Secure PostgreSQL 11 Deployments

NYC PostgreSQL User Group
New York

Magnus Hagander
magnus@hagander.net

Magnus Hagander

- Redpill Linpro
  - Principal database consultant
- PostgreSQL
  - Core Team member
  - Committer
  - PostgreSQL Europe

PostgreSQL 11

Secured?

- What does it mean?
- Can it be done?

PostgreSQL

- Provides a toolbox
- You don't need everything
- Maybe you don't need anything...

Securing what

- Environment
- Communication
- Authentication
- Access

Securing what

- Environment
- Communication
- Authentication
- Access

Environment

- Only as secure as the environment
- If someone owns the OS, they own the db
  - Owns the server = owns the OS
  - Owns the datacenter = owns the server
- Defined trust levels!
  - e.g. outsourcing/cloud vendors

What's the most secure OS?

- The one you know!

PostgreSQL installation

- Use packages!
  - RPM, DEB etc
- If not possible, use installers

Keep updated!

- Platform/package update management
- Not *just* PostgreSQL!
- Monitor!

Storage encryption

- No native PostgreSQL solution
- Full disk encryption
  - What about keys?
- VM level encryption
  - (keys again)
- What's the threat model?

Where to host?

- On-prem?
- Co-located?
- Outsourced?
- Cloud (aka outsourced)?
- DBAAS?

Securing what

- Environment
- Communication
- Authentication
- Access

Securing communication

- (physical)
- VM/software defined
- VPN
- ipsec
- SSL

SSL vs TLS

- PostgreSQL speaks TLS
- But we call it SSL
- (SSLv2 and SSLv3 are forbidden)

SSL in PostgreSQL

- OpenSSL only (for now)
- Certificate/key
- Like any other service
- *Disabled* by default on server
- *Enabled* on client!!
  - But wihout verification!

Enabling SSL

- Add cert and key
- Don't use snakeoil!
- Also don't use LetsEncrypt..

Enabling SSL

```
ssl=on
```

SSL negotiation

- Server provides SSL
- Client decides what to use
- Server potentially rejects choice

Server SSL control

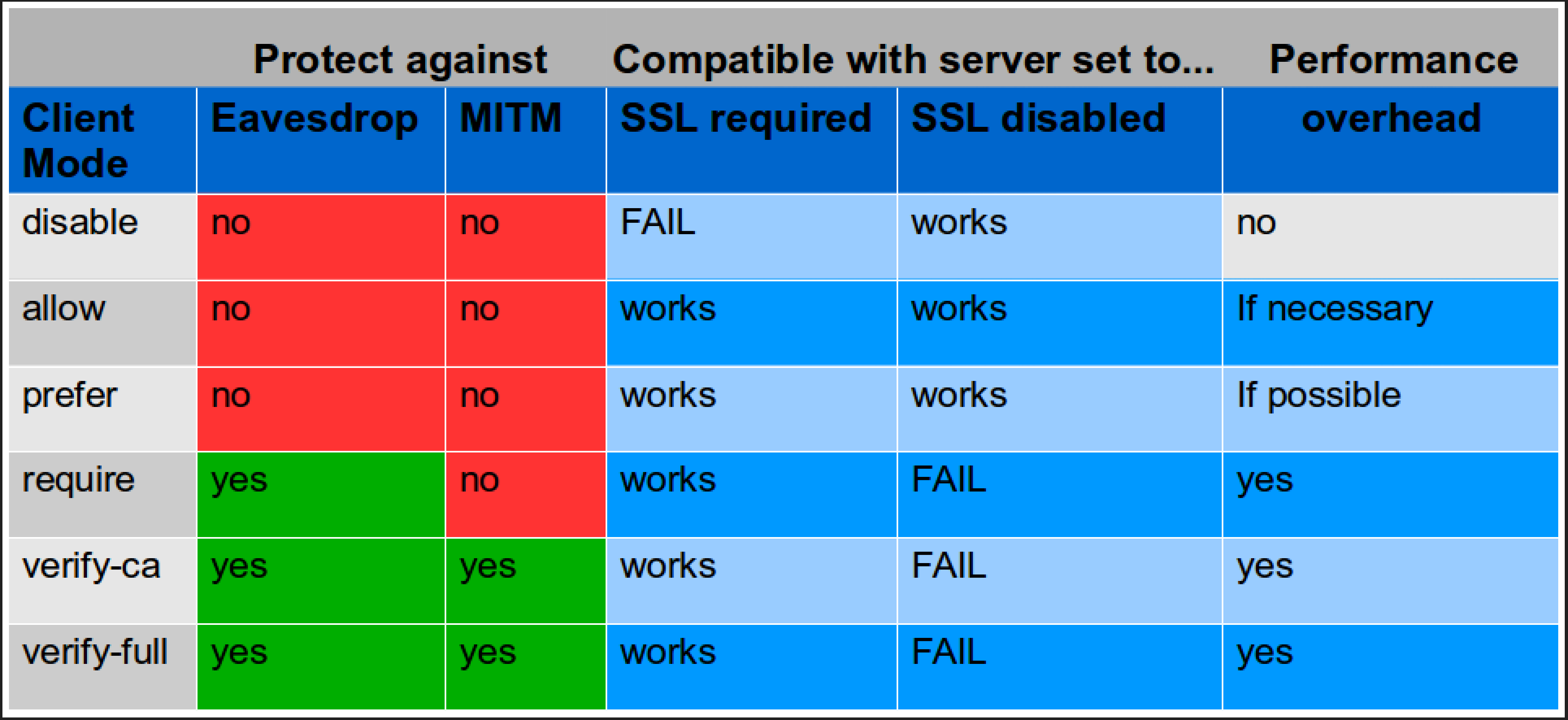
- pg_hba.conf
- Can cause rejects and retries

```
hostssl all all 10.0.0.0/24 scram-sha-256
...
...
hostnossl all all 0.0.0.0/0 reject
```

Client SSL control

- sslmode
  - disable
  - allow
  - prefer
  - require
  - verify-ca
  - verify-full

Client SSL control



	Protect against		Compatible with server set to...		Performance
Client Mode	**Eavesdrop**	**MITM**	**SSL required**	**SSL disabled**	**overhead**
disable	no	no	FAIL	works	no
allow	no	no	works	works	If necessary
prefer	no	no	works	works	If possible
require	yes	no	works	FAIL	yes
verify-ca	yes	yes	works	FAIL	yes
verify-full	yes	yes	works	FAIL	yes

Client SSL root

- PEM format
- ~/.postgresql/root.crt

Client certificate

- Optionally required for connection

```
hostssl all all 10.0.0.0/24 scram-sha-256 clientcert=1
```

- Or used for authentication

```
hostssl all all 10.0.0.0/24 cert
```

Client certificate

On client

- PEM format
  - ~/.postgresql/postgresql.crt
  - ~/.postgresql/postgresql.key
- Or OpenSSL engine

Securing what

- Environment
- Communication
- Authentication
- Access

How to log user in

- Many options
- Local or remote

Integrated authentication

- GSSAPI
  - Use instead of LDAP!
- Cert
- RADIUS

SCRAM

- Added in PostgreSQL 10
- Secure local password management
- Both auth and storage

SCRAM

pg_hba.conf

```
# IPv4 local connections:
host     all    all   127.0.0.1/32   scram-sha-256
# IPv6 local connections:
host     all    all   ::1/128        scram-sha-256
```

SCRAM

Hash storage

```
postgres=# \password kalle
Enter new password:
Enter it again:
postgres=# SELECT passwd FROM pg_shadow WHERE usename='kalle';
               passwd
-------------------------------------
 md563de8bd81c3d9b70b49308f0b0d5f74c
```

SCRAM

Hash storage

```
$ psql -h localhost -U kalle postgres
Password for user kalle:
psql: FATAL:  password authentication failed for user "kalle"
```

In log

```
DETAIL:  User "kalle" does not have a valid SCRAM verifier.
```

SCRAM

Hash storage

```
postgres=# SET password_encryption = 'scram-sha-256';
SET
postgres=# \password kalle
Enter new password:
Enter it again:
postgres=# SELECT passwd FROM pg_shadow WHERE usename='kalle';
               passwd
-------------------------------------
 SCRAM-SHA-256$4096:lLZZzENeAJypMXvLIKDJpQ==$K1vyYpVZuMZd13uP4AXtC
```

SCRAM

Old clients

```
9.5$ bin/psql -h localhost -U kalle postgres
psql: SCRAM authentication requires libpq version 10 or above
```

- libpq (layered): 10
- JDBC: 42.2.1 (Jan 2018)
- NPGSQL: 3.2.7 (Feb 2018)

SCRAM

Channel binding

- New in 11
- Ensures authentication server is same as SSL server
- Currently cannot be enforced

Securing what

- Environment
- Communication
- Authentication
- Access

Securing access

Securing access

- Please stop using superuser!
- 10 made it easier, 11 slightly more so
  - pg_read_server_files
  - pg_write_server_files
  - pg_execute_server_program

search_path

Be careful!

Object creation

- Object creation by untrusted users dangerous
- Can shadow "proper" object by superuser
- Always qualify schema for high priv users

Object creation

- Writable schemas in search path
  - public
- Change default schema search path
- Revoke permissions on *public*

Thank you!

Magnus Hagander
magnus@hagander.net
@magnushagander
https://www.hagander.net/talks/