THOMAS MORE

Enhancing Refracted Security Platform with AI Capabilities

Reflection document

Alfiya Abdimutalipova
Student Bachelor Applied Computer Science

2024 - 2025

Table of Contents

1. Introduction

This reflection document provides an overview of my personal and professional growth during the internship at Refracted Security, where I worked on integrating AI-powered enhancements into their vulnerability management platform. The goal was to support penetration testers by offering intelligent tools that assist in writing, refining, and contextualizing security findings.

The internship challenged me to step outside my comfort zone — both technically and professionally. It involved full-stack development, integration of AI models, and working within a professional software development environment. This document reflects on what I learned, how I grew, the challenges I faced, and how I overcame them.

2. What did I learn?

Throughout the internship, I developed a broad range of technical and professional competencies that significantly enhanced my understanding of software development and artificial intelligence integration within a cybersecurity context.

2.1. Technical Skills

On the technical side, I gained substantial exposure to modern web development technologies such as Next.js, React, and TypeScript, all of which were relatively new to me at the start of the internship. I applied these skills to build interactive user interface components like the Sidebar Chat Helper and Observations Improvement Feature, allowing me to gain confidence in both frontend design and backend logic implementation. Through this process, I became familiar with best practices in UI/UX development and learned how to create scalable, maintainable code structures within a large application.

A significant portion of my work revolved around integrating artificial intelligence capabilities into the platform. I explored various embedding models, ultimately selecting model from Hugging Face for its strong performance in semantic similarity tasks. I also implemented a Retrieval-Augmented Generation (RAG) system using Qdrant as the vector database, enabling the AI to pull from up-to-date cybersecurity knowledge when generating responses. My exploration further extended into tools like LangChain and Ollama, where I experimented with hybrid search strategies to improve the accuracy and relevance of AI suggestions.

An important part of the project involved working with public cybersecurity datasets such as CVEs, CWEs, CAPECs, and MITRE ATT&CK. I built data processing pipelines to fetch, clean, and convert this information into vector representations suitable for semantic search. This required implementing batch processing techniques to handle large volumes of data efficiently. I also introduced validation steps to ensure data consistency and system reliability over time.

2.2. Professional Skills

Beyond technical development, I also strengthened my professional skills through active participation in a real-world development environment. I contributed to weekly planning cycles, learning how to break down complex tasks, estimate effort, and deliver incremental progress. I paid close attention to code quality, ensuring good documentation practices, modular architecture, and separation of concerns in my implementations. These habits not only improved readability but also made future maintenance and collaboration easier.

Communication and teamwork played a crucial role in the success of the project. Regular interactions with my mentor allowed me to clarify requirements, receive feedback, and refine my approach iteratively. I learned to ask targeted questions, articulate technical challenges clearly, and document my progress effectively — all essential skills in a collaborative software engineering setting.

Overall, the internship offered a rich learning experience that spanned both technical execution and professional growth, equipping me with valuable tools and insights applicable to future roles and projects.

2.3. Competencies Addressed

During my internship at Refracted Security, I had the opportunity to apply and further develop several core competencies outlined in the Bachelor Applied Computer Science program. These competencies spanned across technical development, software engineering practices, problem-solving, and professional collaboration. The tasks involved not only building new features but also understanding how they fit into a larger system, ensuring performance, maintainability, and alignment with user needs.

Analytical Thinking	Analyzed technical requirements, compared AI models, and made informed architectural decisions.
Design and Implementation	Designed user interfaces in Figma and implemented them in React, ensuring functionality and usability.
Research and Innovation	Researched embedding models, vector databases, and hybrid search methods to find optimal solutions.
Problem Solving	Debugged API requests, fixed memory synchronization issues, and optimized ingestion pipelines.
Professionalism and Ethics	Worked under IP-sensitive conditions, respected confidentiality policies, and ensured data privacy.
Communication and Collaboration	Participated in discussions, presented progress updates, and clearly documented development processes.

By engaging deeply with these competencies, I not only contributed to a real-world cybersecurity product but also significantly strengthened my readiness for future roles in software development and AI integration.

3. Challenges Faced and How I Solved Them

Throughout the internship, I encountered several technical and non-technical challenges that tested my problem-solving abilities, adaptability, and resilience. These challenges ranged from infrastructure limitations to learning new tools under time constraints. Each was addressed through thoughtful planning, experimentation, and collaboration with my mentor.

3.1. Technical Challenges

1. Embedding Large Datasets Efficiently

One of the first major technical challenges involved handling large volumes of cybersecurity data — particularly Common Vulnerabilities and Exposures (CVEs) — which were computationally expensive to embed into vector representations. Processing the full historical dataset caused performance bottlenecks, especially during ingestion and querying stages.

To address this, I implemented a filtering strategy that limited the dataset to only the most recent entries — specifically those from the past seven days. This ensured the knowledge base remained current while significantly improving system performance. Additionally, I optimized the embedding pipeline by implementing batch insertion logic, reducing API overhead and memory usage.

2. Hybrid Search Implementation

A significant architectural challenge was integrating structured numerical data — such as EPSS scores — with semantic search results obtained from the vector database. Ensuring accurate and relevant output required careful design of the query execution flow.

To solve this, I developed a custom query handler that combined both search paradigms. Queries first attempt exact matches on structured fields using PostgreSQL before falling back to semantic similarity search if no direct match is found. This hybrid approach significantly improved answer accuracy and enabled more nuanced filtering based on both unstructured text and numerical data.

3. Incompatible Data Formats

Working with MITRE DEFEND data introduced additional complexity due to its use of the OWL (Web Ontology Language) format, which is not directly compatible with standard NLP pipelines or vector databases.

As a solution, I explored alternative approaches such as Neo4j and Langchain's GraphCypherQAChain to extract and query knowledge from OWL files. However, due to time constraints and implementation complexity, full integration was deferred to a future phase. Instead, focus was shifted to more accessible datasets, which provided usable APIs and JSON/XML formats.

3.2. Personal / Non-Technical Challenges

4. Learning New Tools Quickly

At the beginning of the internship, I had limited experience with modern web development frameworks like React and TypeScript. Understanding how these technologies fit together within a professional software stack posed an initial challenge.

To overcome this, I dedicated time to self-directed learning, including reviewing documentation, following tutorials, and building small test projects. This foundational work allowed me to confidently implement larger components and contribute effectively to the platform.

5. Health Issues Affecting Productivity

There were periods during the internship where illness temporarily reduced my ability to maintain consistent productivity levels.

In response, I maintained open communication with my mentor, adjusted task deadlines where necessary, and focused on completing smaller, manageable tasks during recovery periods. This approach helped maintain progress without compromising quality or long-term project goals.

4. Conclusion

This internship was a transformative experience that significantly expanded both my technical and professional skill set. I successfully contributed to a real-world cybersecurity product by implementing AI-driven tools that enhance efficiency and quality for penetration testers.

I learned to work independently while staying aligned with team goals, to solve complex problems through research and experimentation, and to respect the constraints of professional environments — especially around data privacy and intellectual property.

The experience has deepened my interest in AI integration, semantic search, and cybersecurity tooling, and I am eager to continue exploring these areas in future roles or studies.