

ALFIYA ABDIMUTALIPOVA

Enhancing Refracted Security Platform with AI Capabilities

2025

Table of contents

3	Company Overview
4	Collaboration Tools
5	Key Deliverables
6-11	Functional Sidebar Chat Helper
12-16	Observations Improvement Feature
17-25	Security Levels (Automated data gathering)
26	Challenges
27	Requirements for Integration
28	Project Planning
29	Technology Stack

Company Overview

Refracted Security

Specializes in vulnerability management for organizations.

Mission: Provide effective tools for assessing, addressing, and mitigating vulnerabilities.

REFRACTED
security



Collaboration Tools

The screenshot shows the Azure DevOps Boards interface. On the left, a sidebar lists various project management sections like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Artifacts. The main area displays a Kanban board for the "Internship - AI tooling integration Team". The board has three columns: To Do, Doing, and Done. Under the Done column, there is a list of completed tasks:

- 1383 Integrate Authentication (Done)
- 1379 AI Text Helper (Done) - assigned to Afiya Abdimalipova
- 1369 Update the Chat Histories feature to work per-user (Done)
- 1385 Add SonarCloud Code Checking (Done)
- 1368 Persist Chat Histories Server-Side (Done)
- 1367 Add a Chat History (Done)
- 1366 Implement the design for the Chatbot Sidebar Messages (Done)

Azure DevOps

The screenshot shows the Microsoft Teams interface. On the left, a sidebar lists "Your teams" and "Development - AI". The main area shows a private channel named "Development - AI". The channel header includes tabs for Posts, Files, and a blue selected tab for Posts. Below the header, there is a welcome message: "Welcome to the Development - AI private channel" and "Let's start the conversation.". At the bottom right, there is a button labeled "Start a post".

Microsoft Teams

Key Deliverables

Overall Goal: Develop AI-based features to enhance the platform's efficiency and user experience.

1. Functional Sidebar Chat Helper
2. Observations Improvement Feature
3. Automated data gathering

Functional SidebarChat Helper.

Floating button that expands into a sidebar chat interface.

The screenshot displays a user interface with a floating button at the bottom left. This button, which has a purple speech bubble icon, is highlighted with a red rectangular box. When expanded, it reveals a sidebar chat interface. The sidebar header includes "Azure" with a dropdown arrow, a "+" icon, and a document icon. Below the header, the text "Edit with an AI Helper" is visible. The main content area of the sidebar features a purple speech bubble icon and the placeholder text "What can I help with?". At the bottom of the sidebar is a white input field with the placeholder "Ask anything..." and a "Send >" button. To the left of the main content area, there is a small circular icon with a lightning bolt symbol. The overall interface is clean and modern, using a light color palette with purple and grey accents.

Upload Tenant specific files: [Upload File](#)

Test Observation Form:

Enter text

Azure [+](#) [document](#)

Edit with an AI Helper

What can I help with?

Ask anything... [Send >](#)

7

Responsive UI using Material UI and React components.

The screenshot displays a user interface for managing tenant-specific files and observations, featuring a sidebar and a main content area.

Upload Tenant specific files:

Test Observation Form:

Enter text

AI The Backup and Restore Vulnerability poses significant risks to organizations by allowing attackers unauthorized access to backup files. This can lead to the extraction of sensitive information, such as login credentials and financial data, which can be exploited for identity theft and fraud.

Ask anything...

Supports multiple AI models (e.g., GPT variants via Azure OpenAI)

The screenshot displays a user interface for managing tenant-specific files and interacting with AI models.

Upload Tenant specific files: A purple "Upload File" button is located here.

Test Observation Form: A text input field labeled "Enter text" is provided for testing.

Azure Model Selection: A dropdown menu titled "Azure" lists several AI models: OpenAI, Azure, Groq, and Anthropic. A red box highlights this dropdown. Below it is a text input field for "Azure API Key".

Chat Interface: On the right, there's a message icon and the text "What can I help with?". At the bottom, there's a text input field "Ask anything..." with a microphone icon, a "Send" button with a right-pointing arrow, and a small circular icon with a lightning bolt symbol.

Persistent chat history across sessions.

The screenshot displays a software interface with a left panel and a right panel. The left panel contains a file upload section with a purple 'Upload File' button, a text input field labeled 'Test Observation Form:' with placeholder 'Enter text', and a lightning bolt icon in a circular button at the bottom left. The right panel features a header with 'Azure' and a red-highlighted AI icon. Below the header is a section titled 'Edit with an AI Helper' with a 'Clear' button. A 'Recent Chats' section lists a message from 'An attacker could ex...' last updated on '15/05/2025', accompanied by a trash can icon. At the bottom right is a message input field with 'Ask anything...', a microphone icon, and a 'Send' button.

Upload Tenant specific files: **Upload File**

Test Observation Form:

Enter text

Azure

Edit with an AI Helper **Clear**

Recent Chats

An attacker could ex...
Last updated: 15/05/2025

Ask anything... **Send ➤**

File Upload and Voice Input Possibilities

The screenshot displays a user interface for managing tenant-specific files and utilizing AI helpers.

File Upload Section:

- Upload Tenant specific files:** A text input field.
- Upload File:** A purple button.

Test Observation Form:

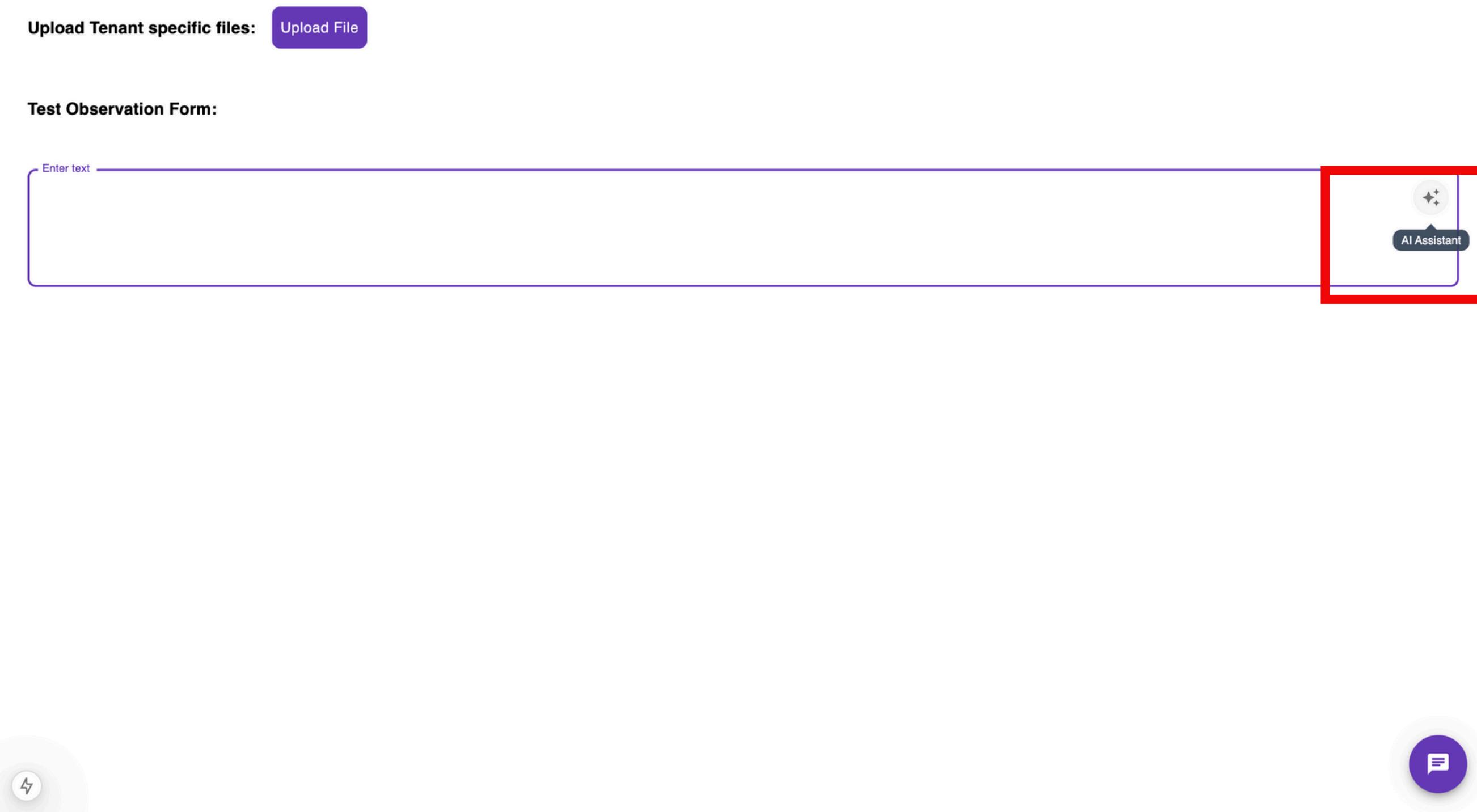
- Text Input:** A text area labeled "Enter text".

AI Helper Section:

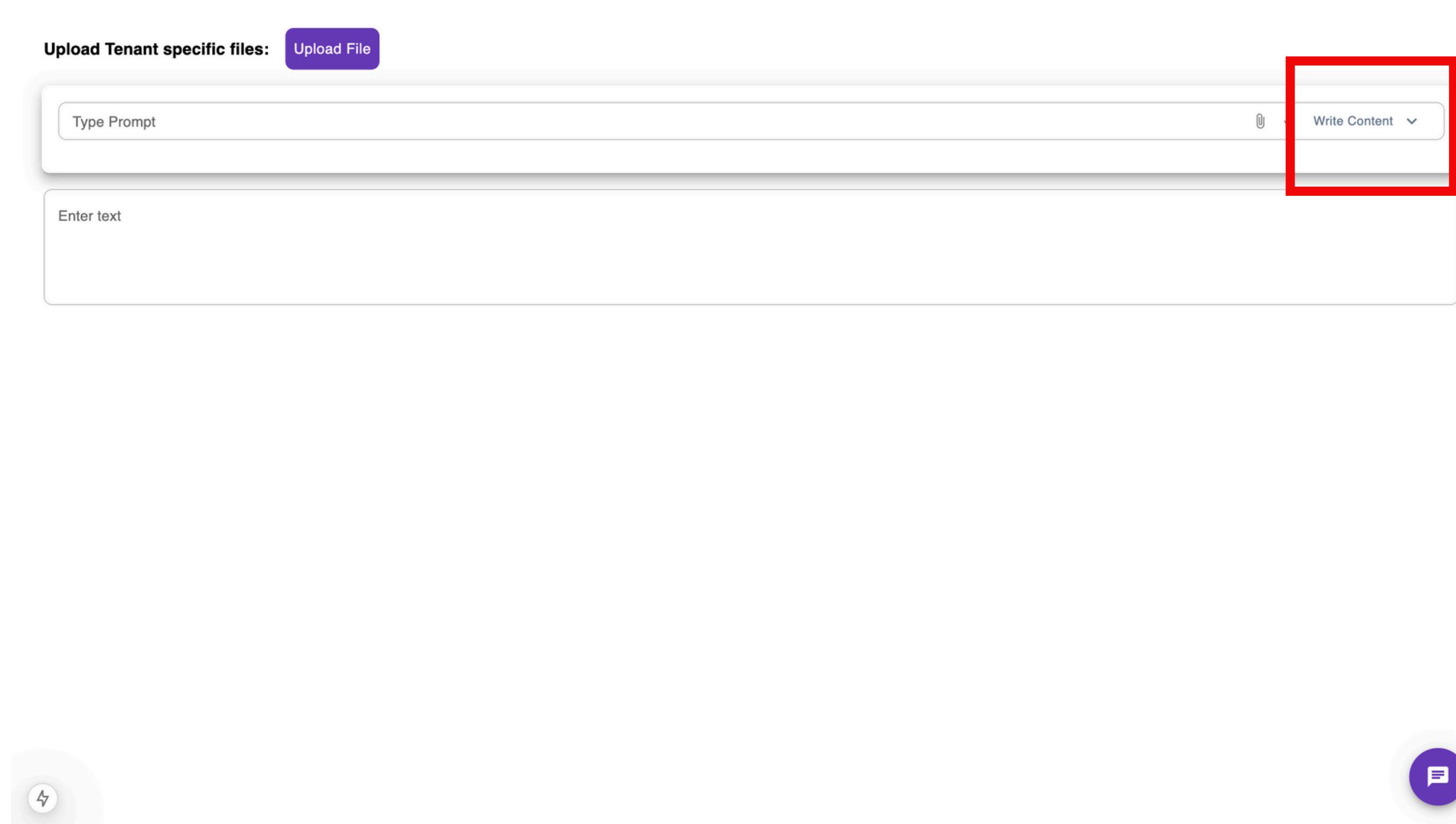
- Azure** dropdown menu.
- Edit with an AI Helper** button.
- Clear** button.
- AI Helper Content:** A purple callout box containing text about backup and restore vulnerabilities.
- AI Summary:** A large text block explaining the risks of backup and restore vulnerabilities, mentioning login credentials and financial data theft, and the potential for identity theft and fraud.
- Feedback:** Text at the bottom of the AI summary asking for more specific information.
- Voice Input:** A red-bordered microphone icon in the bottom right corner of the AI helper panel.
- Send:** A "Send" button with a right-pointing arrow.

Observations Improvement Feature

Dialog-based UI triggered by an AI button dynamically placed at the end of editable fields.



Supports multiple actions: summarize, rewrite, generate new content



AI-placeholder suggestion

Upload Tenant specific files: [Upload File](#)

Type Prompt

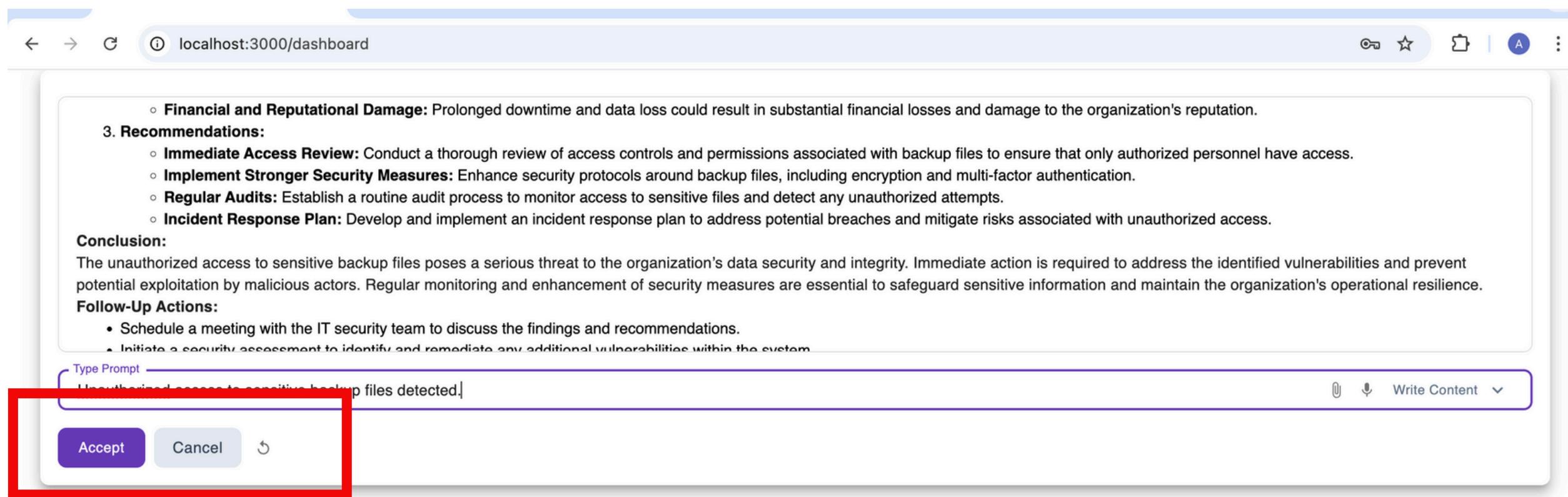
Unauthorized access to sensitive backup files detected.

Write Content

An attacker could exploit the Backup and Restore Vulnerability by gaining unauthorized access to the system's backup files. Once they have access to the backup files, they could extract sensitive information such as login credentials, personal information, or financial data. The attacker could use this information for identity theft, financial fraud, or other malicious purposes. Additionally, the attacker could modify or delete the backup files, making it difficult or impossible for the system to recover from a disaster or data loss event. This could result in significant downtime for the system and potentially cause financial or reputational damage to the organization.



Users can accept or reject AI-generated suggestions.



Security Levels

Overall Goal: Develop Multi-Tenant Security Architecture

- 1.** Public Sources
- 2.** Tenant Specific Files
- 3.** User Temporary Files

Security Levels

Overall Goal: Develop Multi-Tenant Security Architecture

1. Public Sources

2. Tenant Specific Files

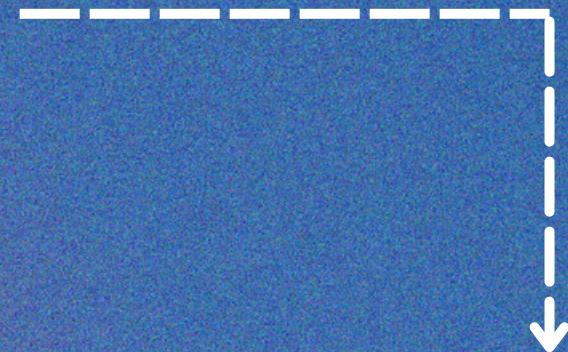
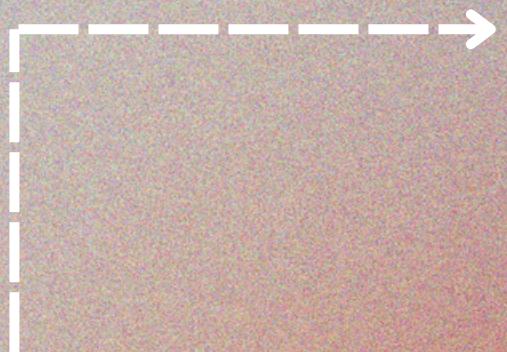
3. User Temporary Files

Data Sources Integrated

1. CVEs - Retrieved from APIs (limited to last 7 days)
2. EPSS Scores - Stored together, but could be updated separately
3. CAPEC & CWE - XML and JSON formats, stored with metadata
4. MITRE ATT&CK - Accessed via public API

Automated data gathering

Hybrid Search



The screenshot shows the drant UI interface. On the left, there's a sidebar with various icons. The main area is titled "Collections" and contains a search bar and a table. The table has columns: Name, Status, Points (Approx), Segments, Shards, Vectors Configuration (Name, Size, Distance), and Actions. There are four entries: capec, cve, cwe, and mitre_attack. Each entry shows a green status dot, approximately 559, 904, 968, and 823 points respectively, and 2 segments, 1 shard, and default vectors configuration. The "Actions" column contains three dots for each entry. At the bottom left, it says "v1.13.5". At the top right, there's a "UPLOAD SNAPSHOT" button.

The screenshot shows the pgAdmin interface. On the left, there's a tree view of database objects under "public.CVE/postgres/postgres@PostgreSQL 17". In the center, there's a "Query" tab with a query history showing two rows: "SELECT * FROM public.\"CVE\"" and "ORDER BY id ASC". Below the query, there's a "Data Output" tab displaying a table of results. The table has columns: core, cvssV2Vector, cvssV2Severity, epsScore, epsPercentile, epsDate, attackVector, attackComplexity, and privilegesRequired. The results show 21 rows of data. At the bottom, it says "Showing rows: 1 to 904". At the very bottom, it says "Servers > PostgreSQL 17 > Databases > postgres > Schemas > public > Tables > CVE".

Vector Database

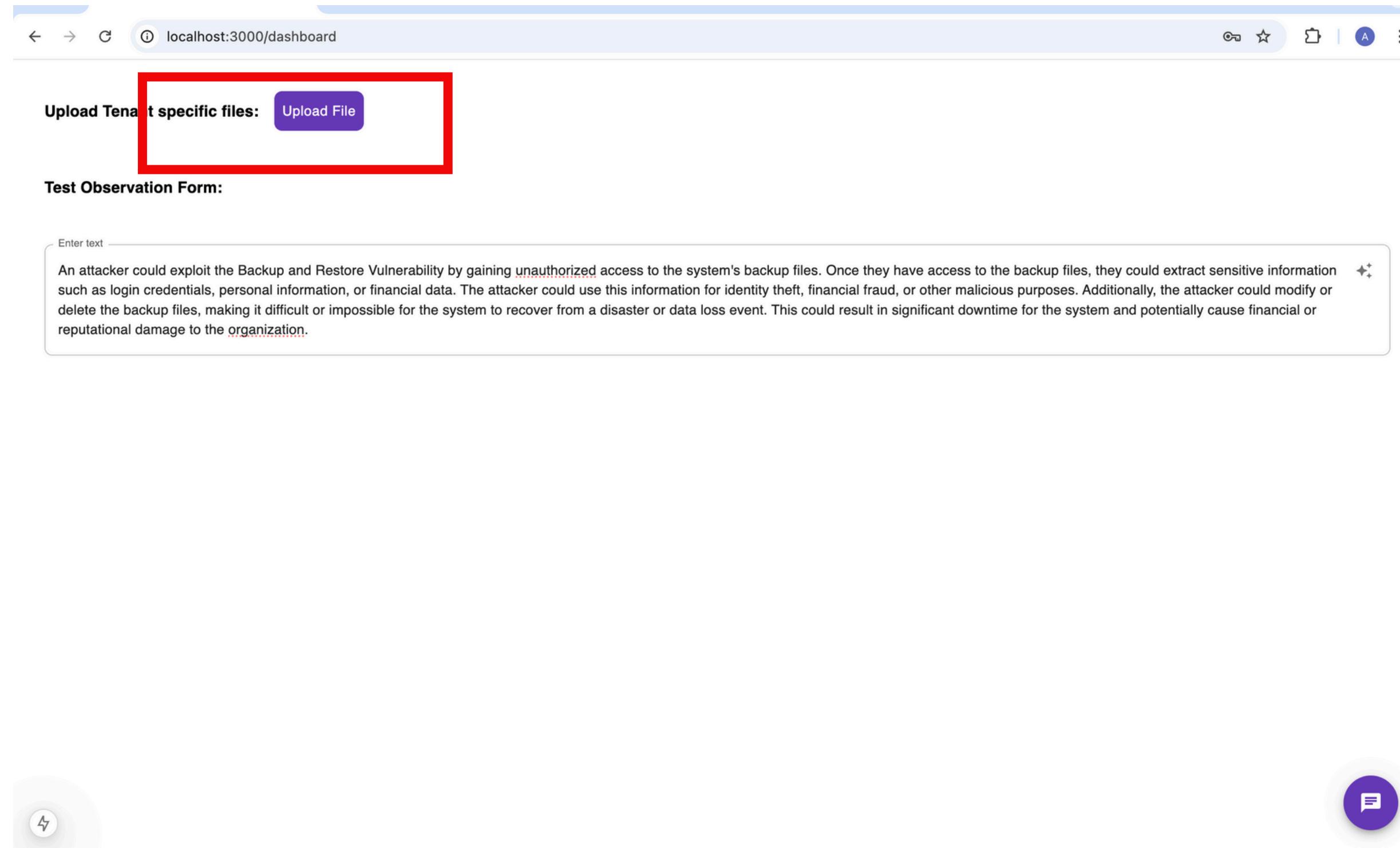
PostgreSQL

Security Levels

Overall Goal: Develop Multi-Tenant Security Architecture

1. Public Sources
2. Tenant Specific Files
3. User Temporary Files

Client-specific data and configurations



The screenshot shows a web browser window with the URL `localhost:3000/dashboard` in the address bar. The page content includes a heading "Upload Tenant specific files:" followed by a red-bordered input field and a purple "Upload File" button. Below this, there is a section titled "Test Observation Form:" containing a text area with placeholder text: "Enter text". The text area contains a paragraph about backup and restore vulnerabilities, mentioning unauthorized access, sensitive information extraction, identity theft, financial fraud, and downtime. A small "Edit" icon is located at the top right of the text area.

Security Levels

Overall Goal: Develop Multi-Tenant Security Architecture

1. Public Sources
2. Tenant Specific Files
3. User Temporary Files

Temporary session-based files (cleared after use)

The screenshot shows a web browser window with the URL `localhost:3000/dashboard`. At the top, there is a header bar with standard browser controls. Below the header, a purple button labeled "Upload Tenant specific files:" has an "Upload File" button next to it. A red rectangular box highlights a white input field labeled "Type Prompt". To the right of this input field is a "Write Content" dropdown menu with icons for text, microphone, and file. Below the input field is a text area with placeholder text "Enter text". Inside this text area, there is a detailed description of a backup and restore vulnerability:

An attacker could exploit the Backup and Restore Vulnerability by gaining unauthorized access to the system's backup files. Once they have access to the backup files, they could extract sensitive information such as login credentials, personal information, or financial data. The attacker could use this information for identity theft, financial fraud, or other malicious purposes. Additionally, the attacker could modify or delete the backup files, making it difficult or impossible for the system to recover from a disaster or data loss event. This could result in significant downtime for the system and potentially cause financial or reputational damage to the organization.

At the bottom of the page, there are two circular icons: one with a lightning bolt symbol and another with a speech bubble symbol.

Challenges

- 1. No Standardized Update Schedule for Public Datasets**
- 2. Unstable Public API Endpoints**
- 3. Qdrant Delete Request Limitation**

Requirements for Integration



Self-Hosting Large Language Model
(LLM)



Hosting an Embedding Model

Project Planning

March - Mid-April

Finalize advanced conversation logic in the sidebar chat.

Integrate RAG for enhanced contextual responses.

Mid-April - Early May

Implement Multi-Tenant Security Layers

Fetch and Embed Public Datasets

Early May - Internship End

Implement automated retraining and data fetching.

Conduct performance evaluation and create documentation.

Technology Stack

LAYER	TECHNOLOGY	PURPOSE
Frontend	React, Next.js, Material UI	UI components, routing, responsive design
Backend	Node.js, TypeScript, Prisma ORM, PostgreSQL	Server logic, API routes, data persistence
AI & RAG	LangChain, Azure OpenAI, ONNX Runtime, Qdrant	Prompt chaining, LLM interaction, embedding generation, vector search
Data Handling	JSON/XML/OWL parsers, multi-retriever chains	Normalizing and retrieving public datasets
Security	Multi-level access control, secure API layer	Preventing data leakage, tenant isolation
Dev Tools	Git, GitHub, SonarCloud, VS Code	Code management, quality checks, development

Thank you