Aide

Security at Aide Support

Whitepaper

Aide.app, support@aide.app

April 3, 2024

Introduction

Aide provides a unified management layer over a variety of customer support operations. Aide may require read-write access to integrated information including customer support tickets, order information, shipping information, and custom integrations with proprietary client software.

Because of this varied data footprint, we understand our security responsibilities go beyond our own management dashboard. This document outlines the practices we've adopted to mitigate various kinds of security breaches known to threaten each part of our system, including unique risks for web applications, machine learning-based systems and third-party replications of client data.

Compliance progress

We are in the process of working with a security compliance automation platform to successfully complete a SOC Type 2 audit.

Threat models

Risk assessments. We periodically review each component of our system and record potential security risks and possible mitigation measures. This includes reviewing software libraries used, studying secure software engineering patterns, and determining the reliability of cloud services and our integrations.

Track record. We have been serving customers since January 2023 and have had 0 breaches to date. The threat model categories shared below are plausible threats that have not occurred to Aide.

In terms of concrete threat models, we focus on the following categories:

1. **Inter-client data leakage (error)**. Within our dashboard, there is a risk that specific endpoints or sessions don't properly isolate client data from other clients. To mitigate this, client data is effectively sharded per client id in our database and all endpoints filter data by client.

2. **External data leakage (malware)**. If malicious software gains access to any active sessions that have database access, there is a risk of a data leak. We mitigate this by logging database access and using well-trusted open source database client libraries (see **Software standards** section.)

3. **External data leakage (physical)**. The physical hard drives that store our data could be accessed by an unauthorized party and tampered with. We mitigate this risk with a

trusted cloud provider and at-rest data encryption (see **Data security** and **Physical security** sections.)

4. **External data leakage (personnel)**. This is a risk of an employee, contractor, or other privileged member of the Aide team purposefully leaking data. We mitigate this by providing access on an as-needed basis, vetting every team member and logging database access (see **Personnel** section.)

Network security

Firewall. All API endpoints for the dashboard and microservices are served by an Nginx server with an Azure-managed ingress controller. This provides SSL encryption, a load balancer, and a firewall.

Authorized endpoints. All API endpoints have authorization middleware, so requests are not served unless a known client or Admin account is logged into the Aide dashboard.

HTTPS. All web-based requests are encrypted via SSL/HTTPS. HTTP is not accepted and if attempted, the client is redirected to HTTPS. Additionally, all database connections in production and development are secured by TLS.

Data security

At-rest encryption. All database files, including backups and logs, are encrypted at rest with an Azure-managed key.

In-transit encryption. All communications over networks are encrypted with SSL.

In-use encryption. AES-256 is used to encrypt sensitive data such as OAuth keys in our database, so that even authenticated SQL sessions, such as by Admins, cannot easily access these keys.

Developer Admin access control

Developers are considered Admins if they need access to the production database and Kubernetes cluster. They have the privilege to develop on these resources and push changes to a production environment.

Whitelisted IPs. Each Admin is given a whitelisted office IP for where they work from. This means our resources can only be accessed from these IPs, and the correct credentials must also be provided.

Unique credentials. An Admin gets a unique set of credentials to access our resources that they are responsible for. This includes unique database username/password pairs, and an Azure account to access our Kubernetes cluster, using the az login CLI tool. Unique credentials means that admins can be deprovisioned easily.

Software standards

In the development of the service, Aide has adhered to the following standards and principles for secure software engineering:

Machine learning inference is sandboxed, less-privileged code. Libraries that provide the code and weights for machine learning model inference, such as transformer-based language models, are treated as less-privileged “remote code.” These libraries are updated often and have large numbers of contributors, meaning there is significant risk of malicious code hidden in otherwise useful source code. Our approach: in addition to initial safety and functionality tests of new models, we sandbox every model that we deploy to production. This means the model runs in its own process and has no access to Aide’s Resources. It receives text inputs and returns text outputs via secured inter-process communication.

SQL injection is mitigated with parameterized queries. Injection attacks (namely SQL injection) have been listed among the OWASP Top 10 risks for every year since 2010. Aide’s SQL database takes various forms of text-based input (such as adding training examples, storing chat messages), exposing it to this risk. Our approach: we require every SQL query to be a “parameterized query” which ensures data is treated as type-safe data and not as potentially dangerous injected code.

Sensitive data are stored as secrets. Aide integrates with many relevant software tools, and part of that integration involves the receipt of API keys and other private IDs. There are also private encryption keys used to encrypt data at rest. If these sensitive data were stored in our code repository or in internal docs, there’s a risk that these keys and IDs could be misused internally or leaked to be misused externally. Our approach: All sensitive data are stored as “secrets” rather than as plaintext or in code. This means that even privileged engineers do not have easy access to sensitive data. In practice, these data are stored as specially-made secrets with our version control provider and as Kubernetes secrets.

Database backups. Azure for PostgreSQL provides daily backups going back 7 days which can be used to restore data in the case of user or provider-side human errors that lead to data loss.

Personnel

Access on an as-needed basis: Access is granted to data on an as-needed basis. Our database can only be connected to from a few whitelisted IPs and with a unique password.

Vetting: Aide performs annual background checks for all employees and contractors.

Information security policy: All employees read and agree to an information security policy.

Physical security

Aide does not store any data on its own servers. Instead, all the data is kept in resources hosted by Microsoft Azure Cloud Computing Services. There are two resources that make up Aide's services:

1. **Database**. All data is stored in Azure Database for PostgreSQL with automated backups. It is not replicated anywhere else.

2. **Kubernetes cluster**. All APIs and background services are implemented by an Azure Kubernetes Service (AKS) cluster.

Azure details. Both Azure resources are located in Azure's US East data center, which is in Boydton, Virginia. Azure's cloud service is world-class for security and has satisfied the following compliance standards: ISO 27001, ISO 27018, SOC 1, SOC 2, SOC3, FedRAMP, HITRUST, MTCS, IRAP, and ENS. For more information about security in Azure, see https://azure.microsoft.com/en-us/explore/security.

Data retention

Easy deletion via database sharding. All client-specific data are stored in shards such that it is easy to access and delete all client-specific records from all database tables. This includes customer support tickets, users, training examples, and any Aide configuration.

Compliance endpoints with integrations. Some integrations, such as Shopify, require compliance with their data retention policies, and we are compliant. If data redaction is requested through such an integration, we delete data within our retention time window.

Retention time. All data is deleted within 30 days of a contract ending, or upon client's request.