Ben Johnston

Unmasking SQL Server

SQLSaturday #1124
September 27, 2025

- **Visit the Sponsor Booths**
- **Lots of Great Raffle Prizes!**
- **Enter your details on Raffle Entry Form & visit Sponsors to fill in Bingo card for Raffle prizes**

Thank you Sponsors!

Gold Sponsors:

Microsoft | AMD

databricks

Silver Sponsors:

TRACE3

Bronze Sponsors:

O'REILLY

PASSMN – News/Info

passmn
SQL Server User Group

Thanks to all our sponsors of 2025!

We need Speakers & Sponsors for 2026 PASSMN Meetings!

- Speakers – email us at speakers@passmn.com
- Sponsors – email us at sponsors@passmn.com

Monthly Meetup: 3rd Tuesday of Each Month
Signup on Meetup:
https://www.meetup.com/MN-SQL-Server-User-Group-PASSMN/

Board Member Elections in Q4:

- Your chance to help the MN Data/SQL Community!
  - https://forms.microsoft.com/r/qcjUg5Cwk4

Connect, share, & learn with peers and thought leaders while celebrating all things data for a week of learning and networking opportunities.

Use Code UGRP007 for $150 off!!!

November 17th Through November 21st in Seattle, WA

https://passdatacommunitysummit.com/

Ben Johnston

Data Architect
I-Tech Solutions, Inc.

https://www.red-gate.com/simple-talk/author/ben-johnston/

linkedin.com/in/bensql

https://github.com/BenSQL/SQLSideChannel

I've worked with SQL Server and database systems for over 25 years, building warehouses, transactional systems, reporting environments, and system integrations.

Consulting with medium and large enterprises, I specialize in database design, performance tuning, security, troubleshooting, and Azure upgrades.

Dynamic data masking / Row level security (RLS)

- Supplemental layers to your security design
- Not encryption
- Obfuscates or hides data
- Allows standard queries without modification
  - No client side changes
  - Data differs depending on user
- Vulnerable to specially crafted attacks

Dynamic Data Masking

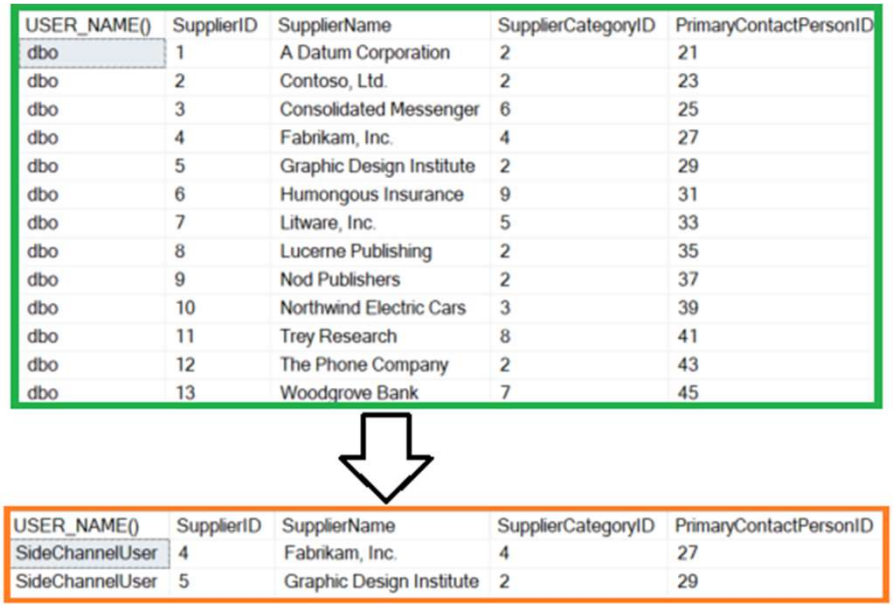
- Specified column data is obfuscated (masked)
- Depends on user security
- DBO, admins, owners, UNMASK see raw data
- All functionality supported
  - SELECT
  - JOIN
  - Functions
  - Comparisons
- Part of the table / column definition

USER_NAME()	CustomerID	CustomerName	PhoneNumber
dbo	74	Tailspin Toys (Indios, PR)	((781) 555-0174
dbo	84	Tailspin Toys (Aceitunas, PR)	((782) 555-0184
dbo	116	Tailspin Toys (Rafael Capó, PR)	((783) 555-0116
dbo	431	Wingtip Toys (Rosa Sánchez, PR)	((784) 555-0431
dbo	454	Wingtip Toys (Corcovado, PR)	((785) 555-0454
dbo	488	Wingtip Toys (Isabela, PR)	((786) 555-0488
dbo	560	Wingtip Toys (Palmas del Mar, PR)	((787) 555-0560
dbo	821	Kwanjai Saowaluk	((788) 555-0821
dbo	1050	Amrita Ganguly	((789) 555-1050
dbo	1051	Sylvie Laramee	((710) 555-1051

USER_NAME()	CustomerID	CustomerName	PhoneNumber
SideChannelUser	74	xxxx	((781)123-4564
SideChannelUser	84	xxxx	((782)123-4564
SideChannelUser	116	xxxx	((783)123-4566
SideChannelUser	431	xxxx	((784)123-4561
SideChannelUser	454	xxxx	((785)123-4564
SideChannelUser	488	xxxx	((786)123-4568
SideChannelUser	560	xxxx	((787)123-4560
SideChannelUser	821	xxxx	((788)123-4561
SideChannelUser	1050	xxxx	((789)123-4560
SideChannelUser	1051	xxxx	((710)123-4561

RLS

- Rows returned vary based on the user
- Non-prescriptive
  - Defined by business rules
  - Can vary by table
- Impacts all users, including admins and service accounts
- Implementation
  - Access predicate
  - Security policy



USER_NAME()	SupplierID	SupplierName	SupplierCategoryID	PrimaryContactPersonID
dbo	1	A Datum Corporation	2	21
dbo	2	Contoso, Ltd.	2	23
dbo	3	Consolidated Messenger	6	25
dbo	4	Fabrikam, Inc.	4	27
dbo	5	Graphic Design Institute	2	29
dbo	6	Humongous Insurance	9	31
dbo	7	Litware, Inc.	5	33
dbo	8	Lucerne Publishing	2	35
dbo	9	Nod Publishers	2	37
dbo	10	Northwind Electric Cars	3	39
dbo	11	Trey Research	8	41
dbo	12	The Phone Company	2	43
dbo	13	Woodgrove Bank	7	45

USER_NAME()	SupplierID	SupplierName	SupplierCategoryID	PrimaryContactPersonID
SideChannelUser	4	Fabrikam, Inc.	4	27
SideChannelUser	5	Graphic Design Institute	2	29

Bypassing security

Attack and leak categories

- Indirect attacks
  - Leaked data, differing security, unintentional or misconfiguration
- Direct attacks
  - Intentionally removing or bypassing security
- Side channel attacks
  - Queries crafted to expose protected data

Indirect attacks

- Pre / post load (data is exposed in other sources)
  - Load files / export files / reports / data governance tools / lower environments / logging
- Admin tools / elevated access
  - Backups
  - Index details
  - Statistic details
  - Trace / extended events
  - Query plans
  - Query store

Direct Attacks

- Remove / change security
  - Disable security function (RLS)
  - Change access predicate (RLS)
  - Alter table definition (masking)
- Alter authorization
  - Add user to authorized group (RLS / masking)
  - CONTEXT_INFO()
  - SESSION_CONTEXT without READ_ONLY
- Bypass security
  - DBCC PAGE
  - DBCC SHOW_STATISTICS
  - sys.dm_db_stats_histogram

Demo – leaks and direct attacks

Side channel attacks

- Use allowed behavior in an unexpected way
- Derive data based on specific behavior
  - Carefully crafted queries
  - Aggregate functionality
  - Metadata from error conditions

Demo – Side channel attacks

Capturing RLS attacks

- Trace / Profiler
  - On-prem only
  - Error 8134
- Extended Events
  - Error 8134

Demo – capturing attacks

Mitigations

- Organizational stance
- Training and documentation
- Lock down users
- Monitoring / alerts
  - CPU usage
  - Error #8134 – divide by zero
- Code reviews
- Permanent data obfuscation and sub setting data
  - Lower environment mitigation
  - Complicated – use tools and established patterns

Mitigations - Continued

- Encrypt sensitive data
- Limit admin accounts
- Leverage data governance / discovery tools
- Audit RLS / masked column rules versus expectations
- SESSION_CONTEXT() READ_ONLY – not CONTEXT_INFO()
- Not all attacks will trigger an error

Demo - Mitigations

Alternatives to masking / RLS

- Encryption
  - Column level (EncryptByKey)
  - Application implementation
- Stored procedures
  - No direct table access
  - RLS rules still apply
- Views
  - Limit columns returned
  - Custom column logic
- Reports
- Application-level security
- Data obfuscation / static data masking for non-prod

Use cases

- Layers within stored procedures
- No direct access
  - Application layer
  - Session context
  - Execute as user
- Highly trusted users
  - Already have access
  - Extra layer of protection
- Simplified report development
  - Understand potential issues
- Actively monitor

Anti-patterns

- PII
- HIPAA protected data
- Credit card / banking information
- Any data that absolutely can’t be leaked
- Non-prod databases
  - Devs have elevated rights and can remove or alter masking / RLS

Final thoughts

- Security in layers
- Hire trusted admins / developers / analysts
- Careful planning is key
- Query engine behavior could change
- Direct data access is a risk
- Use the correct solution for your problem
  - Security design / restricted access
  - Encryption
  - Reporting layer
- Monitor for breaches

Ben Johnston

https://linkedin.com/in/ben-johnston-41609011
https://www.red-gate.com/simple-talk/author/ben-johnston/
https://github.com/BenSQL/SQLSideChannel

I appreciate your attendance.