Keiji AI Cybersecurity & Development Policy

Update Date: Jan 28, 2024

Summary

This is the documentation of Keiji AIs cybersecurity and infosec practices. It is a living document that should be critiqued and updated regularly.

Risk Profile

Critical Assets:

Customer Data

Customer conversations with TrialMind
Customer queries, and agent responses to customer queries generated by the TrialMind system. This includes customer request metadata, such as which agent is being queried, and interaction details like code highlights.

Customer requests to TrialMind Programmatic Interface (TrialMindAPIs)
Requests to the programmatic TrialMind interface are considered sensitive. The content of all HTTP body requests should be protected.

Customer provided data sources (documents, tables, datasets)
Documents which users submit to TrialMind accompanying a query, analytic request, or for any processing are considered sensitive information. Any customer provided datasets,

Encryption during transit, encryption at rest, and tenancy configuration specifications of the customer must be complied with at all times for these critical data assets. These assets should also be backed up regularly, in the case of an outage or emergency so that there is no risk of data loss.

Systems and Microservices

API Gateway
The API Gateway is the primary entry point for the TrialMind web application backend, and TrialMindAPIs. The gateway service is responsible for request authentication with 3rd parties, and for determining user permissions using the role based access control list (RBAC List).

RBAC List (Access Control Authority) Service
The RBAC List Service relates user identifiers to specific application level permissions, and organization tenancy configurations.

Customer Tenancy Permissions List
The customer tenancy permission list maps the user identifiers to their organization, and the organization to a tenancy identifier. Using this list customer requests are mapped to the appropriate customer environment for fulfillment.

TrialMind Application Worker Pool
The TrialMind application worker pool is a horizontally scaled service which contains workers to fulfill customer requests.

Curated Data Sources
Curated data sources are a collection of search indices, and persistent storage volumes which contain Keiji AI's curated data sources. These sources are accessed by TrialMind workers to fulfill customer queries.

Core Security Principles

Principle of Least Privilege: Access to all systems by developers is limited to the bare minimum they need to complete their job. Privileges are granted to users by the engineering leadership team **only** (Chief Architect, Chief of AI).

Defense In Depth: Layered security measures are applied to all architecture components, to protect data and resources.

Regular Updates: The engineering team must ensure all software and systems are regularly updated to patch vulnerabilities. All company members are responsible for reporting identified vulnerabilities, or risks to critical assets immediately. Updates are performed promptly to mitigate unwanted breaches.

Employee Training and Awareness

Security Trainings

Regular Sessions are conducted yearly on best security practices, recognizing phishing attempted, and safe internet usage. Engineering Leadership is responsible for completing appropriate certifications each calendar year

AWS Security Essentials	https://learn.acloud.guru/course/4653d0ed-8896-42af-988f-756d4bcf583e/dashboard
Social Engineering Attacks	https://www.youtube.com/watch?v=QUgLxll_P58
Phishing Examples	https://security.berkeley.edu/resources/phish-tank
Phishing, NIST	https://www.nist.gov/video/youve-been-phished

Clear Communication

Security incidents must be clearly communicated to the engineering team as soon as they are identified. To effectively communicate issues, utilize the communication channels:

Keiji vulnerability slack channel (public): #infosec (keiji-ai.slack.com)
Keiji infosec email (public and internal): support@keiji.ai

Data Protection and Privacy

Encryption

All sensitive data is encrypted at rest, using a production ready encryption protocol. The current approved encryption protocols are: AES-256. Sensitive data is encrypted during transit.

Data Minimization

Only collect and retain necessary data. Data which is considered necessary:

Customer uploaded documents, and document derivatives*
Customer uploaded tables, and table derivatives*
Customer queries, and their accompanying responses*
Customer authentication logs
Customer usage logs of the API and web application

*customer data must be stored according to their tenancy configurations. For example, customers which are part of the sole-tenancy billing tier, must never have their data stored in a manner which violates our agreement.

Data Access Controls (Customers)

Customers are granted access based on their role within their assigned billing organization. Keiji AI implements endpoints for billing organization owners to clear data stored in the Keiji AI infrastructure through the dedicated TrialMindAPIs endpoint.

Data Access Controls (Developers)

Keiji AI developers are granted the minimum permissions required to complete their job. Only the leadership team have administrative rights to grant permissions to other users within the AWS IAM organization.

Authentication

Customer request through the web application and the programmatic interface (TrialMind, and TrialMindAPIs, respectively) is only granted if the request holds a valid session token. Session tokens must be acquired by authenticating the user with a 3rd party (using OpenID + OAuth 2.0).

Authenticated user requests are routed to appropriate tenant environments and granted access to resources based on the RBAC list (available through the Access Control Authority).

Incident Response Plan

The incidence response plan has 3 stages: Preparation, Detection and Reporting, Response and Recovery. This plan describes the procedures and approach taken to preparing, managing, and remediating incidents.

Preparation

Before the incident.

- Keiji AI engineering staff must be assigned to the incident response team
    - Currently: Benjamin Danek, Zifeng Wang, Jimeng Sun
- The engineering staff should be knowledgeable about our services, including infrastructure, and nuances of application logic.
- Relevant services, and databases have regular snapshots configured, as well as backups.

Detection and reporting

During the incident.

- Proper security monitoring must be in place on any production system, ensuring that SLA is monitored
- Notifications are triggered by the production system deviating from standard behavior, triggering an investigation
- Customers and users have a clear method for indicating issues with the service.
- The incident response team knows how to contact 3rd party vendors in the case of a vendor triggered incident.
- The incident response team has contacts for the relevant parties within customer organizations to communicate incident, and impacts.
- The incident response team should promptly resolve the source of the incident, and return the production environment to a healthy state.

Response and Recovery

After the incident has been contained.

- Any infrastructure which was augmented during the course of incident response should be returned to its normal state.
- Contact impacted customers regarding any lost or exposed data.
- Update the status page documenting the incident and impact towards SLA.
- Meetings retrospecting an incident should be conducted in a timely manner (under 2 weeks).
- As a part of retrospection, the team should determine.
    - The root cause of the issue.
    - What could have been done to prevent this issue from happening.
- Monitoring or architectural modifications should be made to prevent the issue from recurring.
- Write an internal report of the incident, documenting
    - Incident timeline
    - Root cause
    - Steps taken to resolve issue
    - Architectural or monitoring changes introduced
    - Customer impact

Secure Development Practices

Code Reviews

The engineering team conducts quarterly and code reviews of our end to end platform. In this process the engineering team must identify

- Possible vulnerabilities
- Points of failure for each software system
- Update threat model for the software component
    - Scenarios where an adversary may be able to exploit our critical assets.
- Develop a plan for remediating vulnerabilities promptly
- For systematic weaknesses
    - Promptly introduce backups of critical assets
    - Promptly introduce monitoring to ensure system health can be measured until the systematic weakness is remediated
    - Develop a concrete timeline to remediate systematic weakness. Commit to milestones to track progress, and assign a stakeholder to track the remediation timeline.
- Retrospect on what should have been done better to avoid introducing this vulnerability or systematic weakness in the first place.

Automated Testing

- Automated unit, and integration testing is used to guide application implementation, and identify application regressions.
- Automated vulnerability detection is applied to all repositories owned by Keiji AI to monitor for common vulnerabilities and exposures (CVE).

Security Ownership

Although the authority to grant access to critical assets is exclusively vested in leadership roles, the responsibility for secure development and compliance with security protocols is shared by every individual contributor (IC). Instead of relying on a singular policing authority as the custodian of security, robust cybersecurity emerges from the collective commitment and individual accountability of each engineer.

Infrastructure Audit Trail

Physical and Network Security

Access Control

Keiji AI's critical assets are all hosted in cloud infrastructure (AWS, Azure), and there is no physical infrastructure such as servers or data centers. Access to the Keiji AI infrastructure is only through the VPN tunnel.

Device Security

Access to any production infrastructure is granted only through the company VPN.

Vendor and Third-party Management

Vendor Assessment: Evaluate the security posture of third-party vendors or service providers.

Contractual Agreements: Ensure agreements include clauses for data protection and security incident reporting.

Regular Review and Update

Continuous Improvement

The security policies should be continuously reviewed, updated, and adapted to new threats and changes in the business.

Software Development Lifecycle

We apply an Agile framework in the development of our product. Our framework has 4 activities which we go through at least once per sprint (2 weeks), sometimes more.

Design

In the design phase, software engineers analyze requirements and identify the best solutions to create the software. Concrete activities include

- Identifying customer use cases, through user stories, and discussion with customer stakeholders
- Identifying performance expectations
- Identifying an implementation path, challenges, and tradeoffs of a proposed architecture
- Drafting system architecture, or modifications to the existing architecture
- Drafting user interfaces
- Identifying key players in the implementation
- Planning implementation from MVP, to iteratively converting MVP to a refined product
- **Identifying security priorities, and planning for their timely delivery in a backlog item sprint**

Implement

In the implementation phase, the development team codes the product. They analyze the requirements to identify smaller coding tasks they can do daily to achieve the final result.

- Develop scrum backlog items for the implementation
- Plan the order of implementation steps
- Identify milestones along the implementation path
- Follow an Agile plan to implement the proposed architecture, with regular check ins with the customer
- **Adhering to security principles during the implementation of our system**

Test
The development team combines automation and manual testing to check the software for bugs. Quality analysis includes testing the software for errors and checking if it meets customer requirements. Because many teams immediately test the code they write, the testing phase often runs parallel to the development phase.

- Implementing unit tests
    - **Including coverage for unauthorized requests, and malicious code, where applicable**
- Implementing integration tests
- Hands on QA
- Bug fixes, and product tweaks
- Updates to our design

Deploy
When teams develop software, they code and test on a different copy of the software than the one that the users have access to. The software that customers use is called *production*, while other copies are said to be in the *staging environment*, or testing environment.

Having separate build and production environments ensures that customers can continue to use the software even while it is being changed or upgraded. The deployment phase includes several tasks to move the latest build copy to the production environment, such as packaging, environment configuration, and installation.

Maintain
In the maintenance phase, among other tasks, the team fixes bugs, resolves customer issues, and manages software changes. In addition, the **team monitors overall system performance, security, and user experience** to identify new ways to improve the existing software. This stage happens in parallel to the previous 4, as we are actively developing in the test environment while maintaining our production services.

Service Level Agreement (SLA)

TrialMind (Chatbot)

Metric	Uptime Commitment (excluding maintenance windows)
Chatbot Uptime	99.99%

Chatbot Uptime = (Hours in time period chatbot UI is available / Hours in time period)

TrialMindAPIs (programmatic interface for analytic capabilities)

Metric	Uptime Commitment (excluding maintenance windows)
API Uptime	99.99%

API Uptime = (Hours in time period chatbot UI is available / Hours in time period)

In addition to SLA metrics, we plan to provide a comprehensive performance report for our TrialMind API and Chatbot.