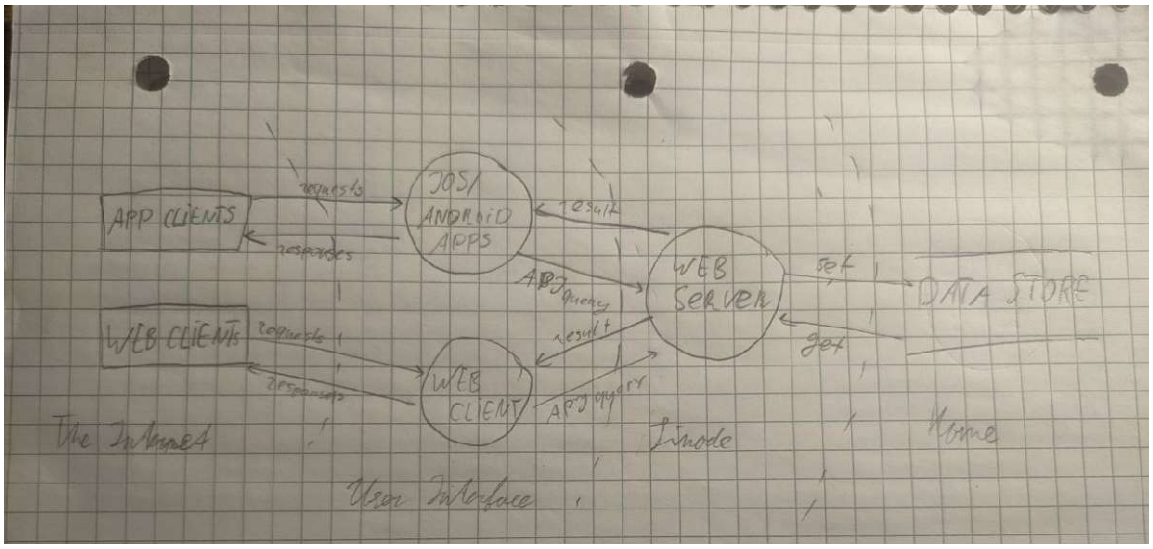
Viktor Chekhovoi



1. **Spoofing.** *Attack*: Mal pretends to be Alice and, from his device, requests the web client to retrieve all 18+ NSFW tapir fanfiction she wrote, for the purposes of blackmail. *Mitigation*: when a user logs in, establish a session token over HTTPS that confirms the user identity; this token should be attached to any request to the API query from the web client to the webserver (also over HTTPS to prevent eavesdropping). If the token isn't correct, an error should be returned. *Attack*: Mal sends an API query directly to the webserver to retrieve the fanfiction. *Mitigation*: require the web client/app to establish go through an HTTPS handshake and verify each other's identity before sending the API request. For efficiency, this connection should be closed if the client doesn't send any queries for a specified amount of time, so it doesn't have to be closed immediately after the API result is sent back.
2. **Tampering**. *Attack*: Mal sends a 'set' request to the database to change Bob's password to 1. More precisely, he requests the entry for Bob's hashed password is changed to "salt || Hash(1)". *Mitigation*: since this attack doesn't necessarily have to be limited to passwords (changing Bob's username to "Ihatetapirs1234" would also be harmful), communication between the webserver and the database server should be done over HTTPS, with both ends being required to prove their identity.
3. **Repudiation**. *Attack:* Chuck assumes control of Carol's account and uses it to retrieve her private message history. *Mitigation*: there are multiple ways for Chuck to get control of another user's account. The two most common ways would be phishing and password picking. Password picking could be mitigated by adding stricter requirements when creating a password, as well as requiring the users to change their password every once in a while. The threat of phishing could be mitigated by adding optional or mandatory 2FA (depending on the frequency of these kinds of attacks) and adding phishing awareness notifications to the web client/app.
4. **Information disclosure**. *Attack:* Eve watches the packets exchanged between the IOS app and the webserver to find vulnerabilities. *Mitigation*: all communication between applications and the webserver should be done over HTTPS
5. **Denial of Service**. *Attack*: Mal uses special software to spam the webserver with requests through a web client, resulting in much longer response times for all other users and the server

eventually crashing. *Mitigation*: limit the number of allowed API queries per minute client-side so that the server can't get easily overwhelmed. Also, require captchas when the user performs too many requests in a short amount of time.

6. **Elevation of Privilege.** *Attack:* Mal creates a fake session token that allows him to get access to Dave's post history. *Mitigation*: make the session token generation less predictable, for example, by making it contain the hash of a value only known to the intended user (like the password).