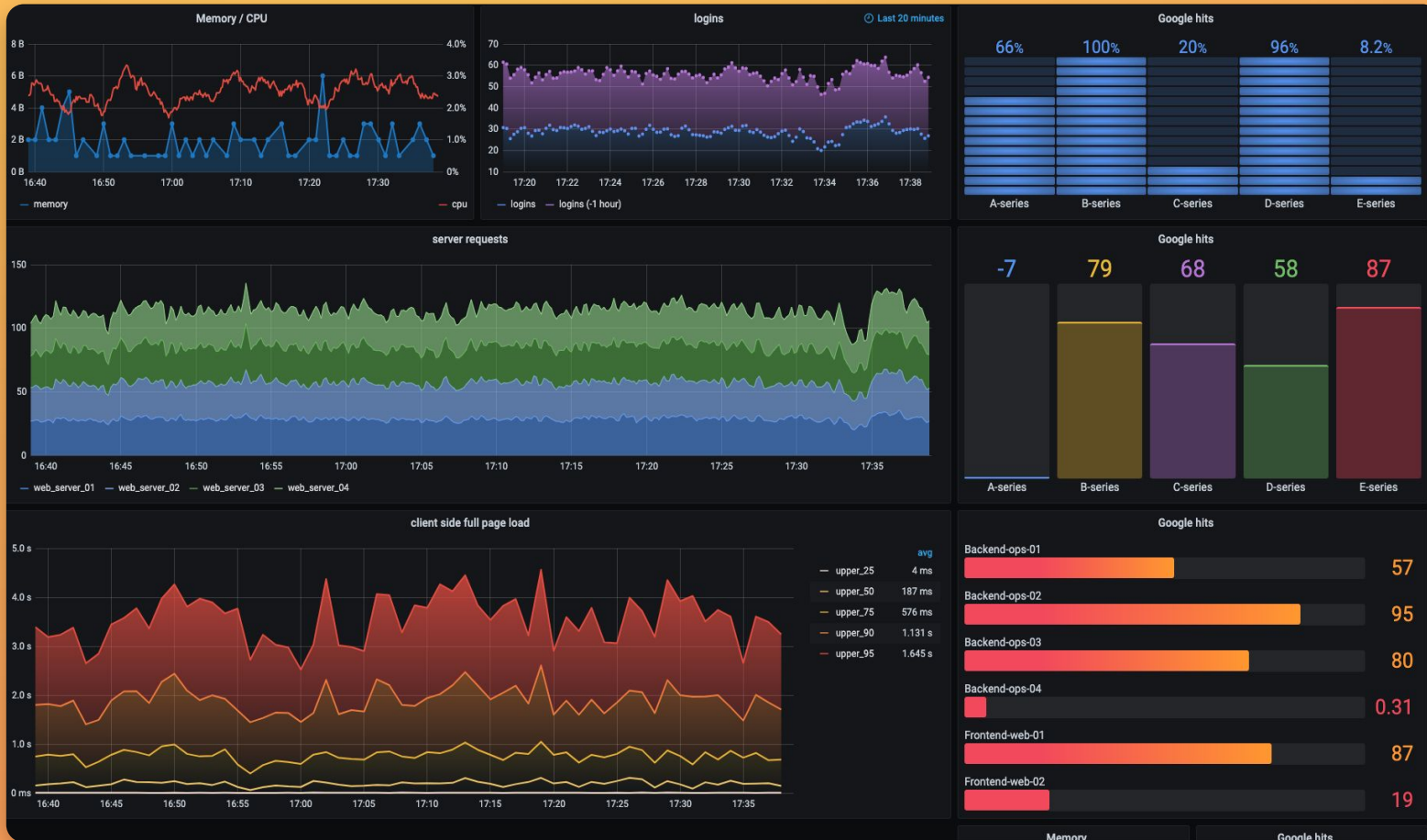
size	filename	content
9050	/etc/passwd	##\x0A# User Database\x0A# \x0A# Note that this file

Query 1 Transform data 0

Data source testdata-DS

Operation SQL

```
1 SELECT size, filename, content
2 FROM read_blob('/etc/passwd');
3
```





Grafana Labs' big tent strategy

A Grafana Logging Hidden

Kick start your query Label browser Explain query Run queries Builder Code

```
min_over_time(  
  {container="attributions", namespace="usage-service"}  
  | = "Report metadata"  
  | json org_id, instance_id, year, month, BillableSeriesBilled  
  | unwrap errorRatio  
  [1d]  
)
```

LOGS

> Options Type: Instant This query will process approximately 556.0 KiB.

B NEW bigquery-grafanal Hidden

Processing location: Automatic location selection Format: Table Run query Builder Code

```
1 select  
2   CAST(org_id AS STRING) AS org_id,  
3   org_slug, org_name, committed_arr  
4 from grafanalabs-global.dbt.core_orgs  
5 where hm_usage > 0  
6 order by committed_arr desc  
7 limit 1000
```







SQL TABLES







✓ This query will process 61.7 MiB when run.








METRICS

TRACES

JSON & CSV

>  `Hi min_over_time({container="attributions",namespace="usage-service"} |= "Report metadata" | jso...`     

> **B**  `NEW bigquery-grafana:k`  *Hidden*    

∨ **C**  `Expression`      

Operation

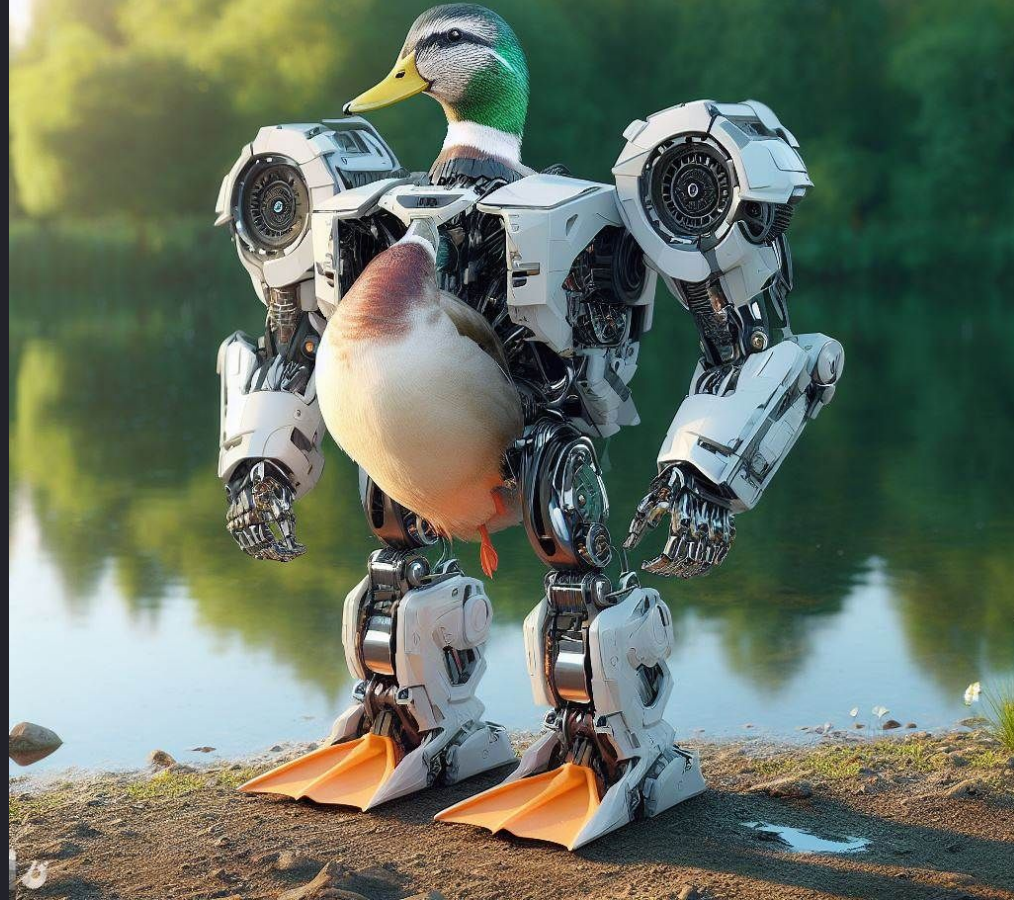
Choose 

```
1 select
2   BillableSeriesBilled,
3   A.org_id,
4   org_slug,
5   committed_arr,
6   instance_id,
7   month,
8   year,
9   abs(Value) AS Error
10 from A
11 LEFT JOIN B
12   ON A.org_id = B.org_id
```




НАС НАТНО 10

OPTIMUS DUCK



security

* Customer security questions: create ...

 747



-  Messages
-  Add canvas
-  Files
-  Bookmarks
-  Pins
- 

[View newer replies](#)

26 September 2024 ▾



todd 16:54

while debugging something else, i noticed that the SQL Expression uses the duckdb CLI under the hood and duckdb's CLI has a [dot command](#) that allows you to run shell commands. the SQL expression does some checks to prevent running anything that isn't SQL. i attempted to bypass the [escaping of single quotes here](#), but i'm wondering if someone from [#security](#) could double check this



21 replies Last reply 4 months ago





1.1.3 (stable) ▾



`.shell CMD`
`ARGS...`

Run `CMD ARGS...` in a system shell

`.show`

Show the current values for various settings

`.singleline`

Set single-line mode

`.system CMD`
`ARGS...`

Run `CMD ARGS...` in a system shell

https://duckdb.org/docs/api/cli/dot_commands.html





1.1.3 (stable) ▾



`read_blob(source)`

Description Returns the content from `source` (a filename, a list of filenames, or a glob pattern) as a `BLOB`. See the [read_blob_guide](#) for more details.

Example `read_blob('hello.bin')`

Result `hello\x0A`

https://duckdb.org/docs/sql/functions/blob.html#read_blobsource



```
SELECT size, filename, content
```

```
FROM read_blob('/etc/passwd');
```

size	filename	content
9050	/etc/passwd	##\x0A# User Database\x0A# \x0A# Note that this file is consulted directly only when

Query 1 Transform data 0

Data source testdata-DS

Query ...

MD = auto = 1415

Interval = 15s

Query inspector

B (Expression)

Operation SQL

```
1 SELECT size, filename, content
2 FROM read_blob('/etc/passwd');
3
```



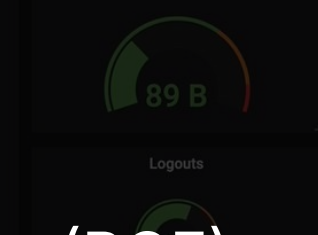
Memory / CPU

logins

timeshift -1h

Memory

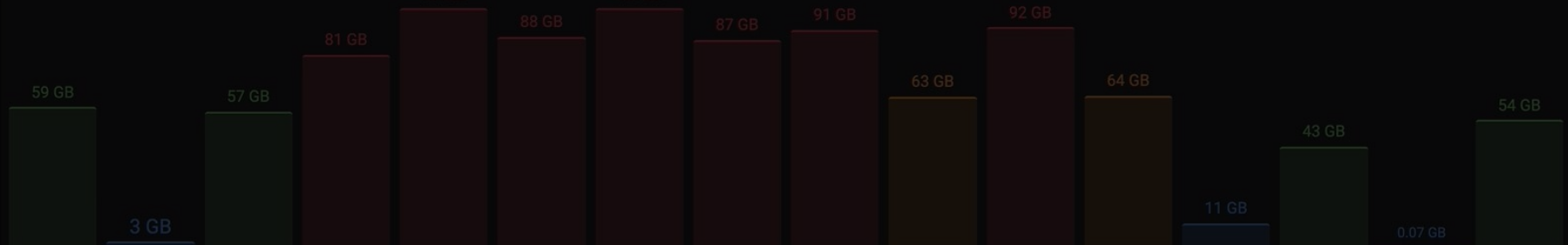
Sign ups



`.shell` ⇒ Remote Code Execution (RCE)

`read_blob` ⇒ Local File Inclusion (LFI)

CVSS rating:
 9.9 / 10 (CVSSv3) CRITICAL
 9.4 / 10 (CVSSv4) CRITICAL



The Fallout

- Security impact relatively small
 - Identified internally
 - Behind feature flag - only ~2 customers enabled
 - Checked every log-line. No evidence of exploitation
- High effort to remedy
 - 2-week SLO for Critical vulnerabilities
 - Remove the feature entirely
 - Roll to our hosted SAAS
 - Rebuild Grafana Enterprise and Grafana OSS
 - Disclose and share privately, with 2 week embargo
- We learned a lot
 - Blameless post-mortem



How did we get here?

- We implemented on the Grafana backend, which is written in Go

Documentation / Client APIs

Go

The DuckDB Go driver, `go-duckdb`, allows using DuckDB via the `database/sql` interface. For examples on how to use this interface, see the [official documentation](#) and [tutorial](#).



Note

The `go-duckdb` project, hosted at <https://github.com/marcboeker/go-duckdb>, is the official DuckDB Go client.

Installation

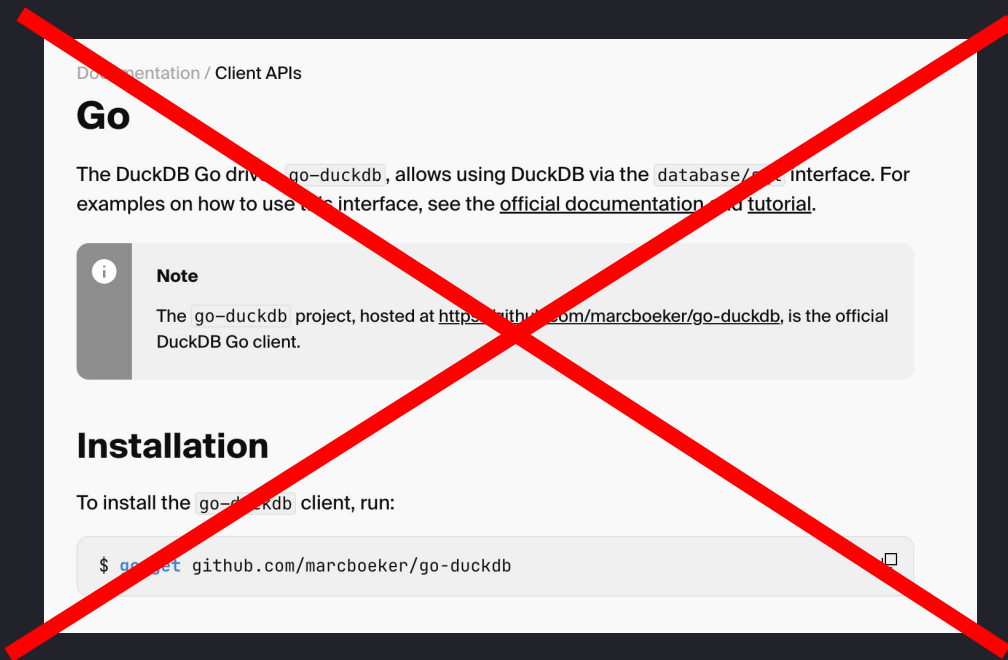
To install the `go-duckdb` client, run:

```
$ go get github.com/marcboeker/go-duckdb
```



How did we get here?

- We implemented on the Grafana backend, which is written in Go
- But we **can't use CGo**



How did we get here?

- We implemented on the Grafana backend, which is written in Go
- But we **can't use CGo**
- So we used the CLI 🤖



Avoid our mistake!

Lesson 1

- Embed, don't use the CLI (or [parse & validate](#))

Documentation / Operations Manual

Embedding DuckDB

CLI Client

The Command Line Interface (CLI) client is intended for interactive use cases and **not for embedding**. As a result, it has more features that could be abused by a malicious actor. For example, the CLI client has the `.sh` feature that allows executing arbitrary shell commands. This feature is only present in the CLI client and not in any other DuckDB clients.

```
.sh ls
```



Tip

Calling DuckDB's CLI client via shell commands is **not recommended** for embedding DuckDB. It is recommended to use one of the client libraries, e.g., [Python](#), [R](#), [Java](#), etc.



Avoid our mistake!

Lesson 2

- Disable file access

Documentation / Operations Manual / Securing DuckDB

Securing DuckDB

DuckDB is quite powerful, which can be problematic, especially if untrusted SQL queries are run, e.g., from public-facing user inputs. This page lists some options to restrict the potential fallout from malicious SQL queries.

The approach to securing DuckDB varies depending on your use case, environment, and potential attack models. Therefore, consider the security-related configuration options carefully, especially when working with confidential data sets.

If you plan to embed DuckDB in your application, please consult the [“Embedding DuckDB”](#) page.

Reporting Vulnerabilities

If you discover a potential vulnerability, please [report it confidentially via GitHub](#).

Disabling File Access

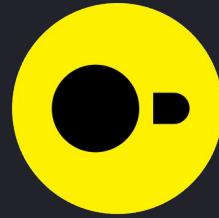
DuckDB can list directories and read arbitrary files via its CSV parser’s [read_csv function](#) or read text via the [read_text function](#). For example:

```
SELECT *  
FROM read_csv('/etc/passwd', sep = ':');
```

This can be disabled either by disabling external access altogether (`enable_external_access`) or disabling individual file systems. For example:

```
SET disabled_filesystems = 'LocalFileSystem';
```





Thank you



Sam Jewell

 @sojewell

Have more questions?

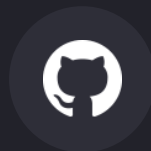
Join us at community.grafana.com

or Grafana public slack [#grafana](#)

Get involved:



[#grafana](#)



[grafana](#)



community.grafana.com