

5 Things to Include on Your PostgreSQL Security Checklist

Securing data is mission-critical for the success of any enterprise, as well as for the safety of its customers. This PostgreSQL Security Checklist infographic presents a framework and a series of recommendations to secure and protect a PostgreSQL database.

The following five checkpoints encompass physical security, network security, host access control, database access management, and data security.

1

Securing Access

- ✓ **Physical Access**
Secure Facility, CCTV Monitoring, Locking Racks
- ✓ **Connecting (Unix Domain / TCP/IP)**
Remote connection, Securing direct remote attacks
- ✓ **Firewall (Local/Cloud)**
Unauthorized access to ports, inbound and outbound rules
- ✓ **Transport Encryption**
Encryption of traffic through networks

2

Securing authentication

- ✓ **pg_hba.conf Trust**
Server connection with no further authentication, to use with caution
- ✓ **pg_hba.conf Peer & Ident**
Authenticating by the underlying operating system
- ✓ **pg_hba.conf Md5 vs. SCRAM**
Prevent sniffing, and store hashed passwords on the server
- ✓ **pg_hba.conf LDAP vs. Kerberos**
Implementation of Single Sign-On (SSO) systems
- ✓ **pg_hba.conf TLC Certificates**
Authenticate automated systems that need to connect across the network to a Postgres server
- ✓ **authentication_timeout**
The maximum amount of authentication time allowed before the server closes the connection
- ✓ **Auth_delay**
Restrict brute force attacks

3

Securing roles

- ✓ **Password Complexity**
C development or use an external identity service
- ✓ **Password Profiles**
Ensure users maintain strong passwords
- ✓ **SET ROLE**
Using NOINHERIT keyword when creating a role
- ✓ **Monitoring Roles**
Grant specific privileges to roles that are used to monitor the system

4

Data access control

- ✓ **ACLs**
Make use of group roles to simplify privilege management for individual login roles
- ✓ **GRANT & REVOKE**
Give the group roles the minimum level of privilege required to work
- ✓ **RLS (Row Level Security)**
Define policies that limit the visibility of rows in a table to certain roles
- ✓ **Views**
Restrict the ability to select from the underlying tables, and have to access the data from the view instead
- ✓ **Security barriers**
Ensuring that the function never sees the hidden rows
- ✓ **Security Definer Functions**
Provide specific functionality to roles that cannot perform those tasks directly themselves
- ✓ **Data redaction**
Hide specific pieces of sensitive information from users

5

Securing encryption

- ✓ **Pgcrypto**
SQL functions for encryption and hashing that can be utilized as a part of the logic in database design
- ✓ **Key management**
Store keys in a secure service separately from the database and application
- ✓ **File system and Disk**
Protect against physical attacks on non-running hardware

This infographic is intended as a comprehensive overview that will help you examine the security of your PostgreSQL deployment from end to end.

To learn more best practices for PostgreSQL security, download our whitepaper!

[Get the Whitepaper](#)