An Agentic GraphRAG Architecture for Organization-Aware Cyber Threat Intelligence

Philipp Fuxen
Department of Computer Science and Mathematics
Ostbayerische Technische Hochschule Regensburg
Regensburg, Germany
e-mail: philipp.fuxen@oth-regensburg.de

Prof. Dr. Rudolf Hackenberg
Department of Computer Science and Mathematics
Ostbayerische Technische Hochschule Regensburg
Regensburg, Germany
e-mail: rudolf.hackenberg@oth-regensburg.de

***Abstract*—Cyber Threat Intelligence (CTI) is a key component of modern security operations, yet existing systems struggle to integrate heterogeneous intelligence sources and to provide contextualized, organization-aware analysis. Knowledge-graph-based approaches offer structured and explainable representations of threats, while Large Language Models (LLMs) enable flexible semantic analysis of unstructured information. However, these paradigms are typically applied in isolation and fail to support continuous, context-driven threat modeling. This paper proposes an Agentic Graph Retrieval-Augmented Generation (GraphRAG) architecture for CTI analysis. The approach integrates structured and unstructured CTI into a persistent knowledge state consisting of a cybersecurity knowledge graph, a document store, and a vector index. Graph-first retrieval is combined with agentic orchestration to iteratively assemble bounded, task-relevant context for LLM-based reasoning, enabling grounded, explainable, and organization-specific threat analysis. We present a layered reference architecture and a detailed methodology describing knowledge construction, graph-first retrieval, agentic analysis workflow, and grounded reasoning. The proposed approach is designed to support security analysts and automated security workflows in operational CTI environments and provides a foundation for adaptive and explainable CTI systems that combine structured knowledge representation with flexible AI-driven analysis.**

Keywords-cyber threat intelligence; knowledge graph; graph retrieval-augmented generation; agentic AI systems; large language models; context-aware threat analysis; explainable AI.

I. Introduction

Cyber Threat Intelligence (CTI) has become a central component of modern security measures, enabling companies to detect, analyze, and respond to cyber threats in a timely manner. However, the increasing volume, speed, and heterogeneity of threat data pose a significant challenge for existing CTI systems. Security-related information is scattered across structured feeds, logs, vulnerability databases, and unstructured text sources, including incident reports, security blogs, and dark web forums. As a result, security analysts are confronted with fragmented information landscapes that hinder the correlation, contextualization, and prioritization of emerging threats. [1]

To address this challenge, recent research has explored knowledge-based representations and Artificial Intelligence (AI) techniques for automating CTI. Cybersecurity Knowledge Graphs (CSKGs) are often used to model relationships between threat actors, vulnerabilities, attack techniques, and assets in a structured and traceable manner [2]. While graph-based approaches enhance reasoning and traceability, they often rely on manual curation or static data ingestion pipelines, which limits their adaptability to rapidly changing threat landscapes [3]. This restricts their ability to reflect evolving attack campaigns, newly observed tactics, and organization-specific threat contexts in time. In parallel, Large Language Models (LLMs) have shown promising results in extracting threat entities, tactics, and techniques from unstructured text sources [4]. Despite their flexibility, LLM-based approaches suffer from limited consistency, a lack of explicit structure, and challenges regarding explainability and validation, which limit their use in operational CTI systems.

Although both paradigms offer complementary strengths, they are typically applied in isolation in existing CTI systems. Graph-based systems provide a persistent and structured knowledge state but lack flexible semantic reasoning over newly observed intelligence, while LLM-based pipelines offer powerful semantic analysis but operate without a stable, verifiable memory. Consequently, current approaches do not support continuous, context-aware threat modeling that combines a structured persistent knowledge state with adaptive semantic reasoning. To our knowledge, the integration of LLM-based reasoning and knowledge-graph-based retrieval as a unified Graph Retrieval-Augmented Generation (GraphRAG) pipeline remains largely unexplored in existing CTI architectures.

In this paper, we propose an Agentic GraphRAG-based CTI architecture that combines LLMs and CSKGs to enable dynamic and organization-specific threat analysis. Heterogeneous structured and unstructured CTI sources are integrated into a persistent CSKG, which is subsequently used as a retrieval layer to provide context for LLM-driven reasoning tasks, such as threat prioritization, contextual analysis, and explanation generation. The main contributions of this work can be summarized as follows:

- An agentic GraphRAG architecture for organization-aware CTI analysis
- A methodology for constructing and maintaining a persistent knowledge state from heterogeneous CTI sources
- A graph-first retrieval and analysis concept enabling grounded and explainable CTI reasoning

The rest of this paper is organized as follows: Section II reviews related work, Section III defines the problem and design

goals, Section IV introduces the system architecture, Section V describes the proposed Agentic GraphRAG methodology, and Section VI concludes the paper and outlines future work.

II. Related work

This section examines related work in the areas of CTI processing, CSKGs, and CTI analysis based on LLMs. We structure the discussion along the pipeline of the proposed GraphRAG architecture to highlight the limitations of current approaches and motivate a unified solution. Recent work has begun to explore hybrid retrieval and reasoning pipelines that combine knowledge graphs and language models, but these approaches are not yet systematically formulated as persistent agentic GraphRAG pipelines for continuous CTI analysis.

A. CTI Feed Integration and Knowledge Graph Construction

CTI systems are typically based on standardized exchange formats, such as Structured Threat Information eXpression (STIX) and Trusted Automated eXchange of Indicator Information (TAXII), which provide interoperable representations and transport mechanisms for cross-platform exchange of CTI objects and collections [5][6]. While these standards are widely used, they primarily address syntactic interoperability and do not inherently provide semantic enrichment or organization-specific contextualization.

To support more comprehensive integration and reasoning across heterogeneous CTI, recent work increasingly uses CSKGs as structured, explainable representations of entities, such as threat actors, malware, vulnerabilities, techniques, and affected assets. Survey papers highlight CSKGs as a holistic approach to merging different cybersecurity data sources and enabling large-scale correlations and reasoning [2][7]. Furthermore, construction-oriented review papers discuss current pipelines for CSKG construction, including extraction, fusion, and inference components, and outline open challenges related to automation and data quality [3]. However, many CSKG construction approaches remain constrained by manual curation, schema engineering, or batch-oriented collection, limiting their ability to continuously integrate rapidly evolving and highly unstructured CTI sources.

B. Graph-based Threat Modelling and Contextualisation

Graph-based threat modeling has a long tradition in cybersecurity, including attack graphs and related graphical security models that support the analysis of multi-stage exploits and the prioritization of remediation measures. A recent survey summarizes the state of the art in the automatic generation and use of attack graphs and related models, highlighting challenges related to scalability, maintenance, and automation in rapidly changing vulnerability and threat landscapes [8]. Beyond classic attack graphs, more comprehensive research on graph mining in cybersecurity shows how graph representations are used for detection, correlation, and security analysis tasks across different data modalities [9].

In the CTI field in particular, work such as HINTI uses graph-based learning to highlight the interrelationships between heterogeneous indicators in order to quantify relevance and improve intelligence mining, thereby demonstrating the advantages of structured graph representations for CTI reasoning [10]. Nevertheless, many graph-based approaches assume that relevant knowledge is already available in structured form. They usually offer only limited mechanisms for the continuous integration of unstructured narrative information (reports, blogs, forum posts) and for the dynamic selection of organization-specific subgraphs as context for downstream conclusions.

C. Large Language Models for CTI Analysis

LLMs have enabled new forms of CTI extraction and analysis from unstructured text. Benchmarking initiatives, such as CTIBench systematically evaluate the performance of LLMs on CTI tasks and highlight both the strengths and limitations of LLMs in the CTI context [4]. Complementary work proposes fully LLM-driven or LLM-assisted approaches for processing CTI reports, with the goal of reducing the manual workload of analysts while leveraging the semantic capabilities of modern models [11]. Despite their flexibility, LLM-only pipelines often lack a persistent, verifiable state. This limits the traceability, reusability, and validation of extracted threat intelligence and increases the risk of hallucinations and inconsistent results, which are important considerations for operational CTI deployment.

D. Hybrid LLM-Knowledge Graph Approaches and GraphRAG

To combine structured representations with semantic extraction, current work uses LLMs to create or enrich CSKGs from CTI text sources. CTINexus proposes optimized contextual learning to support data-efficient CTI extraction and CSKG creation in data-scarce environments [12]. CTIKG uses prompt-based segmentation and multiple agents to create security-oriented knowledge graphs from CTI articles while accounting for the limitations of LLMs, such as hallucinations and context constraints [13]. These approaches demonstrate that LLMs can effectively populate and enrich CSKGs, but they focus primarily on graph construction or offline enrichment rather than continuous analysis.

Recent benchmark work further highlights the importance of knowledge-augmented reasoning for CTI. CTIARENA introduces a comprehensive evaluation suite covering structured, unstructured, and hybrid CTI tasks under both closed-book and retrieval-augmented settings [14]. Their results show that LLM performance remains limited without external knowledge and improves when augmented with security-specific retrieval strategies, including CSKG-guided retrieval. However, CTIARENA focuses on benchmarking and evaluation rather than proposing an operational architecture or methodology for continuous CTI analysis.

In contrast, our work conceptualizes CTI analysis as a persistent agentic GraphRAG process: a CSKG acts as a persistent knowledge state, graph-based retrieval selects organization-specific contexts (e.g., relevant entities, relationships, and subgraphs), and the LLM performs informed reasoning and explanation over the retrieved context.

E. Summary

In summary, previous work has advanced standardized CTI sharing via STIX/TAXII, CSKG construction and graph-based security analysis, and LLM-based CTI extraction and evaluation. Recent benchmark studies further confirm the importance of knowledge-augmented reasoning for CTI, while highlighting the limitations of LLMs when operating without persistent external knowledge. However, existing approaches typically focus either on structured graph construction or on isolated retrieval-augmented analysis and do not yet systematically integrate persistent knowledge representation, graph-based retrieval, and organization-aware LLM reasoning into a unified GraphRAG pipeline for continuous CTI analysis.

III. Problem Statement and Design Goals

This section derives the core problem and design goals of the proposed approach based on the constraints identified in the previous sections. First, we formalize the challenges of current CTI systems in terms of continuous knowledge integration, contextualization, and organization-specific analysis. We then define a set of design goals that serve as a guide for the architecture and method presented in the following sections.

A. Problem Statement

Modern CTI systems must integrate large amounts of heterogeneous data from structured feeds, internal telemetry, and unstructured text sources. While existing approaches offer either structured threat representations or flexible semantic analyses, they mostly do not support continuous, organization-aware threat modeling in a consistent manner. Knowledge-graph-based CTI systems offer persistent and explainable representations, but lack mechanisms for adaptive semantic interpretation of emerging information. Conversely, LLM-based CTI pipelines offer semantic extraction and analysis capabilities, but operate without a stable, verifiable knowledge state, limiting traceability and reusability.

From an operational perspective, security analysts need contextual threat assessments that reflect both global threat activity and local organizational conditions, such as the criticality, vulnerability, and relevance of assets. Current systems typically treat threat intelligence as generic and static, offering limited support for tailoring analysis to organization-specific environments. This gap leads to delayed prioritization, incomplete situational awareness, and increased manual effort for analysts.

To address these limitations, a CTI system must (i) maintain a persistent representation of threat knowledge, (ii) continuously integrate heterogeneous and evolving information, and (iii) enable adaptive reasoning about this knowledge in an organization-specific context. These requirements motivate a GraphRAG approach, in which a CSKG serves as a persistent structured knowledge state and retrieval mechanism, while LLMs draw informed conclusions about retrieved subgraphs.

B. Design Goals

Based on the identified constraints and requirements, the proposed architecture is guided by this main design goals:

- **DG1 – Persistent and structured knowledge state:** The system should maintain a continuously evolving CSKG that integrates structured and unstructured CTI sources into a unified, verifiable representation.
- **DG2 – Continuous integration of heterogeneous CTI sources:** The architecture must support the automated collection and normalization of various CTI feeds, including structured indicators and unstructured text, to enable timely updating of threat knowledge.
- **DG3 – Graph-based contextual querying:** The system must enable selective querying of relevant subgraphs based on organizational context, e.g., asset exposure, sector, or threat relevance, to provide targeted inputs for analysis.
- **DG4 – LLM-based reasoning and explainability:** LLMs may only be used for interpreting retrieved graph context to enable explainable and traceable threat analysis while avoiding unfounded generations.
- **DG5 – Organization-specific threat analysis:** The architecture must support the prioritization and interpretation of threats tailored to individual organizational environments, rather than generating general threat summaries.
- **DG6 – Agentic task orchestration:** The system should support task-driven orchestration of retrieval, reasoning, and auxiliary analysis steps to enable iterative and goal-oriented CTI analysis workflows.

These design goals form the basis for the system architecture described in the following section and ensure that the proposed approach directly addresses the limitations of existing CTI systems identified in Sections I and II.

IV. System Architecture

The proposed architecture implements an Agentic GraphRAG approach for analyzing CTI. It is designed as a multi-layered architecture that separates data processing, knowledge representation, retrieval, orchestration, and interaction. Figure 1 illustrates the overall architecture, highlighting the role of the CSKG as the persistent knowledge state of the system. A typical analysis starts with a user-defined task, which is interpreted by the agentic orchestration layer to plan and execute a sequence of graph-first retrieval and reasoning steps over the persistent knowledge state. This layered separation reflects the design goals defined in Section III by decoupling knowledge construction, grounded retrieval, and task-driven control.

A. CTI Data Sources

The architecture integrates heterogeneous sources of CTI, including structured CTI feeds (e.g., indicators, vulnerability data), unstructured information, such as reports and alerts, and organization-specific contextual information. These sources provide the raw data for knowledge building but are not directly accessed during analysis.

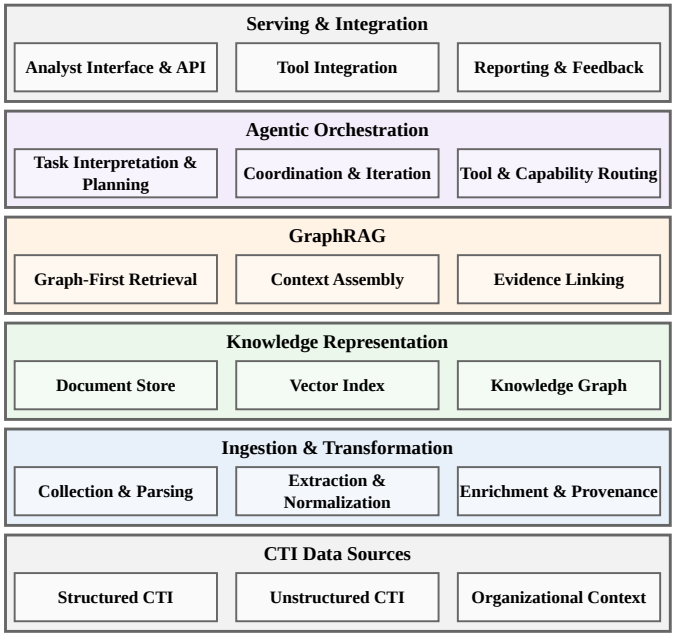

Figure 1. System architecture of the proposed approach.

B. Ingestion and Transformation Layer

All incoming CTI data is collected, analyzed, and converted into a uniform internal representation. This layer performs entity and relationship extraction, normalization, enrichment, and provenance annotation to transform raw data into structured knowledge elements. By consolidating both structured and unstructured sources, the system ensures consistent and traceable integration of information prior to storage.

C. Knowledge Representation Layer

The knowledge representation layer maintains the persistent state of the system and consists of three complementary components: a document store, a vector index, and a CSKG. The document store preserves the original source material for traceability and evidence verification. The vector index enables semantic similarity search over textual intelligence. The CSKG models entities, relationships, and temporal connections, providing a structured and comprehensible representation of the threat landscape. All subsequent analysis steps operate on this unified knowledge state.

D. GraphRAG Layer

The GraphRAG layer provides structured retrieval capabilities over the persistent knowledge state. It performs graph-first retrieval to identify relevant subgraphs based on the analysis task and organizational context, and assembles these subgraphs into a bounded context for downstream reasoning. The linking of evidence ensures that the retrieved graph elements can be traced back to the original documents or sources, enabling traceable and explainable analysis. This layer forms the primary grounding mechanism for downstream reasoning and ensures that LLM-based analysis remains anchored in verified graph context.

E. Agentic Orchestration Layer

The agentic orchestration layer interprets user requests and analysis tasks and decomposes them into executable steps. It coordinates retrieval, reasoning, and auxiliary capability invocation in an iterative manner. Depending on the task, the orchestration layer may trigger multiple retrieval rounds, refine queries, or invoke specialized functions, such as scoring, validation, or enrichment. This design enables flexible, task-oriented analysis while keeping reasoning grounded in retrieved context rather than exposing raw data or the full knowledge base to the reasoning components. By separating control logic from knowledge storage and reasoning, the orchestration layer supports adaptive, goal-driven CTI workflows.

F. Serving and Integration

The service and integration layer makes the results of the analysis available to users and external systems. It provides analyst interfaces, APIs, and integration points for security platforms, such as Security Information and Event Management (SIEM) or Security Orchestration, Automation, and Response (SOAR) systems. The feedback generated in this layer can be incorporated into the persistent knowledge state to support the continuous development of the threat model.

The following section details the GraphRAG workflow and the agentic analysis process in a step-by-step manner.

V. Methodology

This section describes the proposed Agentic GraphRAG methodology for CTI analysis. An overview of the methodology is shown in Figure 2 as an iterative analysis workflow. Building on the layered architecture introduced in Section IV, the methodology specifies how knowledge is constructed, retrieved, and analyzed to support organization-aware threat assessment. The following subsections detail the problem formulation, knowledge construction process, graph-first retrieval strategy, agentic analysis workflow, and grounded reasoning approach.

A. Problem Formulation

The objective of the proposed methodology is to support organization-specific cyber threat analysis by combining persistent, structured threat knowledge with adaptive, task-driven reasoning. Given the continuously growing volume and heterogeneity of CTI, analysts require mechanisms that not only retrieve relevant information but also contextualize, correlate, and prioritize threats with respect to their organization.

In our setting, the system maintains a persistent knowledge state that consists of (i) a CSKG capturing entities, relations, and temporal context, (ii) an associated document store preserving original sources as evidence, and (iii) a vector index enabling semantic similarity search over textual intelligence. An analysis task provided by a user or an external system defines the analytical objective, such as threat prioritization, exposure assessment, or campaign interpretation, optionally parameterized by organizational context (e.g., assets, sector, or risk posture).

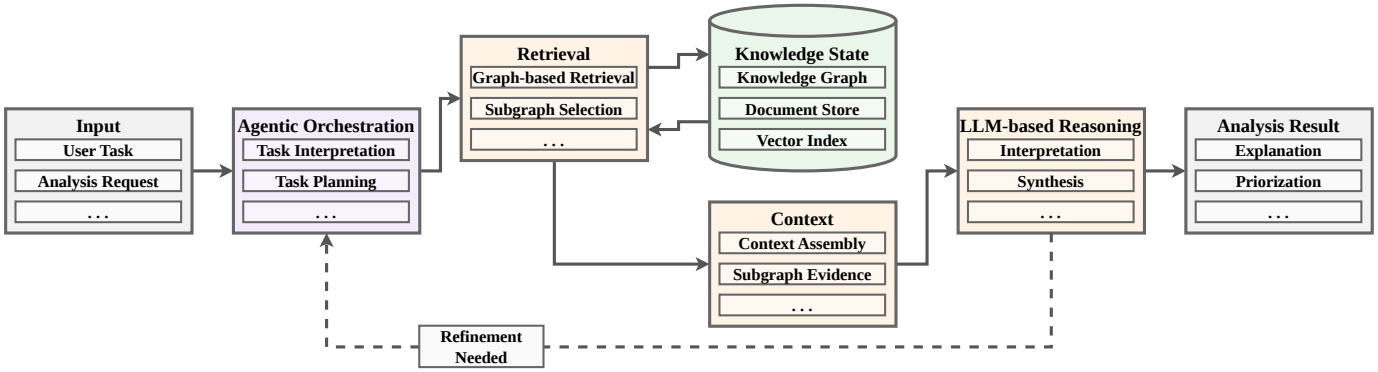

Figure 2. Agentic GraphRAG methodology for CTI analysis.

The methodology aims to produce analysis results that are grounded in verified knowledge and traceable to original evidence. To this end, the process iteratively performs graph-first retrieval to select coherent subgraphs relevant to the task and organizational context, assembles a bounded context window including linked evidence, and applies LLM-based reasoning over this structured input. Agentic orchestration controls the sequence of retrieval and reasoning steps, enabling multi-step analysis, query refinement, and auxiliary capability invocation while keeping the reasoning component isolated from raw sources and the full knowledge base.

This formulation ensures that analysis results remain explainable, organization-aware, and adaptable to evolving threat landscapes while preserving a persistent and verifiable knowledge state.

B. Knowledge Construction Process

To realize the persistent knowledge state defined in the problem formulation, the methodology implements a continuous knowledge construction process that transforms heterogeneous CTI inputs into a unified and verifiable representation supporting structured retrieval and reasoning. This process operates continuously and incrementally, enabling the system to integrate newly observed intelligence without requiring complete reconstruction of existing knowledge.

Structured CTI sources are mapped to a canonical schema and directly transformed into graph entities and relationships. Unstructured sources, such as reports or advisories, are processed using extraction techniques to identify relevant entities, relations, and contextual attributes. All extracted elements are normalized, enriched, and annotated with provenance and temporal metadata to ensure consistency and traceability.

The resulting knowledge is stored in three tightly coupled representations. Original documents are preserved in a document store, providing verifiable evidence and auditability. Textual content is embedded into a vector index to enable semantic similarity search and matching. In parallel, extracted entities and relationships are integrated into the CSKG, where they are linked to corresponding documents and vector representations.

By consolidating structured and unstructured intelligence into a single persistent knowledge state, the construction process establishes a stable and extensible foundation for graph-first retrieval and agentic analysis. This separation between knowledge construction and knowledge utilization ensures that downstream reasoning operates over curated and verifiable context rather than raw and potentially noisy input data.

C. Graph-Based Retrieval

Graph-first retrieval provides the primary mechanism for grounding all downstream analysis in structured and organization-relevant knowledge. Given a task provided by a user or an external system and optional organizational context, the retrieval process operates over the CSKG to identify coherent subgraphs that capture relevant threat entities, relationships, and temporal dependencies.

Retrieval begins by translating the analysis task into one or more graph queries that encode semantic intent and contextual constraints. These queries may incorporate organizational factors, such as asset exposure, sector-specific relevance, or previously observed activity. Rather than retrieving isolated entities or documents, the process selects connected subgraphs that preserve relational structure and enable holistic interpretation of threats.

The retrieved subgraphs are subsequently ranked and filtered based on relevance, structural coherence, and contextual alignment. To support explainability, each graph element is linked to corresponding evidence in the document store and vector index, allowing retrieved context to be traced back to original intelligence sources.

By operating on explicit relationships and graph topology, the proposed retrieval strategy enables precise context selection that is not achievable with purely text-based retrieval. This graph-first approach ensures that all reasoning steps are grounded in structured knowledge and that retrieved context remains bounded, coherent, and verifiable.

D. Agentic Analysis Workflow

The agentic analysis workflow controls the execution of retrieval and reasoning steps in a task-driven and iterative manner. When an analysis task is provided by a user or external system, the orchestration layer interprets the task objective and decomposes it into a sequence of intermediate goals, such as

identifying relevant threats, correlating activities, or assessing organizational exposure.

For each intermediate goal, the orchestration layer determines the required retrieval and processing actions. This may include issuing graph-first retrieval queries, refining previously retrieved context, or invoking auxiliary capabilities, such as scoring, validation, or enrichment. Retrieved subgraphs and associated evidence are incrementally assembled into a bounded analysis context that is passed to the reasoning component.

The workflow supports iterative refinement by allowing the orchestration layer to re-evaluate intermediate results and trigger additional retrieval or processing steps when necessary. Iteration continues until the task objectives are satisfied or predefined stopping criteria are met, ensuring that the analysis remains focused and bounded.

By explicitly separating control logic from reasoning, the agentic workflow enables adaptive, multi-step analysis while maintaining strict grounding in verified knowledge, allowing the system to dynamically adjust retrieval and reasoning strategies based on intermediate results without exposing the full knowledge base or raw data to the reasoning model.

E. Reasoning and Output Generation

Reasoning is performed exclusively over the bounded and verified context assembled by the graph-first retrieval and agentic orchestration layers. The reasoning component receives a structured subgraph, linked evidence from the document store, and task-specific instructions, ensuring that analytical conclusions remain grounded in persistent knowledge.

The reasoning process focuses on interpreting relationships, identifying patterns, and synthesizing insights relevant to the analysis objective. Typical outputs include threat prioritization, contextualized explanations, exposure assessments, or hypothesis generation. Because the reasoning model operates exclusively on retrieved context, the risk of hallucination is reduced and outputs remain traceable to original sources.

Generated results are returned through the serving layer and may include references to supporting evidence or graph elements to support analyst validation. Optionally, validated outputs can be incorporated back into the persistent knowledge state, enabling incremental refinement of the threat model.

VI. Conclusion and Future Work

This paper introduced an Agentic GraphRAG architecture and methodology for organization-aware CTI analysis. The proposed approach combines a CSKG with graph-first retrieval, agentic orchestration, and grounded LLM-based reasoning to address the challenges of heterogeneity, scale, and contextualization in modern CTI environments. By separating knowledge construction, retrieval, control, and reasoning, the architecture enables continuous integration of structured and unstructured intelligence while maintaining traceability and explainability. The agentic analysis workflow supports multi-step and context-driven threat assessment without exposing raw data or the full knowledge base to the reasoning model, thereby reducing hallucination risks and improving analytical reliability.

Future work will focus on extending the prototype implementation and conducting systematic evaluations in realistic operational scenarios. Planned directions include quantitative assessment of retrieval quality and reasoning accuracy, as well as comparisons against baseline approaches such as text-based RAG pipelines, CSKG-only retrieval, and LLM-only analysis. Further work will investigate scalability aspects for continuously growing CTI datasets, including incremental graph updates and efficient subgraph retrieval. Robustness considerations such as knowledge graph poisoning and prompt injection attacks will also be explored. In addition, the knowledge graph will be extended to support temporal and causal reasoning to enable deeper analysis of evolving attack campaigns.

Overall, the proposed architecture establishes a foundation for adaptive and explainable CTI systems that integrate persistent structured knowledge with flexible AI-driven analysis.

References

[1] ENISA, "Enisa threat landscape 2025", European Union Agency for Cybersecurity, 2025. DOI: 10.2824/1946374.

[2] L. F. Sikos, "Cybersecurity knowledge graphs", *Knowledge and Information Systems*, vol. 65, no. 2, pp. 351–379, 2023. DOI: 10.1007/s10115-022-01734-4.

[3] X. Zhao, R. Jiang, Y. Han, A. Li, and Z. Peng, "A survey on cybersecurity knowledge graph construction", *Computers & Security*, vol. 136, p. 103 524, 2024. DOI: 10.1016/j.cose.2023.103524.

[4] M. T. Alam, D. Bhusal, L. Nguyen, and N. Rastogi, *Ctibench: A benchmark for evaluating llms in cyber threat intelligence*, 2024. arXiv: 2406.07599.

[5] OASIS, *STIX Version 2.1*, OASIS Standard, 2021.

[6] OASIS, *TAXII Version 2.1*, OASIS Standard, 2021.

[7] A. M. Konsta and X. D. Koutsoukos, "A survey of automatic generation of attack trees and attack graphs", *Computers & Security*, vol. 125, p. 103 043, 2023. DOI: 10.1016/j.cose.2022.103043.

[8] A.-M. Konsta, A. Lluch Lafuente, B. Spiga, and N. Dragoni, "Survey: Automatic generation of attack trees and attack graphs", *Computers & Security*, vol. 137, p. 103 602, 2024. DOI: 10.1016/j.cose.2023.103602.

[9] B. Yan et al., "Graph mining for cybersecurity: A survey", *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 2, pp. 1–52, 2023. DOI: 10.1145/3610228.

[10] J. Zhao, Q. Yan, X. Liu, B. Li, and G. Zuo, "Cyber threat intelligence modeling based on heterogeneous graph convolutional network", in *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, USENIX Association, 2020, pp. 241–256.

[11] A. Krašovec, G. Steri, G. Karopoulos, and M. Trapani, "Large language models for cyber threat intelligence: Extracting mitre with llms", in *Proceedings of the 20th International Conference on Availability, Reliability and Security (ARES)*, Springer Nature Switzerland, 2025, pp. 80–89.

[12] Y. Cheng, O. Bajaber, S. A. Tsegai, D. Song, and P. Gao, *Ctinexus: Automatic cyber threat intelligence knowledge graph construction using large language models*, 2025. arXiv: 2410.21060.

[13] L. Huang and X. Xiao, "Ctikg: Llm-powered knowledge graph construction from cyber threat intelligence articles", in *First Conference on Language Modeling*, 2024.

[14] Y. Cheng, Y. Liu, C. Li, D. Song, and P. Gao, *Ctiarena: Benchmarking llm knowledge and reasoning across heterogeneous cyber threat intelligence*, 2025. arXiv: 2510.11974.