Lyhyesti

Projektissa on huomioitu kauttaaltaan seuraavat haasteet:

- Toimintavarmuus: palvelun tulee toimia luotettavasti myös suurilla käyttäjämäärillä sekä sietää pienen mittakaavan palvelunestohyökkäykset.
- Vakoilu: muiden ihmisten viestintää ei pitäisi voida salakuunnella.
- Käyttäjäoikeudet: ulkopuolisen ei pitäisi pystyä hankkimaan itselleen luvattomia oikeuksia tai lähettämään tekaistuja viestejä toisen henkilön nimissä.

Uskomme ohjelmiston olevan riittävän turvallinen käyttötarkoitukseensa nähden. Haavoittuvaisuuksia todennäköisesti löytyy, mutta kaikki ilmiselvät ongelmat olemme korjanneet, eikä kenelläkään todennäköisesti ole motivaatiota käyttää useita päiviä murtaakseen Järvenpään sosiaali- ja terveyspalveluiden chatin. Toimintavarmuudelle suurin uhka ovat samanaikaisuusongelmat. Vaikka olemme korjanneet kaikki löytämämme bugit, todellisessa usean käyttäjän samanaikaisessa toiminnassa voi syntyä tilanteita, joissa sovellus ei toimi oikein.

Esimerkkejä tietoturvan huomioimisesta

- Selaimen päivitys-nappia painamalla palvelin lähettää kanavan viestihistorian uudestaan. Jos joku lähettää esimerkiksi 1000 päivityspyyntöä sekunnissa, emme halua, että sovellus tekee 1000 raskasta tietokantakutsua ja lähettää viestejä 1000 kertaa. Sovellus toimii niin, että viimeistään ensimmäisen pyynnön jälkeen viestit on haettu tietokannasta muistiin siten, että ne ovat nopeasti haettavissa jatkossa. Viestejä ei myöskään lähetetä 1000 kertaa: ensimmäiseen pyyntöön vastataan heti, toiseen pyyntöön vastataan 100 millisekunnin viiveellä. Kun kolmas pyyntö tulee, se jätetään huomioimatta, sillä toiseen pyyntöön ei olla vielä ehditty vastata. Ainoastaan sellaiset pyynnöt jätetään huomioimatta, joihin on jo ajastettu vastaus. Siten suojaus ei myöskään aiheuta haittaa todellisille käyttäjille.
- Kaikki internetistä tulevat pyynnöt validoidaan ennen niiden suorittamista. Esimerkiksi viestin lähettäjän aitous varmistetaan ja kanavien kuuntelussa varmistetaan käyttäjän oikeus kuunnella kanavaa.
- Jos käyttäjä kirjoittaa JavaScript-koodia viestiin, sitä ei suoriteta muiden käyttäjien selaimissa (XSS), vaan viesti näytetään tavallisena tekstinä. Palvelin ei myöskään vahingossa suorita käyttäjän lähettämää koodia (esimerkiksi SQL-injektiona).
- On tyypillistä käyttää samaa salasanaa useaan palveluun. Olisi ikävää, jos yhteen palveluun murtautumalla saisi paljon salasana-käyttäjätunnus-yhdistelmiä, joilla voisi kirjautua muihin palveluihin. Tämän vuoksi salasanoja ei tallenneta selkokielisinä tietokantaan, vaan ne suolataan ja lopuksi kustakin suolatusta salasanasta lasketaan hajautusarvo, joka tallennetaan tietokantaan.

- Suojaus XSRF-hyökkäyksiä vastaan on toteutettu sekä osana Spring Securityä että käyttämällä userId:tä, jotka pidetään selaimen JavaScriptin suorituksen muistissa (mihin toisen sivuston skripti ei normaalitilanteessa pääse käsiksi).