All work by Brendan Banfield

Threats:

Spoofing:
-users could request info from other users accounts. Solution: user requests for nonpublic info require use rname and password

Tampering:
-information sent to the user could be changed by someone with network control. Solution: include an enc rypted hash of the data with the data
-a user could provide incorrect information on behalf of another user. Solution: make sure user authentica tion is required for changing their data

Repudiation:
-someone with network access could send the wrong data to a user. Solution: include a digital signature with all messages

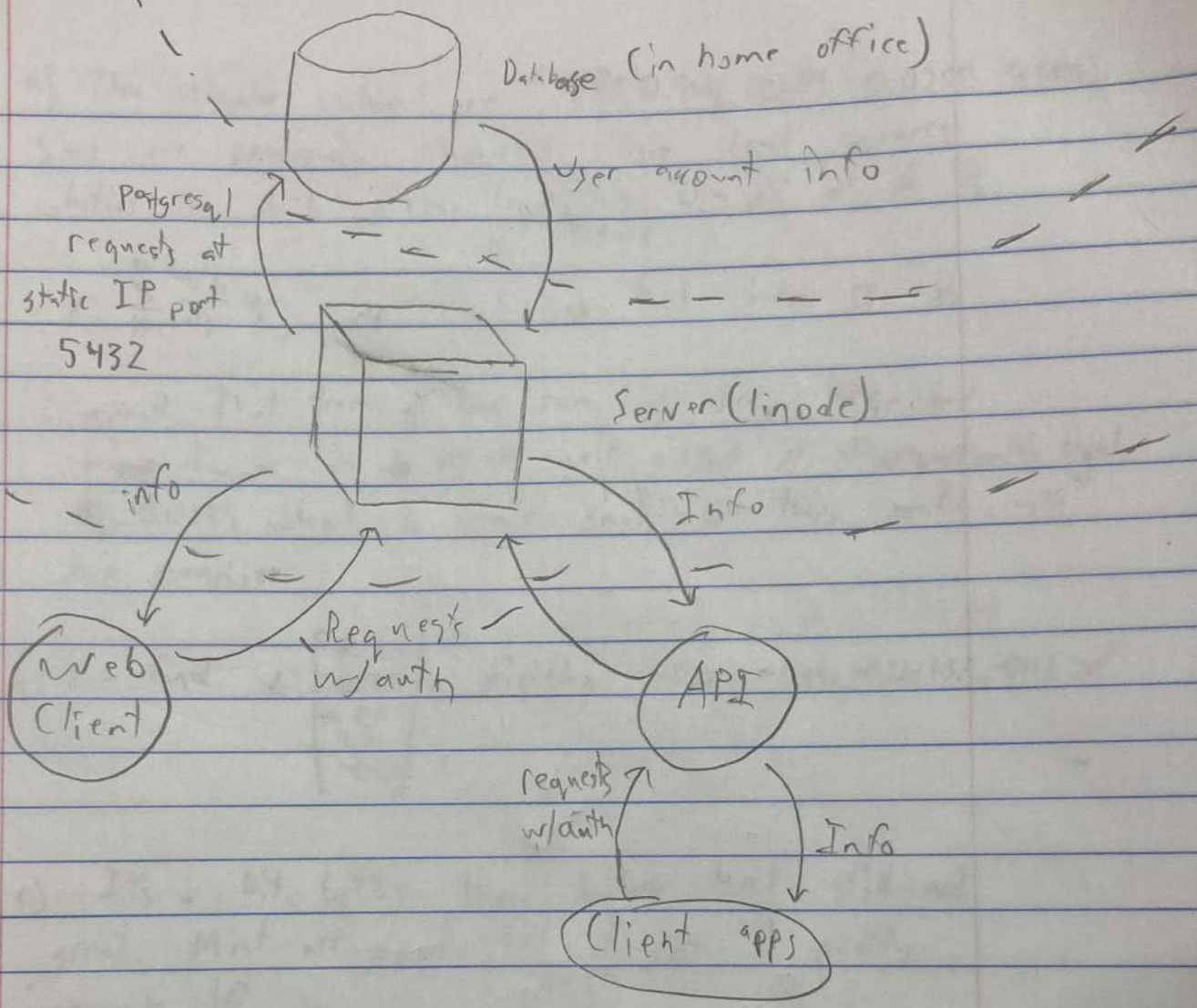
Information disclosure:
-users could listen to network traffic between the server and database. Solution: requests and responses should be encrypted
-if SQL injection or another method of database access is possible, highly sensitive information such as p asswords and credit card info are accessible. Solution: use a seperate server for sensitive info and never allow users to request from it
-network traffic to/from users could be read to see sensitive data. Solution: only allow HTTPS connections

-If someone gains access to your password database, they could try username and password combos on other websites. Solution: don't store passwords, only password hashes
-If someone gains access to your password hash database, they could calculate hashes of common pass words to quickly identify vulnerable users. Solution: salt all password hashes

Denial of service:
-users could spam the system with requests to prevent other users getting information. Solution: block req uests from IPs that have sent too many requests recently
-any exploit being found would require you to take down servers to fix it. Solution: extensively test your se rvices and look for potential threats, and address them asap

Elevation of priviledge:
-unauthorized user uses sql injections to read other users information. Solution: sanitize inputs, use para metrized sql commands
-someone guesses an admin password. Solution: require 2 factor authentication for admins and enforce ri gorous password requirements

Other/I can't decide:
-Someone could break into your house to physically access the database. Solution: have physical securit y around the database, or use a cloud hosting service which takes care of it for you


Database (in home office)
User account info
Postgresql requests at static IP port 5432
Server (linode)
info
Info
Requests w/auth
Web Client
API
requests w/auth
Info
Client apps