



SUPPORTING HIGH-SECURITY POSTGRESQL

HANS-JÜRGEN SCHÖNIG

ABOUT CYBERTEC



Wiener Neustadt
AUSTRIA



Tallinn
ESTONIA



Zurich
SWITZERLAND



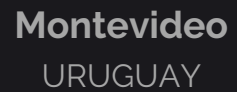
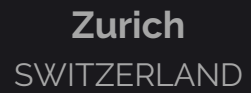
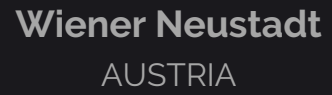
Montevideo
URUGUAY

DATA Science

- Artificial Intelligence
- Machine learning
- BIG DATA
- Business Intelligence
- Data Mining
- Etc.

POSTGRESQL Services

- 24x7 Support
- Training
- Consulting
- Performance Tuning
- Clustering
- Etc.



CYBERTEC Worldwide



WILLHABEN.AT®



Audi

 SBB CFF FFS

hims



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna | Austria

Rappi



TARGET



Auswärtiges Amt

NOVOMATIC

TOMTOM 

NOKIA
Connecting People

SIEMENS



Lufthansa



BOSCH



 **Bank Austria**
Member of  UniCredit

 **ncs**
making IT happen

CYBERTEC
DATA SCIENCE & POSTGRES SQL

Customer Sectors

University 

Automotive 

Government 

Industry 

Electronic 

Finance 

Trade 

Etc.... 



SECURITY REQUIREMENTS OF THE FUTURE

REQUESTS WE SEE IN THE FIELD

- Easy configuration and maintenance
- Checking security status
- PostgreSQL Encryption
- Anomaly detection



EASY CONFIGURATION
AND MAINTENANCE

WHAT SUPPORT PEOPLE SEE IN THE FIELD

Some people struggle to configure permissions properly

- Setting is the easy part
- “Checking” is the hard part
- Getting an overview is hard!

CHECKING SECURITY STATUS

- Extracting a full overview in PostgreSQL is not easy
- \z is ...
 - well, not so user friendly

pg_permission: COMING TO THE RESCUE

Freely available at:

- https://github.com/cybertec-postgresql/pg_permission

pg_permission can do two things:

- Provide overview
- Calculate the difference between “is” and “should be”

pg_permission: OVERVIEW

```
test=# \d all_permissions
```

```
View "public.all_permissions"
```

Column	Type	Collation	Nullable	Default
object_type	obj_type			
role_name	name			
schema_name	name			
object_name	text	C		
column_name	name			
permission	perm_type			
granted	boolean			

Triggers:

```
permissions_trigger INSTEAD OF UPDATE ON all_permissions  
FOR EACH ROW EXECUTE FUNCTION permissions_trigger_func()
```

pg_permission: REACHING TARGETS

```
INSERT INTO public.permission_target
  (role_name, permissions,i
   object_type, schema_name)
VALUES
  ('appuser', '{USAGE}',
   'SCHEMA', 'appschema');

...
INSERT INTO public.permission_target
  (role_name, permissions,
   object_type, schema_name, object_name)
VALUES
  ('appuser', '{USAGE}',
   'SEQUENCE', 'appschema', 'appseq');
```

pg_permission: REACHING TARGETS

```
SELECT * FROM public.permission_diffs();
```

missing	role_name	object_type	schema_name	object_name	column_name	permission
f	laurenz	VIEW	appschema	appview		SELECT
t	appuser	TABLE	appschema	apptable		DELETE

(2 rows)

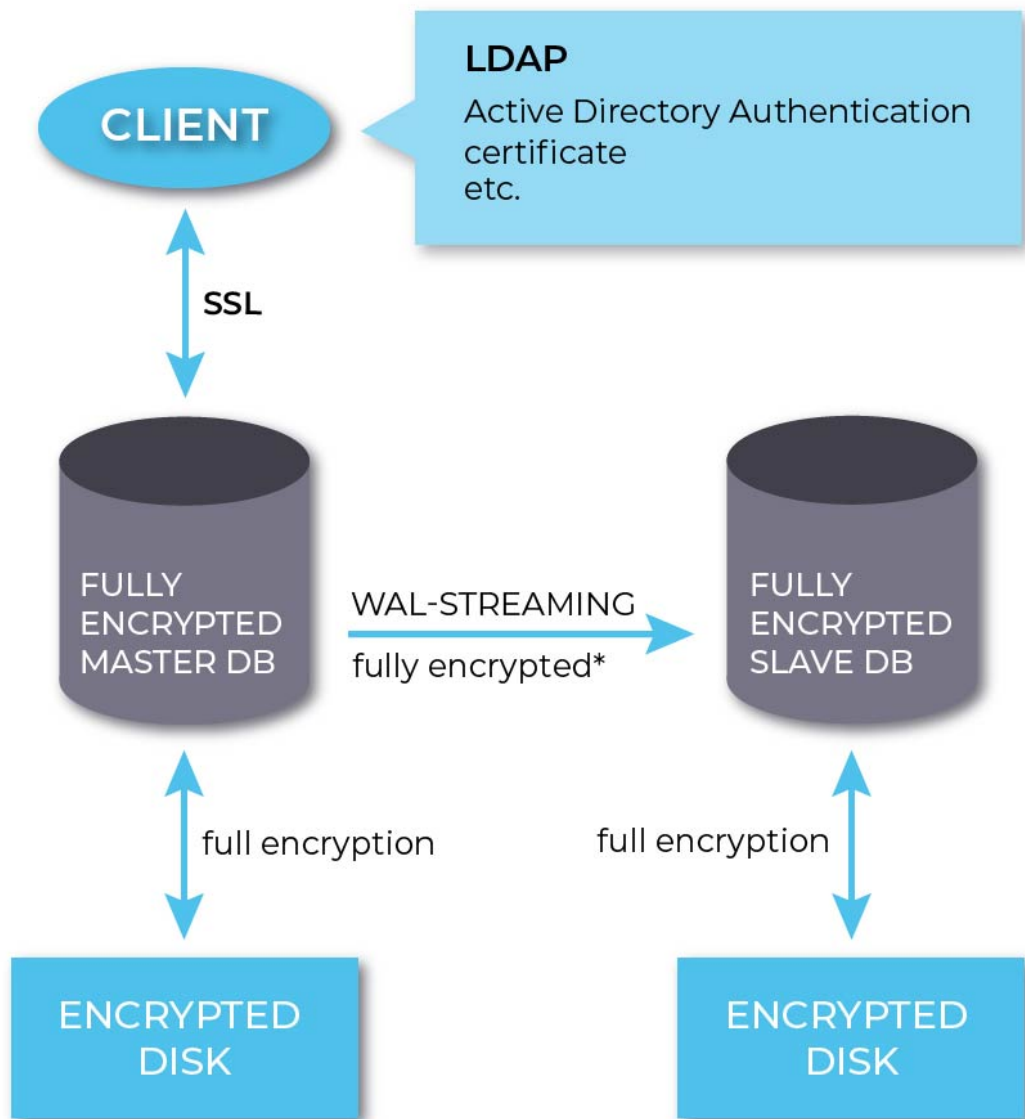
PostgreSQL Encryption

- Standard PostgreSQL stores data on disk in a non-encrypted way
- Many companies need that
 - Sometimes legal requirements
 - Sometimes company requirement

PostgreSQL TDE: Transparent Data Encryption

- We developed a solution for a client
- We will try to push it into PostgreSQL 13+
 - Still major discussions on mailing list
 - We cannot wait





*WAL-STREAMING with SSL

HOW POSTGRESQL TDE WORKS

VERSIONS AVAILABLE

Code for PostgreSQL 9.6 available on our website

- www.cybertec-postgresql.com

PostgreSQL TDE 12 will be Open Source

- released in November
- Includes hardware acceleration

FULL SUPPORT AVAILABLE

- 24x7 support available
- We provide consulting
 - Help with key integration
 - Integration
- Services available world-wide

ANOMALY DETECTION

- Is there strange user behavior?
 - Maybe DDLs at unusual times?
 - Maybe strange changes in data?
- CYBERTEC can detect that
 - Helps to detect intrusion

CEO Hans-Jürgen Schönig

MAIL hs@cybertec.at
PHONE +43 2622 930 22-2
TWITTER @postgresql_007



WEB www.cybertec-postgresql.com
TWITTER @PostgresSupport



ANY
QUESTIONS?

