Oaxaca Online-Local Database Security Report

Aim: How to create a security system that secures the connection between the online and local database.

Online Database: AWS (Amazon Web Services) Server
Local Database: Postgres 13.5

*Please find the conclusions of this research on the second page.

Research Notes:
Possible Security Options: (UpGuard Inc., 2022).

- Encryption (Kerberos or Hash-Based columns use for values, such as passwords)
  (UpGuard, 2022) and (The PostgreSQL Global Development Group, 2022)
- Limiting the number of access points to the Restaurant (PostgreSQL) database according to the number of users (restaurant staff and customers). (UpGuard Inc., 2022)
- Use the latest software updates to ensure that the latest bugs and attacks cannot infiltrate the database. (UpGuard Inc., 2022)
- Password Access.
- Separating Datasets according to section access rights and needs. (UpGuard Inc., 2022)
- Creating a monitoring system of the datasets so that the data cannot be edited by mistake or without permission. (UpGuard Inc., 2022)
- User Access rights – via Privileges (The PostgreSQL Global Development Group, 2022)

Securing a PostgreSQL Database: (Gravitational Inc., 2021)

- Network-level:
  - Iptables – restricts access to only authorized ports, IP addresses, subnets.
  - Reverse Tunneling - 1 way connection from the host to the local database, thus, from the online database (server) to the local database (client).
  - Listening addresses – restricts user access rights to only (ports, IP addresses and subnet) connections prelisted into the database by the programmer.
- Transport-level:
  - Certificate Authentication
    Requires Oaxaca to obtain encryption certificates and requires our team to code the configuration of the authentication of the device to database via the use of the pg_hba.conf file
  - SSL mode validates the connection to the database via the use of verify_full and verify_ca classes
- Database-level:
  - Database role assignments according to the person's role in the company (management, restaurant staff or customer) using PostgreSQL privileges to allocate access rights. Examples of such

include: managementRole users (for Oaxaca restaurant managers and owner), newRole users (for managers and HR), everydayRole user (for restaurant staff) and customerRole user (for customers visiting the restaurant). User accounts can also inherit roles as and when the staff join and leave the company.
    - Row-level security access permissions – allocating generalized access permissions according to a user's level of seniority
    - Auditing – keeps logs of who logs in, what they do and timestamps it – done via a third-party extension (such as PostgreSQL pg_audit), however, the logs need to be saved. The extension can be found here: https://www.pgaudit.org/

The Oaxaca Security Conclusions:

- We should disable any remote SSH access to the Oaxaca database (in the pg_hba.conf file) and instead set up <local> databases on staff devices to enable them to use the database remotely without editing the master database on the online server and for the restaurant to continue business even when updates are being made to the system.

- We should set an auditing and tracking tool on the database (with the pg_stat_statements PostgreSQL extension) to monitor any unauthorised queries implemented (e.g., DELETE, DROP, UPDATE, or INSERT) on the online server main database. This will flag any unwanted actions done either accidently or maliciously.

- We should edit the PostgreSQL trust settings (found in the pg_hba.conf file) to use encryption, such as MD5 or Kerberos, that is accepted by PostgreSQL 13.5 (the one used for developing the Oaxaca system). So that the database security is not only reliant on password accessibility.

- We should keep on top of security by ensuring that we use the most current possible AWS server, PostgreSQL database updates. This will ensure that they system is properly protected against any previous, current, and future bugs and hacks that could both compromise the developed system and the Oaxaca company.

- We should set access rights using PostgreSQL privileges as well as horizontal access rights according to the role the user has, their hierarchal position and data access needs in the Oaxaca company. This would permit an additional layer of data security and would reduce accidental edits to the whole system.

- We should use 'listening addresses', so that only devices prelisted by the Oaxaca team can access the database. This would bring in an extra level of security to the database and the system. For this, we would need to write code to allow the management team to add access rights to the devices they use, as and when new devices are set up with the business.

References:

Gravitational Inc., 2021. *Securing your PostgreSQL Database.* [online]. April 2, 2021. Available at: < https://goteleport.com/blog/securing-postgres-postgresql/ >. [Accessed 20 January 2022].

The PostgreSQL Global Development Group, 2022. *Chapter 5 Data Definitions 5.7 Privileges.* [online]. Unknown. Available at: < https://www.postgresql.org/docs/13/ddl-priv.html >. [Accessed 20 January 2022].

The PostgreSQL Global Development Group, 2022. *Chapter 17 Configuration Options Kerberos Authentication.* [online]. Unknown. Available at: < https://www.postgresql.org/docs/6.5/config12739.htm >. [Accessed 20 January 2022].

The PostgreSQL Global Development Group, 2022. *PostgreSQL 8.1.23 Documentation, Chapter 16 Operating System Environment 16.6 Encryption Options.* [online]. Unknown. Available at: < https://www.postgresql.org/docs/8.1/encryption-options.html>. [Accessed 20 January 2022].

The PostgreSQL Global Development Group, 2022. *PostgreSQL 9.1.24 Documentation, Chapter 17 Server Setup and Operation, 17.9 Secure TCP/IP Connections with SSL.* [online]. Unknown. Available at: < https://www.postgresql.org/docs/13/ddl-priv.html >. [Accessed 20 January 2022].

UpGuard Inc., 2022. *How to secure your PostgreSQL Database – 10 Tips.* [online]. January 11, 2022. Available at: <https://www.upguard.com/blog/10-ways-to-bolster-postgresql-security >. [Accessed 20 January 2022].