The assignment is centered around Condution, an open-source task management platform. Data processing and querying is done on-device, but raw user tables and data is stored in a remote PosgreSQL server that's either officially hosted and supported or optionally on-prem hosted by the user.

1 | What are we protecting?

Because of the open-source nature of Condution, and the fact that all data processing code is done with the client application, the most important asset to actually protect is any user's data (tasks, perspectives, due dates, etc.).

This means creating a secure pipeline between our servers and that of the clients, that from self-hosted servers and that of their clients, and finally the clients themselves.

The scope of Condution is has too narrow scope to protect anyone but direct Condution user's data, including misuses of the client app or API but not including users of a third-party services with indirect API interfaces to Condution.

2 | Who are we protecting it from, and what are their motivations?

2.1 | Protecting from…

- Advertisers who may want to advertise based on notes of users
- Foreign companies and groups who are attempting to access privileged information
- Bad actors looking to leverage a "weak-point" in a user's security profile
- Well-meaning but security-unconscious users

2.2 | Not Protecting from…

- Organized, state-supported exploits
- Third party authorization and our hosting partners (AWS S3)
- Intentional, non-data leaking misuses (DDOS; creating fake accounts, etc.)
- Authorized third-party access without prior credential-sharing

As the on-site hosting and UI is the primary service that Condution is providing, it is not useful to divert time in preventing the intentional or accidentally shutdown of the reference server. Although, it is useful to use services like Cloudflare as a no-effort prevention for such attacks.

Also, user data is also locally backed up. Hence, it is possible to restore services by simply hosting another instance and restoring data.

3 | What methods of attacks do we prevent?

3.1 | Software Attacks

Software attacks mostly center around attacking the user-facing (both server and client) software and exposing its vulnerabilities.

- Auth pipeline password cracking
- Cross-site cookie sniffing
- XSS
- UI design injection to gain protected features
- Tampering with self-hosted server authentication UI

3.2 | Pipeline Attacks

Pipeline attacks mostly interfere with our DevOps pipeline to gain unauthorized access.

- Breaching of PAM on our pipeline (once it gets to AWS IAM, its their problem)
- Breaching keys and cookies for our deployment system
- Breaching access points to Continuous Delivery workflow
- Leaking signing keys

3.3 | Social Attacks

Social attacks work on hijacking user's information by exposing social vulnerabilities of the user.

- Security misinformation and "hacks" distributed over the internet (like self-XSS via the JavaScript Console)
- Weak passwords that exacerbate the problems above
- Issues with the storage of user passwords and cookies (i.e. accidentally committing all of `.config`, which would therefore contain Chromium cookies)

4 | What are the possible effects of these attacks

1. Digital or physical harm to users via leaking of privileged information
2. Digital or physical harm to those whom the users interact with (say, lawyers) via the leaking of privileged information
3. Loss of trust in the Condution platform and ecosystem
4. Breaking of data privacy laws like CCPA or GDPR

5 | What are the resources of the attackers

The attacker's resources are primarily limited.

Because of the fact that we are limiting our scope to non-organisational non-nation-state attacks, and the fact that the net value of user data — although important — is very low, the attacks will probably be limited in scope.

- Advertisers: the value of random user data is reasonably low, so they will likely only engage in low-effort systems-wide attacks but not much of specific, targeted attacks
- Foreign companies and groups who are attempting to access privileged information: the value of this is very high, depending what the informational content is. However, there are likely better sources of attacks than the partial and fragmented to-do list system
- Bad actors looking to leverage a ”weak-point” in a user’s security profile: this attack likely has very low effort due to the fact that this process casts a generally large net and exposes very shallow vulnerabilities
- Well-meaning but security-unconscious users: no value nor tangiable resources

6 | What are our resources?

Directly accessible resources are reasonably limited: have 8 engineers on all departments, limited cybersecurity experience, and only freely or cheaply available commercial tools for PAM, authentication, or security.

However, it is possible — in cases with pipeline leaks and database security breaches — to request emergency help from AWS and GCP. Due to the fact that our interests generally align with theirs with respect to user data safety, they already have systems in place to perform emergency triage and freeze data to protect security.

7 | What should we do?

1. Align and protect resources and partnerships with hosting services to allow emergency triage
2. Secure CI/CD pipelines and update services to authorized users, with two-factor PAM authentication
3. Implement warnings and education for password security (forcing users to choose more secure passwords, providing self XSS warnings in the console, etc.)
4. Create systems for data shutdown, reversal, and freezing in case of emergency and to comply with privacy laws
5. Scan for cookie signatures on the internet (via security services like Pingdom) to analyze for public leaks of user security information