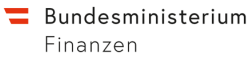
MINERVA Ergebnisbericht

Finanziert im Sicherheitsforschungs-Förderprogramm KIRAS des Bundesministeriums für Finanzen



Durchgeführt von

contact@cyberaci.com

https://www.cyberaci.com/minerva/

office@ikarus.at

https://www.IKARUSsecurity.com

31. Jänner 2025

Executive Summary

Das MINERVA-Projekt zielte darauf ab, die IKARUS Threat Intelligence Platform (TIP) mit einer KI-gestützten Lösung zu erweitern. Dazu wurde ein Open-Source-basiertes Large Language Model (LLM) integriert, das Analyst:innen dabei unterstützt, strukturierte und unstrukturierte Daten effizient zu verarbeiten. Ein zentraler Fokus lag auf der natürlichen Sprachinteraktion, um komplexe Bedrohungsanalysen zu erleichtern. Aufgrund begrenzter GPU-Ressourcen und der hohen Komplexität der TIP-Datenbank wurde eine neue Bedrohungsdatenbank entwickelt, die eine präzisere und leistungsfähigere Verarbeitung ermöglicht.

Kernfunktionen von Minerva:

- Natürliche Sprachsuche: Automatische Generierung von SQL-Abfragen für Datenbankanfragen.
- Dokumenten-Chat: KI-gestützte Extraktion relevanter Informationen aus unstrukturierten Daten.
- Multi-Hop-Fragebeantwortung: Verknüpfung von SQL-Abfragen und Dokumentenanalyse für komplexe Anfragen.

Trotz Herausforderungen wie Hardware-Beschränkungen und Datenkomplexität konnten alle Kernziele erreicht werden. Minerva wurde erfolgreich entwickelt und ist sowohl in der Cloud als auch On-Premise einsetzbar. Damit bietet die Lösung eine flexible und leistungsfähige Unterstützung für die Bedrohungsanalyse und Cybersecurity-Entscheidungen. Details zu MINERVA unter: https://www.cyberaci.com/minerva/

Inhaltsverzeichnis

1. Einleitung

Das zentrale Ziel des Projekts war es, die bestehende Threat Intelligence Platform (TIP) von IKARUS durch den Einsatz einer künstlichen Intelligenz (KI) zu erweitern. Dabei sollte ein Open-Source-basiertes Large Language Model (LLM) verwendet werden, das vollständig on-premise bei IKARUS betrieben wird. Dieses LLM sollte sowohl strukturierte als auch unstrukturierte Daten verarbeiten können, um relevante Informationen effizient abzurufen und komplexe Fragen von Analyst:innen in natürlicher Sprache zu beantworten.

Zu Beginn des Projekts wurde die Entwicklung des LLMs auf der bestehenden Infrastruktur von IKARUS gestartet. Die KI wurde darauf trainiert, Berichte und Bedrohungsdaten aus der TIP zu verarbeiten. Allerdings stieß das System aufgrund von hardwareseitigen Einschränkungen – insbesondere der limitierten GPU-Kapazitäten (2x NVIDIA L40) – an seine Grenzen. Die vorhandenen GPUs waren voll ausgelastet, was eine effiziente Verarbeitung der Daten erschwerte. Um dennoch Fortschritte zu erzielen, wurde entschieden, ein externes LLM eines Service Providers einzusetzen, um die geplanten Funktionen zu validieren und Minerva weiterzuentwickeln.

Ein weiterer zentraler Aspekt des Projekts war die Integration der KI in die strukturierte Datenbasis von IKARUS, um mithilfe natürlicher Sprache komplexe Datenbankabfragen zu ermöglichen. Hierbei wurde das LLM mit einer Vielzahl von Beispielen trainiert, um die abstrakte und hochkomplexe Struktur der TIP-Datenbank zu verstehen. Trotz enormer Bemühungen stellte sich heraus, dass die TIP-Datenbank aufgrund ihrer Komplexität nicht ideal für die direkte Integration war. Um dieses Hindernis zu überwinden, wurde eine neue, interne Bedrohungsinformations-Datenbank aufgebaut. Diese Datenbank war einfacher strukturiert und erlaubte eine verlässliche Verifizierung der von der KI generierten Antworten. Diese Datenbank

wurde mit Bedrohungsinformationen aus Quellen wie Tenable, Qualys, Microsoft Defender und NVD befüllt. Das LLM wurde erfolgreich mit dieser internen Datenbank trainiert und zeigte dabei eine deutliche Verbesserung in der Genauigkeit und Zuverlässigkeit der Antworten. Grundsätzlich konnten alle primären Projektziele erreicht werden. Die KI ist in der Lage, sowohl strukturierte als auch unstrukturierte Daten effizient zu verarbeiten und Analyst:innen bei der Beantwortung komplexer Fragen zu unterstützen.

Die entwickelte Endversion von Minerva wurde aufgrund hardwareseitiger Einschränkungen nicht on-premise bei IKARUS und mit ihrer TIP umgesetzt. Stattdessen erfolgte die Entwicklung auf einer Cloud-Plattform, bei der die GPU-Ressourcen der Cloud genutzt wurden, um die notwendige Rechenleistung sicherzustellen. Zudem wurde eine neue, speziell für das Projekt entwickelte Bedrohungsdatenbank aufgebaut, die nicht auf der IKARUS-TIP basiert.

Durch diese Flexibilität kann Minerva sowohl in einer Cloud-Umgebung als auch, bei Vorhandensein ausreichender GPU-Ressourcen, in einer air-gapped On-Premise-Umgebung betrieben werden. Dies bietet Unternehmen die Möglichkeit, Minerva entsprechend ihren spezifischen Sicherheits- und Infrastrukturbedürfnissen zu nutzen.

Die Abbildung unten zeigt, wie Analyst:innen Informationen in natürlicher Sprache aus unstrukturierten Daten (z. B. einer Vielzahl von Dokumenten) sowie aus strukturierten Daten (mithilfe von Text-to-SQL) effizient aus Minerva abrufen können.

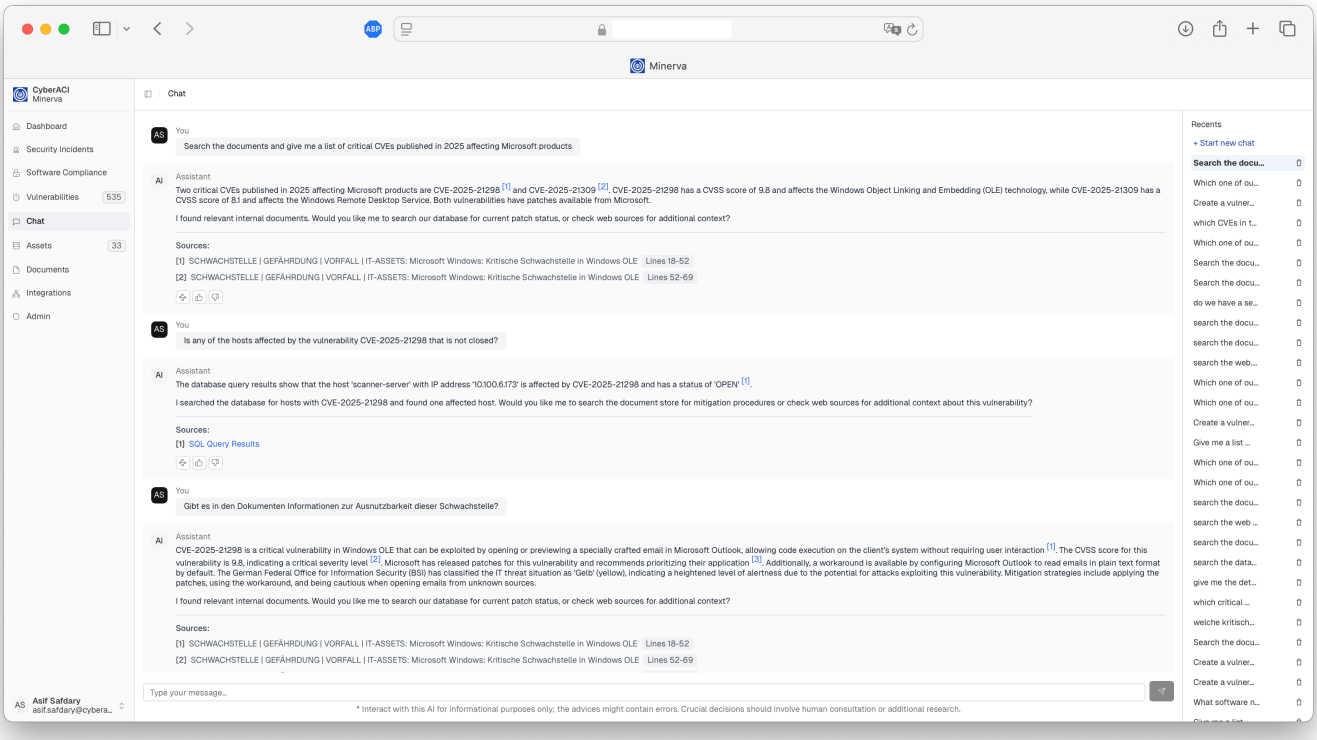


Abbildung 1: Minerva Darstellung einer natürlichen Sprachinteraktion mit strukturierten und unstrukturierten Daten.

Während der Gespräche kristallisierten sich einige Fragen heraus, die Analyst:innen häufig zur Unterstützung ihrer Analysetätigkeiten benötigen. Es wurde im Team diskutiert, wie diese Daten ohne KI-Funktionalitäten direkt auf der Benutzeroberfläche angezeigt werden können, um eine schnelle und einfache Entscheidungsfindung zu ermöglichen.

Die folgende Abbildung zeigt eine Übersicht über die Bedrohungslage eines Unternehmens und die historischen Daten.

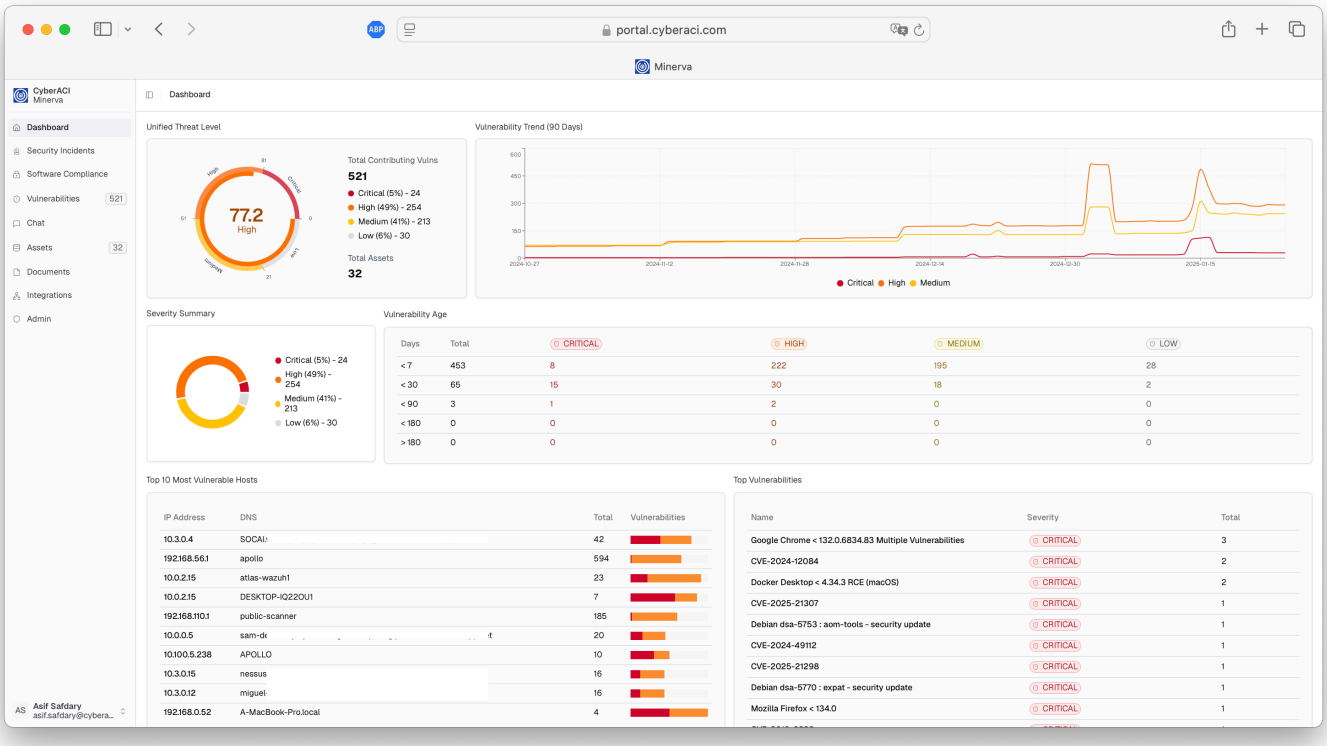


Abbildung 2 : Übersicht über die Bedrohungslage und die historischen Daten

Zusätzlich wurde das Software Compliance Modul entwickelt, um den gesamten Softwarebestand zu inventarisieren. Dieses Modul erfasst, welche Versionen installiert sind und ob diese Software aufgrund bekannter Schwachstellen aktualisiert werden muss, sowie auf welche Versionen diese aktualisiert werden sollten.

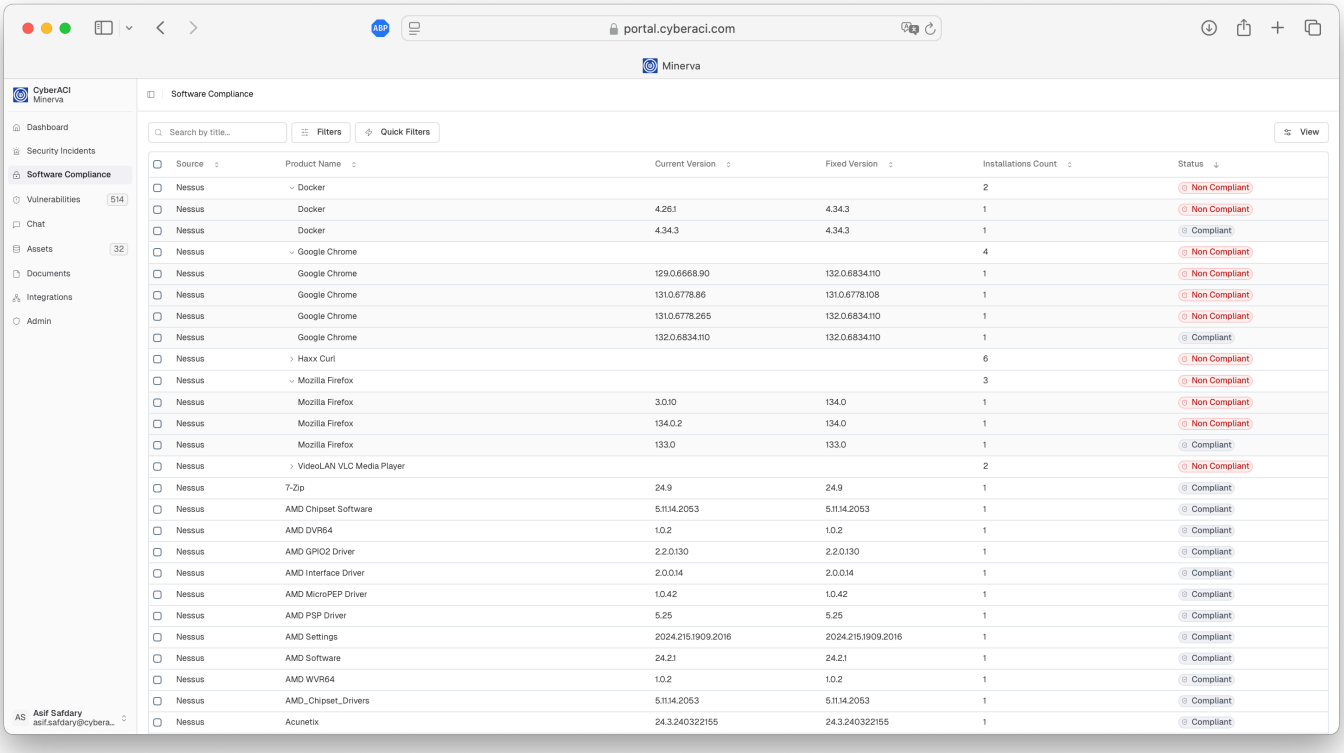


Abbildung 3: Software Compliance Modul

Lessons Learned

Eine KI mit Datenbanken zu trainieren, die komplexe Geschäftslogiken beinhalten, benötigt extrem viel Zeit und eine enge Abstimmung mit Personen, die sich mit dem Datenbankschema bestens auskennen – vor allem, wenn aus dem Schema nicht eindeutig hervorgeht, welche Tabellen ausschließlich Bedrohungsinformationen beinhalten.

Das Aufsetzen von RAG (Retrieval-Augmented Generation) und die Verarbeitung von unstrukturierten Daten mit geeigneten LLMs (in unserem Fall LLaMA 70B) ist relativ schnell möglich, erfordert jedoch leistungsstarke GPUs, um effizient arbeiten zu können.

Eine ausreichende Infrastruktur mit passenden Hardware-Ressourcen ist entscheidend. KI-Entwickler:innen sollten sich auf die Entwicklung konzentrieren können, ohne zwischen

Performance-Optimierungen, Entwicklungsarbeit und Hardware-Beschränkungen jonglieren zu müssen.

Eine enge Abstimmung zwischen KI-Entwickler:innen und Cybersecurity Analyst:innen ist essenziell, um der KI die relevanten Business Cases beizubringen, die Analyst:innen in ihrer täglichen Arbeit benötigen.

Ein derart komplexer Use Case benötigt mehr Zeit und Entwicklungsaufwand, um robust und den Anforderungen entsprechend implementiert werden zu können.

2. Durchgeführte Arbeiten und ihre Ergebnisse

2.1. Zugang, Entwicklungsumgebung und Infrastruktur

Sämtliche vereinbarte Arbeitspakete wurden termingerecht umgesetzt. Diese Arbeitspakete umfassen folgende Punkte

- Zugang und Entwicklungsumgebung einrichten
- Zugang und Berechtigungen zum IKARUS-Netzwerk erhalten
- Definition der Hardwareanforderungen
- Entwicklungsumgebung einrichten
- Entwurf des Produktionseinsatzes on-premise
- Bestellung und Lieferzeit
- Assemblierung und Einbau der Hardware
- Entwicklungsumgebung ist eingerichtet und der Fernzugriff funktioniert

Beschreibung der Architektur für ein spezialisiertes Offline LLM

Die Architektur besteht im Wesentlichen aus drei Komponenten

1. Bedrohungsinformationsplattform inkl. Multi Source Bedrohungsinformationen

1.1. IKARUS betreibt und entwickelt bereits seit 2022 eine Bedrohungsinformationsplattform auf der Basis von Threat Connect, einem amerikanischen Softwarehersteller im Bereich der Bedrohungsinformationsplattformen.

1.2. Innerhalb dieser Plattform erfolgt die Datenaggregation der unterschiedlichsten Bedrohungsinformationsquellen, um diese einem Analysten nutzbar zu machen.

2. LLM-System:

2.1. Natural Language Processing (NLP): Unterstützung für natürliche Fragestellungen, um technische Hürden zu reduzieren.

2.2. Dynamische Antworten: Aufbereitung relevanter Berichte mit Quellennachweis.

3. Technologische Infrastruktur:

3.1. Modellbasis: Verwendung von etablierten LLM-Frameworks.

3.2. Hardware: Hochleistungsserver (On-Prem) mit NVIDIA GPUs für rechenintensive Aufgaben wie Training und Inferenz.

3.3. Integration: Verbindung mit dem internen SIEM (Security Information and Event management) auf Basis von Elastic sowie dem internen EDR (Endpoint Detection and Response) auf Basis von HarfangLab Guard feat. IKARUS

2.2. Chat mit Dokumenten

Die Funktionalität, Informationen aus Dokumenten im Chat zu extrahieren, war eines der zentralen Ziele des Projekts. Ziel war es, relevante Textabschnitte aus Dokumenten basierend auf Benutzeranfragen mittels Retrieval Augmented Generation (RAG) zu extrahieren, diese durch ein LLM zusammenfassen zu lassen und präzise Antworten zu generieren. Dadurch werden aus unstrukturierten Daten nutzbare Erkenntnisse für Cybersecurity Expert:innen erstellt. Diese Funktionalität stellt einen essenziellen Bestandteil von Minerva dar.

Durchgeführte Arbeiten:

Embedding-Modell-Auswahl: Das Team entschied sich für das Jina-Embedding, das durch überzeugende Benchmark-Ergebnisse hervorstach. Die Implementierung der "late-chunking[1]" Technik, eingeführt im Rahmen der RAG-Forschung von Jina, optimierte die Kontextgenauigkeit bei der Suche nach relevanten Informationen in den Dokumenten. Dieses Gebiet bleibt dynamisch, da global kontinuierlich alternative Embedding-Modelle evaluiert werden.

Chat-LLM-Auswahl: Für allgemeine Chat-Funktionalitäten kam zunächst Llama 3.1 70B INT4[2] zum Einsatz. Aufgrund eingeschränkter Fähigkeiten in der Verarbeitung von Instruktionen und nicht zufriedenstellender Ergebnisse bei der Textzusammenfassung wechselte das Team jedoch zu einem Cloud-Provider, der Llama 3.3 70B[3] unquantisiert bereitstellt. Dieser Schritt war notwendig, da die on-premise gehosteten GPUs nicht über ausreichende Rechenleistung für die unquantisierte Variante verfügten. Die Llama 3.3 70B Variante ermöglicht eine komplexere Verarbeitung und verbesserte die Such- und Antwortgenauigkeit von Minerva erheblich.

Dokumentenverarbeitung: Die Verarbeitung der umfangreichen IKARUS TIP Dokumentensammlung stieß aufgrund hardwareseitiger Einschränkungen an Grenzen. Insbesondere die zwei verfügbaren NVIDIA L40 GPUs begrenzten die Verarbeitung von Dokumenten und somit die Entwicklungsarbeiten erheblich, was zu Verzögerungen führte.

---

[1] https://jina.ai/news/late-chunking-in-long-context-embedding-models/
[2] https://huggingface.co/hugging-quants/Meta-Llama-3.1-70B-Instruct-AWQ-INT4
[3] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

Hyperparameter-Optimierung: Hyperparameter sind Anpassungsparameter[4], die das Verhalten des LLMs beeinflussen, indem sie unterschiedliche Aspekte der Daten unterschiedlich gewichten. In diesem Projekt wurden drei Hauptdimensionen berücksichtigt: Semantik, die den Sinn und Zusammenhang eines Satzes erfasst; Keywords, die spezifische Begriffe im Text erkennen; und Metadaten, die Informationen wie Autor:innen, Titel und Veröffentlichungsdatum enthalten[5]. Diese Parameter wurden manuell kalibriert, um ein ausgewogenes Verhältnis zwischen diesen Dimensionen zu erreichen. Das Ziel war, eine effiziente und präzise Strategie zur Informationssuche und -extraktion zu entwickeln, bei der keine einzelne Dimension übermäßig priorisiert wird, sondern alle gemeinsam die bestmögliche Suche ermöglichen und die Retrieval Qualität zu optimieren. Eine automatisierte Optimierung wurde nicht durchgeführt, da das Gebiet der Hyperparameter-Suche weiterhin ein aktives Forschungsfeld ist. Stattdessen wurde die Kalibrierung der Parameter iterativ und manuell vorgenommen, da keine etablierte Methode existiert, die alle Anforderungen gleichermaßen erfüllt.

Pipeline-Entwicklung: Die Wissensextraktions- und Fragebeantwortungspipelines wurden erfolgreich entwickelt. Eine dedizierte Route für Common Vulnerabilities (CVE) bezogene Anfragen wurde eingeführt, um die Präzision der Ergebnissuche zu verbessern. Diese Route umfasst spezifische logische Schritte, die gezielt auf die Verarbeitung solcher Anfragen abgestimmt sind.

Die Verarbeitung der großen Menge an Dokumenten belastete die Hardware-Ressourcen erheblich. Eine grobe Schätzung ergab, dass mit der bestehenden Infrastruktur etwa drei Monate

[4] https://arxiv.org/abs/2406.19251
[5] https://dl.acm.org/doi/abs/10.1145/3637528.3671470

dauern würde, um alle PDFs der IKARUS TIP-Datenbank zu verarbeiten. Zudem wurde die Hyperparameter-Optimierung iterativ durchgeführt, was zusätzliche Entwicklungszeit erforderte.

Abweichungen, Anpassungen und Auswirkungen

Der Grundansatz des RAG Frameworks blieb unverändert, wurde jedoch durch spezifische Anpassungen optimiert. Dazu zählten spezialisierte Routen für CVE-Abfragen, Verbesserungen der Embedding-Modelle und verfeinerte Retrieval-Prozesse. Diese Anpassungen erhöhten die Präzision und Effizienz des Systems erheblich. Trotz der Herausforderungen konnte die Gesamtdauer des Projekts eingehalten werden, wodurch ein robustes und voll einsatzbereites System entstand.

Dieses Arbeitspaket wurde vollständig abgeschlossen. Alle geplanten Funktionalitäten sind erfolgreich implementiert und betriebsbereit. Diese Fortschritte bilden eine solide Grundlage für die sichere und effiziente Verarbeitung von unstrukturierten Daten in Cybersecurity-Anwendungen. Minerva kann Dokumente sowohl aus einem internen Verzeichnis abrufen und verarbeiten als auch manuell hochgeladene Dateien integrieren.

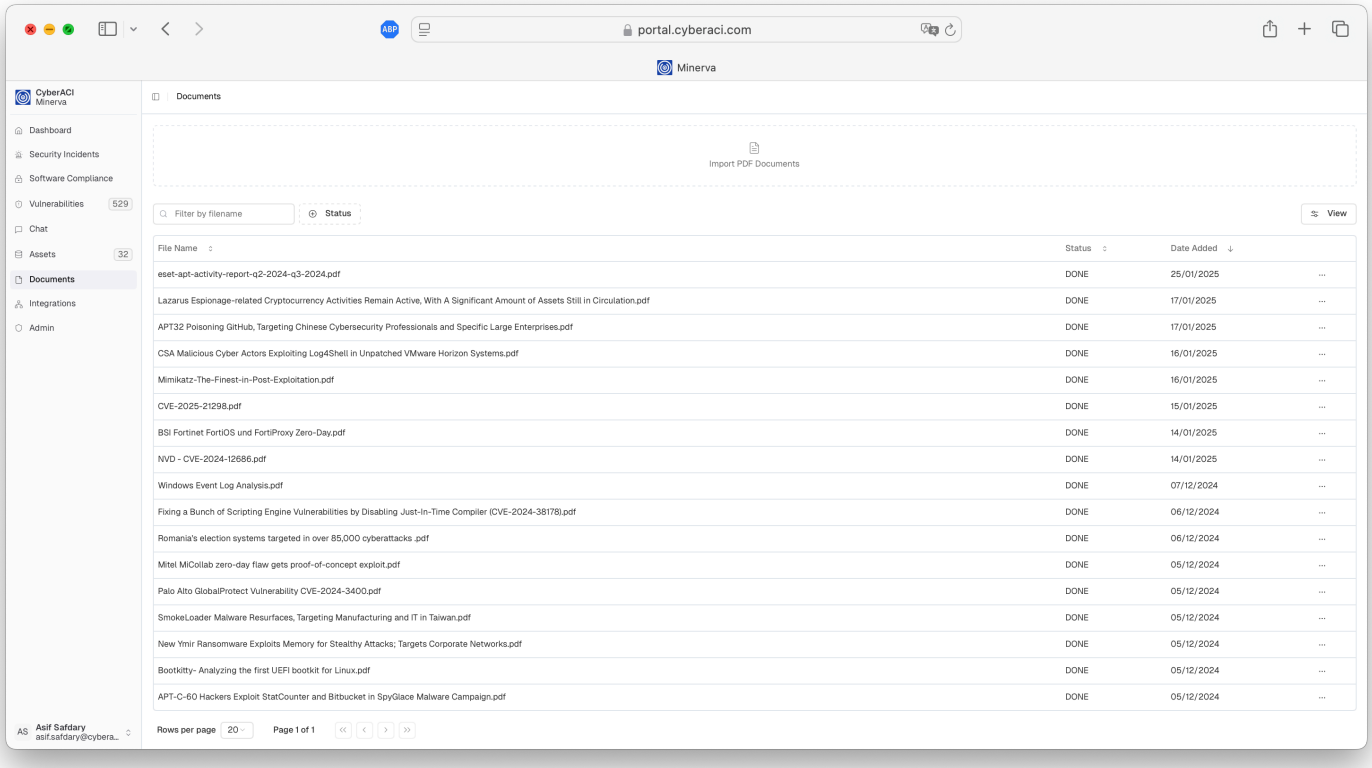


Abbildung 4: Dokumente können manuell oder automatisiert zum Verarbeiten hochgeladen werden.

Folgende Abbildung veranschaulicht, wie Cybersecurity Analyst:innen Informationen aus einer Vielzahl von Dokumenten in Minerva abfragen und extrahieren können, die für sie relevant sind.

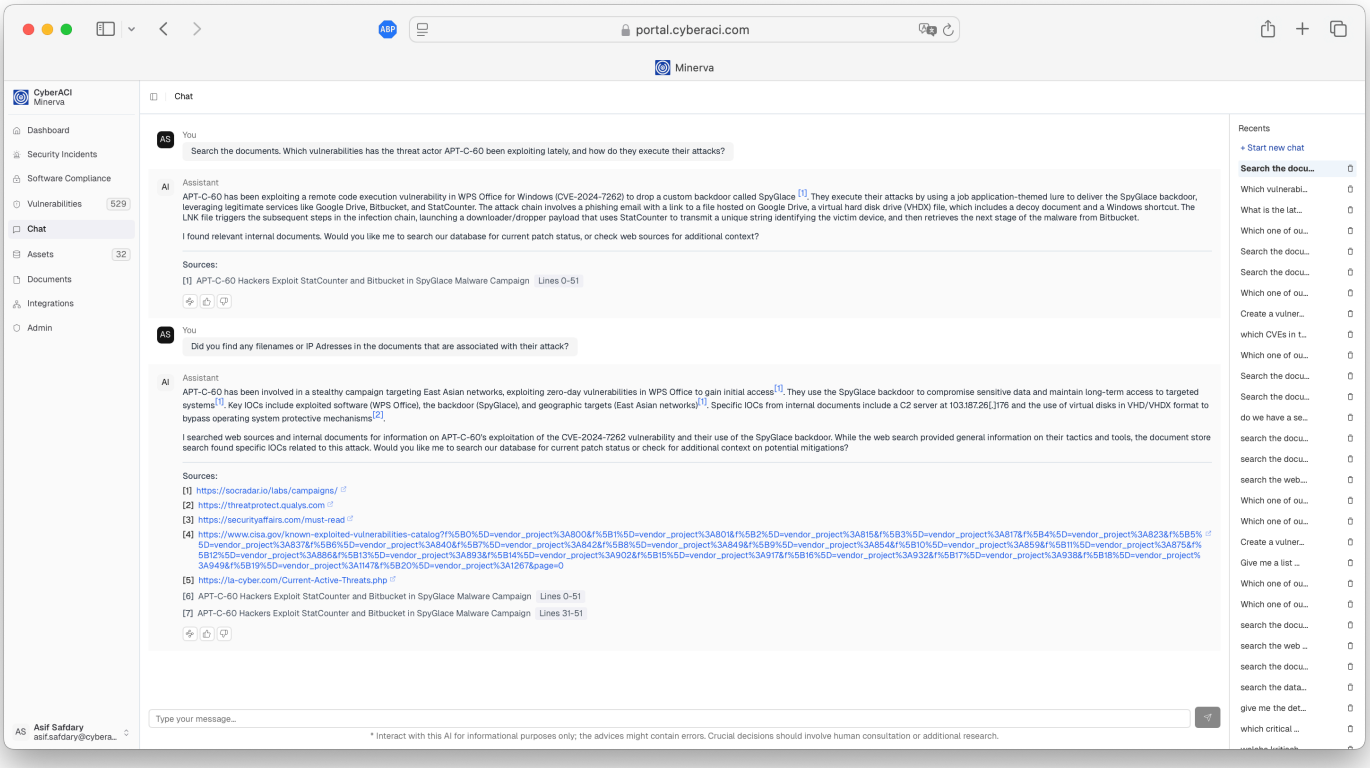


Abbildung 5: Interaktives Chatten mit Dokumenten

2.3. Durchsuchen der Datenbank mit natürlicher Sprache

Die Funktionalität, eine Datenbank mit natürlicher Sprache zu durchsuchen, basiert darauf, Benutzeranfragen in SQL-Befehle zu übersetzen. Eine der zentralen Herausforderungen bestand darin, das Datenbankschema so effektiv zu nutzen, dass das LLM die Struktur und die Beziehungen innerhalb der Datenbank verstehen konnte. Dies war essenziell, um präzise und sinnvolle SQL-Abfragen zu generieren, die Analyst:innen einen schnellen Zugriff auf relevante Informationen ermöglichen.

Durchgeführte Arbeiten

Workshops und Log-Analyse: Das Team organisierte Workshops[6] mit Sicherheitsexperten von IKARUS, um typische Fragestellungen aus ihrem Arbeitsalltag zu identifizieren, die oft mit SQL-Abfragen beantwortet werden. Parallel dazu wurden die Datenbank-Logs der TIP analysiert, um die durch Benutzeroberflächenaktionen generierten SQL-Abfragen zu verstehen. Diese Abfragen waren oft hochkomplex und spiegelten die abstrakte Struktur der TIP-Datenbank wider.

SQL-Generator-LLM-Auswahl: Aufgrund von Hardware-Beschränkungen wurde das universelle LLM LLaMA 70B für alle Anwendungsfälle eingesetzt. Die gleichzeitige Nutzung eines SQL-spezifischen und eines allgemeinen LLMs war aufgrund der GPU-Limitierungen in der Infrastruktur nicht möglich. Während des Projekts wurden Varianten des LLaMA 70B INT4 verwendet, bevor das Team später zu einer Cloud-Lösung mit dem vollständigen LLaMA 70B-Modell wechselte, da die GPU-Beschränkungen der IKARUS-Infrastruktur dies erforderten.

Verarbeitung des Datenbankschemas: Die hochgradig abstrakte Struktur des TIP-Datenbankschemas stellte die größte Herausforderung dar. Trotz mehrfacher Ansätze erwies es sich als unmöglich, mit LLaMA 70B INT4 zuverlässige SQL-Abfragen aus natürlicher Sprache für dieses Schema zu generieren. Daher wurde eine interne Bedrohungsdatenbank mit einer vereinfachten Struktur entwickelt, die sich besser für das Training des LLM und die SQL-Generierung eignete. Diese Datenbank wurde mit Bedrohungsinformationen aus Quellen wie

[6] https://arxiv.org/abs/2407.15186

Tenable, Qualys, Microsoft Defender und NVD gefüllt, anstelle der ursprünglich vorgesehenen Daten aus der IKARUS TIP.

Testdaten-Set und Pipeline: Ein spezielles Test-Set aus Benutzeranfragen und zugehörigen SQL-Beispielen wurde erstellt, um das Modell zu trainieren und zu validieren. Im Vergleich zur Erstellung von Frage-Antwort-Paaren für die IKARUS TIP-Datenbank erwies sich dieser Schritt für von uns erstellte vereinfachte Bedrohungsdatenbank als deutlich einfacher, da die vereinfachte interne Datenbank bekannt war und die Antworten direkt validiert werden konnten. Basierend auf diesen Tests wurde eine stabile SQL-Generierungspipeline implementiert, die den gesamten Prozess automatisiert und durch eine sogenannte "Explainability Trace" Transparenz gewährleistet. Diese Funktion dokumentiert die Entscheidungsfindung des LLMs bei der SQL-Generierung, um einerseits die EU-AI-Act-Kompatibilität voranzutreiben und andererseits Cybersecurity-Expert:innen die Nachvollziehbarkeit der Schritte zu ermöglichen.

Verzögerungen und Probleme

Ein erheblicher Teil der Projektzeit wurde investiert, um das komplexe TIP-Datenbankschema zu verstehen und zu verarbeiten. Letztendlich erwies sich die Struktur jedoch als zu abstrakt und komplex für eine direkte Integration. Zudem verhinderten die Hardware-Beschränkungen den parallelen Einsatz von SQL-spezifischen und allgemeinen LLMs, was die Weiterentwicklung der Lösung einschränkte.

Abweichungen, Anpassungen und Auswirkungen

Das Team entschied sich, von der direkten Integration mit der IKARUS TIP-Datenbank abzusehen und stattdessen eine interne Bedrohungsdatenbank zu entwickeln. Diese Umstellung

beschleunigte die Entwicklung, da die SQL-Abfragenerstellung vereinfacht wurde. Gleichzeitig verhinderte sie weitere Verzögerungen durch die Komplexität des IKARUS TIP-Datenbankschemas. Die Nutzung der internen Datenbank ermöglichte die Bereitstellung einer funktionalen SQL-Generierungspipeline, die innerhalb der bestehenden Hardware-Beschränkungen stabil und performant bleibt. Dieser Ansatz stellte sicher, dass die Kernfunktionalität der natürlichen Sprachsuche erfolgreich implementiert wurde, trotz der ursprünglichen Herausforderungen.

Fertigstellungsgrad

Integration mit der IKARUS TIP-Datenbank: Nicht abgeschlossen.

Minerva-Produktfunktionalität: 100% abgeschlossen.

Die Abbildung unten zeigt, dass das LLM die Frage des Nutzers als eine Datenbankanfrage klassifiziert hat, daraufhin die passende Datenbankabfrage generiert und abschließend die Antwort geliefert hat.

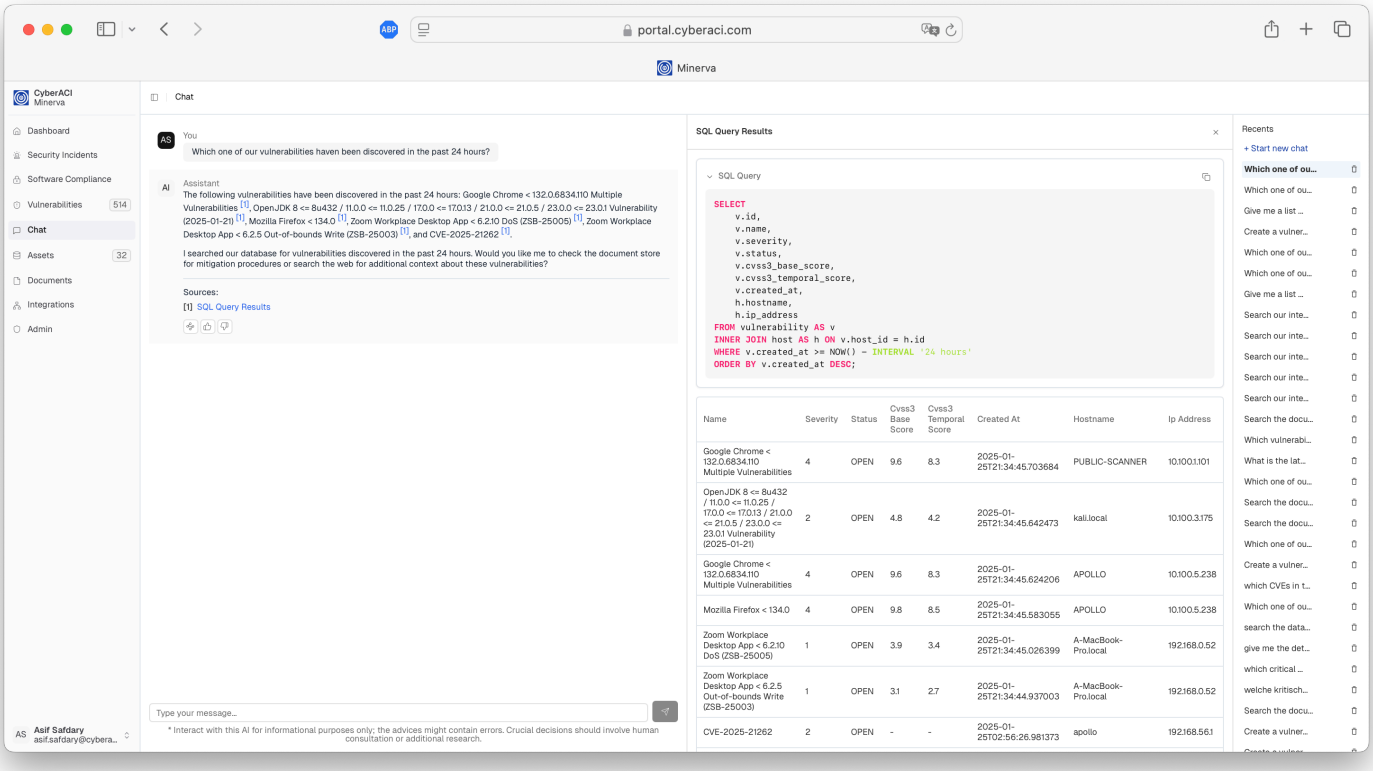


Abbildung 6: Minerva Datenbank Fragebeantwortung

2.4. Multi-hop Frage-Antwort in Kombination mit SQL und Dokumentenfunktionalität und Kostenoptimierung

Multi-hop-Frage-Antwort beschreibt die Fähigkeit eines LLM, mehrere Schritte sequenziell auszuführen, wobei das Ergebnis eines Schritts den nächsten leitet, um eine einzelne Benutzeranfrage zu beantworten[7]. Ein Schritt kann beispielsweise eine Datenbanksuche oder eine Dokumentensuche umfassen. Diese Funktionalität ist entscheidend, um komplexe Fragen zu beantworten, die iterative Analysen über mehrere Datenquellen hinweg erfordern.

[7] https://arxiv.org/abs/2204.09140

Durchgeführte Arbeiten

Die SQL-Generierungsfunktionalität wurde bereits in der Planungsphase integriert, um eine stabile Grundlage für Multi-hop-Frage-Antworten zu schaffen. Anstelle klassischer Kostenoptimierungsstrategien lag der Fokus auf der maximalen Nutzung der vorhandenen GPU-Ressourcen. Der Betrieb großer LLMs wie Llama 70B wurde innerhalb der begrenzten VRAM-Kapazitäten so effizient wie möglich gestaltet. Dazu wurden quantisierte Modelle eingesetzt, um den Ressourcenbedarf zu reduzieren, und die Inferenz-Einstellungen wurden sorgfältig konfiguriert, um die Hardwareleistung optimal auszunutzen. Aufgaben wurden auf Abhängigkeiten hin analysiert. Nicht abhängige Aufgaben wurden parallelisiert und in Batches verarbeitet, um die Ausführungszeit zu optimieren und Verzögerungen zu minimieren.

In der IKARUS-Infrastruktur ermöglichte der Einsatz quantisierter LLMs den Betrieb größerer Modelle auch unter eingeschränkten Ressourcen[8]. Zusätzliche Optimierungen der Inferenz-Frameworks steigerten die Leistung weiter. Ein dynamischer Multi-hop-Agent wurde entwickelt, der als Entscheidungsprozess fungiert. Dieses System arbeitet als Entscheidungsschleife, in der Zwischenergebnisse ausgewertet werden. Dabei nutzt es verfügbare Werkzeuge wie Text-to-SQL und RAG, um die nächsten Schritte bei der Beantwortung komplexer Fragen effizient zu planen und durchzuführen.

Diese Funktionalität der Unterteilung der Fragen und Antwortgenerierung wurde vollständig implementiert und ermöglicht es dem LLM, Benutzerfragen in umsetzbare Schritte zu zerlegen.

[8] https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html

Diese Schritte werden iterativ mithilfe der verfügbaren Werkzeuge gelöst, um eine bestmögliche Antwort zu generieren, selbst wenn nicht alle Fragen vollständig beantwortet werden können.

Herausforderungen und Abweichungen

Der Fokus der Kostenoptimierung verlagerte sich von der Auswahl kleinerer Modelle hin zur effektiven Nutzung der bestehenden GPU-Infrastruktur. Das Team entschied sich, ein einziges leistungsstarkes LLM (LLaMA 70B) für alle Aufgaben einzusetzen. Dies stellte einen Kompromiss zwischen Ressourceneinschränkungen und Systemleistung dar. Die Entwicklung des Multi-hop Agents erforderte umfangreiche Tests, um sicherzustellen, dass Zwischenergebnisse korrekt verarbeitet und Aktionen dynamisch angepasst werden konnten. Aufgrund von Hardware-Beschränkungen war es jedoch nicht möglich, mehrere spezialisierte LLMs für die Optimierung spezifischer Aufgaben einzusetzen. Dies vereinfachte die Infrastruktur, schränkte jedoch potenzielle Leistungssteigerungen ein. Daher konnte die Multi-hop-Funktionalität auf der IKARUS TIP nicht genutzt werden, da die Text-to-SQL-Generierung auf dieser Plattform nie umgesetzt wurde. Multi-hop wurde jedoch ausschließlich für die Dokumentensuche eingesetzt.

Nach dem Wechsel zu einer Cloud-Lösung wurde der Multi-hop-Dynamic-Planning Agent erfolgreich auf der CyberACI-Infrastruktur implementiert und ist voll funktionsfähig. Durch die Integration von SQL-Generierung, Dokumentensuche und iterativen Prozessen liefert das System präzise und umfassende Antworten auf komplexe Fragen. Die hardwareorientierte Kostenstrategie erwies sich als ausreichend, um die Projektziele ohne Funktionseinbußen zu erreichen.

Fertigstellungsgrad

Dieses Arbeitspaket wurde zu 100 % abgeschlossen. Das Multi-hop-Frage-Antwort Agent ist vollständig implementiert und operativ einsetzbar.

Folgende Abbildung zeigt, dass dass LLM Benutzerfragen in kleinen Schritten zerlegt. Es durchsucht zunächst die Dokumente, leitet die Ergebnisse an die Datenbanksuche weiter und liefert schließlich eine Antwort an den Benutzer.

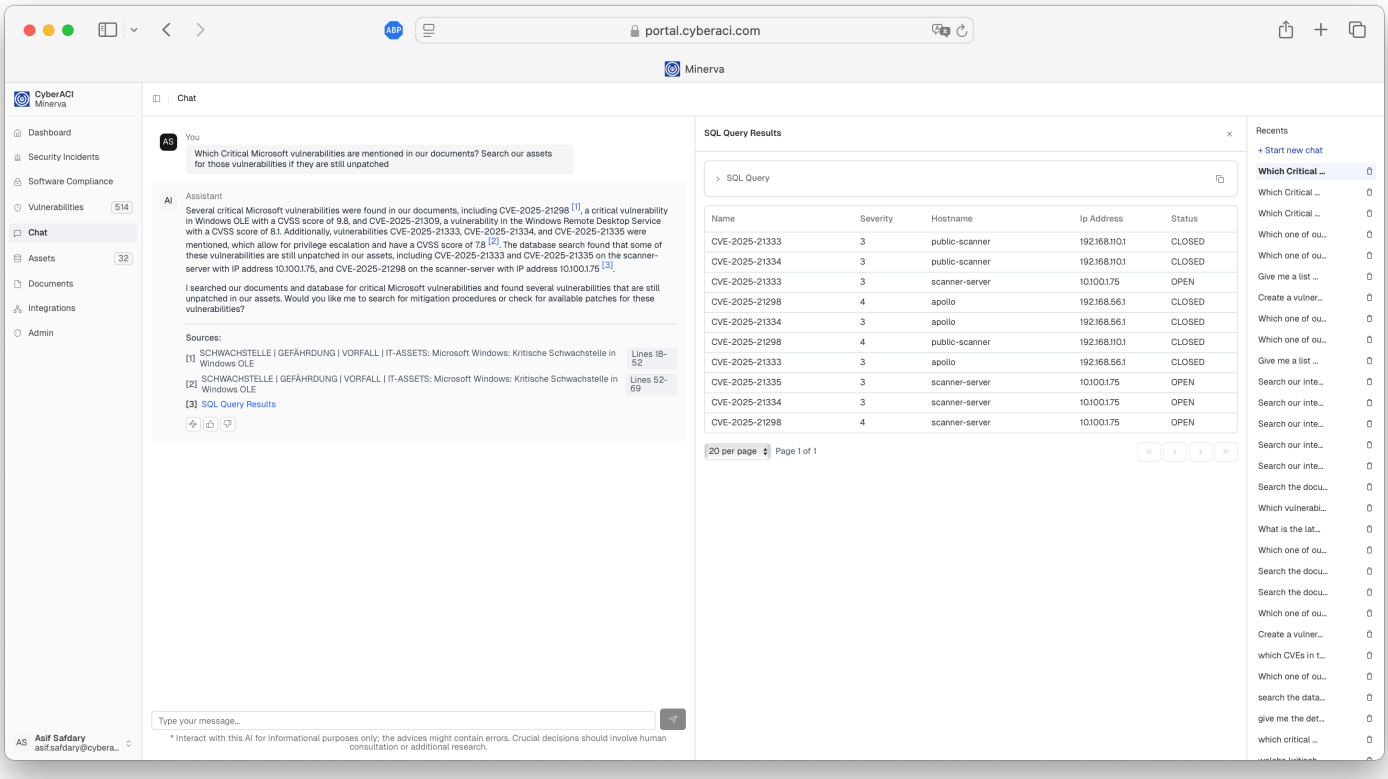


Abbildung 7: Multi-hop Fragebeantwortung

2.5. Testen und Funktionierende On-Premise-Implementierungen

Das Ziel dieses Arbeitspakets war die Validierung der Funktionalität des Systems in einer On-Premise-Umgebung, unter strikter Einhaltung der Anforderungen an Datensouveränität und

Sicherheit. Der Fokus lag insbesondere auf der Stabilität- und Vertraulichkeit des Systems, der SQL-Generierung und der Verarbeitung von Dokumenten.

Durchgeführte Arbeiten

Die Tests wurden ausschließlich von den Minerva-Entwickler durchgeführt. Der Schwerpunkt lag auf der Validierung von Kernfunktionalitäten, wie der Text-to-SQL Generierung innerhalb der TIP-Benutzeroberfläche sowie der Verarbeitung von Anfragen an Cybersecurity Berichte.
Es wurden mehrere Feedback-Sitzungen organisiert, um vorläufige Ergebnisse zu bewerten. Diese Sitzungen umfassten jedoch keine umfassenden Tests mit Endnutzer:innen oder produktive Einsätze. Alle Tests fanden in einer strikt abgeschotteten On-Premise Umgebung statt. Damit wurde sichergestellt, dass keine Daten die Infrastruktur von IKARUS verlassen und alle Anforderungen an Datensouveränität und Sicherheit erfüllt wurden.

Herausforderungen und Abweichungen

Die abstrakte Struktur der TIP-Datenbank stellte eine erhebliche Herausforderung dar, insbesondere bei der Validierung der Text-to-SQL Generierung. Aufgrund der Komplexität des Schemas waren die Ergebnisse nicht immer zuverlässig. Daher entschied sich das Team, eine interne, einfacher strukturierte Datenbank einzusetzen, um die Validierung der LLM-generierten Ergebnisse zu erleichtern.
Zusätzlich begrenzten die Hardware-Ressourcen der On-Premise-Umgebung die Verarbeitung großer Mengen an Dokumenten. Eine Analyse ergab, dass die bestehende Infrastruktur nicht effizient genug war, um alle relevanten IKARUS-Dokumente zu verarbeiten. Deshalb wurde der

Fokus auf öffentlich verfügbare Cybersecurity-Dokumente gelegt, um die Weiterentwicklung voranzutreiben.

Ergebnisse

Die Funktionalität des Systems wurde erfolgreich validiert. Trotz der Herausforderungen mit der TIP-Datenbank konnte die SQL-Generierung in einer angepassten Umgebung stabil getestet werden.

Alle Tests wurden unter strenger Einhaltung der Anforderungen an Datensouveränität durchgeführt. Es wurde sichergestellt, dass keine sensiblen Daten die Infrastruktur von IKARUS verlassen haben.

Fertigstellungsgrad

Die Tests in der On-Premise-Umgebung wurden zu 100 % abgeschlossen. Obwohl keine umfassenden Nutzertests mit Endanwender:innen durchgeführt wurden, bietet die Validierung eine solide Grundlage für die weitere Optimierung und den produktiven Einsatz des Systems.

Abbildungsverzeichnis

Abkürzungsverzeichnis

AI – Artificial Intelligence (Künstliche Intelligenz)

CVE – Common Vulnerabilities and Exposures

EDR – Endpoint Detection and Response

GPU – Graphics Processing Unit

TIP –Threat Intelligence Platform

KI – Künstliche Intelligenz

LLM – Large Language Model

NLP – Natural Language Processing

NVD – National Vulnerability Database

RAG – Retrieval-Augmented Generation

SIEM – Security Information and Event Management

SQL – Structured Query Language