MSSQL Security Model

What is MSSQL Security Model

- Imagine if you will, that you own a house.
- Because you are the owner, you have a key to the house.
- You can do anything you want in any room of the house.
- You decide who else gets keys, and what they can do inside.
- We’ll assume your spouse other also has a key and equal rights to all rooms.
- lets assume you have 2 kids (Suzie and John).
- Suzie and John also both have keys so they can get inside after school.
- However, they are not allowed into each other’s rooms, nor are they allowed to change the locks, paint the walls or use the stove.
- So, 4 people can enter the house.   2 can do whatever they want, 2 have some restrictions.   Simple enough?

What is MSSQL Security Model (contd..)

- In the SQL Server security model the above translates as follows:
  - House – an instance of SQL Server
  - Keys to the house – Logins
  - Mom and Dad – Logins with sysadmin role (God rights…)
  - Suzie and John – Logins without sysadmin
  - Rooms – databases
- You have to have a key/Login, to even open the door/connect to the instance.
- You need specific permissions to access a database/room.
- Bobby is not allowed in suzie’s room and vice-versa.
- Neither kid is allowed in dad’s man-cave, but mom is.
- Suzie and Bobby need to be granted access to use a database/room, so they are created as USERS in the database.

What is MSSQL Security Model (contd..)

- Mom and Dad do not explicitly need to be granted as USERS in any database, because they own the place and can do whatever they want at the server/house or database/room level.
- If Uncle Joe is coming for a visit, he would also get a key/LOGIN and access to the rooms/databases he will be using.   When he leaves, he gives his key back (USER/LOGIN removed).
- If you want, imagine John has a brother and they share a room. In their room there are two beds. They don't share and they don't switch. In this example the beds are database objects such as tables. You can grant or deny access down to that level as well.
- So, LOGIN is created first (can't go into a room until you get in the house).
- USER access for non sysadmins is created next at the database level.
- Inside the database, you GRANT or DENY permissions to objects in the room (if you choose).
- You can get very, very granular if your application/employer requires it.
- Login can be either Windows based or created within SQL Server depending on how the instance is configured and your security teams requirements.