Legal Name: Paraclete Health LLC
D/B/A (if applicable): Wisely Health
Company EIN: 37-2138848

(1) Do you maintain a HiTrust and/or SOC II Type II certification?
If yes: please attach a copy of the respective report and reply N/A to all questions below.
If no: please provide an answer all questions listed below:

No

(2) Do you ask users for their permission before sharing or selling their data with other entities?

We have no plans to share or sell our users data with other entities. If we were to do so, we would change our companies **terms and conditions** to reflect that and give our users 60 days to opt out of the sharing agreement with an option in their settings to turn off sharing.

(3) After a user revokes your access to their data, do you delete the user's historical data from your systems?

Yes

(4) In the event of a data breach and/or security incident, do you notify your users of such breach and/or event?

Yes

(5) Is all data exclusively used, accessed, and transferred in the United States of America?

Yes

(6) Is all of the data you maintain encrypted in transit and at rest in accordance with generally recognized encryption processes?

All data is encrypted in transit and at rest.

- All traffic that is ingressed into our system is sent over TLS 1.3 with an SSL certificate managed by AWS Certificate Manager as our authority.
- We use Amazon RDS with encryption enabled to provide this guarantee at rest.

(7) Do you perform regular security testing of your application and promptly remediate any discovered issues?

We conduct a security review on a quarterly basis. As the surface area of our product and infrastructure increases, we will revisit this cadence.

(8) Does your application require the creation of strong passwords and/or multi-factor authentication?

(9) Do you store the users' password/login credentials?

We store salted passwords when they are provided.

(10) Do you secure your applications in accordance with OWASP top 10 Web Application Security Risks?

1. **A01:2021-Broken Access Control**
   We operate on the principle of least privilege. User identifiers required for resource lookup are stored in the encrypted payload we send through the "token" cookie which we use to perform RBAC (e.g. users can only access their resources, employees can only access administrative information such as facility metadata)

2. **A02:2021-Cryptographic Failures**
   All external traffic into Wisely is encrypted with modern security protocols (i.e. TLS 1.3). Data is encrypted on its way to the server, to our DB and at rest. IV for passwords are generated via the bcrypt salt function and we have the ability to rotate keys when needed.

3. **A03:2021-Injection**
   We use Prisma as our ORM which performs type checking over query inputs. Even in the scenario where we have to send 'raw' queries to the database; the ORM still sends queries as prepared statements.

4. **A04:2021-Insecure Design**
   We conduct a security review on a quarterly basis. As the surface area of our product and infrastructure increases, we will revisit this cadence. During these reviews we examine changes in data flow and access controls in our documented user stories and perform penetration testing to identify any major gaps in our security proactively.

5. **A05:2021-Security Misconfiguration**
   Due to the small surface area of our application, there are only a handful of entities at the infrastructure level (e.g. AWS Buckets, Managed Policies, Security Groups) we have to monitor.
   - Ports are enabled via allowlist
   - S3 bucket access is consolidated via managed policies
   - SES emails sends are mediated via managed policies.
   - Connected compute resources for our database (RDS) are actively audited.

   Data for the deployed application is generated by and owned by Wisely meaning no default accounts currently exist. The company has a policy of logging errors internally and returning a string constant that was written by the developer ahead of time.

   Lastly, Wisely's backend is a JSON API server upon authorization we respond with the HttpOnly value set on our token cookie.

6. ## A06:2021-Vulnerable and Outdated Components
   The team categorizes dependencies in three categories: frontend dependencies (React), backend server (Express/Node.js) dependencies and infrastructure related dependencies.

   Frontend/Backend
   Both our backend and frontend are written in javascript and we utilize npm to perform installation and audits. We regularly update packages when fixes are available and perform risk/benefit analyses of remediating the issues ourselves.

   Infrastructure
   We are subscribed to AWS security bulletin feed and review updates on a weekly basis to determine if remediation is needed in our own stack (i.e. RDS connections, S3, SES & EC2)

   Besides AWS, we do not use any third party vendors within our infrastructure nor have we self-hosted open source platforms for large scale compute (e.g. Airflow/Luigi/Dagster).

7. ## A07:2021-Identification and Authentication Failures
   Wisely Sidesteps a majority of the attacks that try to exploit this risk by using passwordless sign in. Users can either sign in with Google SSO or alternatively have a one time link sent to their email address.

8. ## A08:2021-Software and Data Integrity Failures
   To demonstrate this I need to write up what our policy is around auditing our software stack and how we audit/address the vulnerabilities. This is probably a 1-pager at most since our stack is pretty small right now.

9. ## A09:2021-Security Logging and Monitoring Failures
   The system keeps a trailing log of all the times that users have been authenticated and accessed their resources from our website.

10. ## A10:2021-Server-Side Request Forgery
    Post request bodies are parsed against standardized protocol buffers to provide initial sanitization. We delegate the responsibility of further sanitization to our ORM when looking up resources stored in our DB.

    At the time of writing a trusted entity such as the server does not have additional read access to our other pieces of infrastructure (e.g. Email Service, S3, EC2 Orchestration) - nor do we yield information stored on local files back to the user.

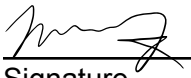
(11) Does your application meet the accessibility requirements outlined in the Americans with Disabilities Act?

Yes

By signing below you hereby expressly certify that all answers provided herein are true and accurate. In addition, you hereby expressly agree that any and all uses of the 1upHealth Services shall be subject to, and in accordance with, and you hereby expressly agree to be bound by, the terms and conditions attached hereto (Patient Access API Terms and Conditions.docx)

Wilson Kurian

Name



Signature

07/08/2024

Date