# CyKG-RAG: Towards knowledge-graph enhanced retrieval augmented generation for cybersecurity

Kabul Kurniawan[1,2,*], Elmar Kiesling[1] and Andreas Ekelhart[3,4]

[1] WU Wien, Institute for Data, Process and Knowledge Management, Vienna, Austria
[2] Austrian Center for Digital Production (ACDP), Vienna, Austria
[3] SBA Research, Vienna, Austria
[4] University of Vienna, Vienna, Austria

## Abstract

Cybersecurity threat detection and analysis have been increasingly prominent research areas in recent years. Like in other fields, the rise of generative AI and Large Language Models (LLMs) have created new opportunities to advance cybersecurity, but also highlighted some of the key challenges LLMs suffer from – including hallucinations, knowledge deficiencies, and lack of ability to handle factual information. To tackle such limitations, Retrieval Augmented Generation (RAG) – which dynamically retrieves relevant information from external sources to enhance the capabilities of LLMs – has shown promise in many domains. However, naïve RAG approaches typically operate on text and do not consider symbolic representations, conceptual meaning and relations in the data. In the cybersecurity domain, such RAG approaches do not allow to consider pivotal aspects such as network structure, threat attack patterns, and intricate factual security knowledge which are essential to identify suspicious activities and combine clues to discover and reconstruct attacks. To address this challenge, we proposed CyKG-RAG, a novel framework that integrates Knowledge Graphs (KGs) with the RAG approach and is tailored to improve cybersecurity detection and analysis. Our framework utilizes the rich semantic relationships and structured data within KGs to provide contextually relevant information, thereby enhancing the accuracy and reliability of cyber threat detection. We validate our approach through real-world use cases, which demonstrate its effectiveness and show promising results for improving cybersecurity measures.

## Keywords

RAG, LLM, Cybersecurity, Knowledge Graph

## 1. Introduction

Knowledge-driven approaches have gained prominence in cybersecurity, facilitating the standardization of terminology and the automation of decision-making processes through the use of ontologies and knowledge graphs. As a result, a number of formal knowledge representations in cybersecurity (e.g., [1, 2, 3, 4, 5, 6]) and methodologies for the construction of knowledge graphs for cybersecurity [7] are now available.

Despite the active development of Semantic Web technologies and the increasing adoption of graph-based methods in various domains, a notable discrepancy persists between the competen-

kabul.kurniawan@wu.ac.at (K. Kurniawan); elmar.kiesling@wu.ac.at (E. Kiesling); andreas.ekelhart@univie.ac.at (A. Ekelhart)
0000-0002-0877-7063 (K. Kurniawan); 0000-0002-0877-7063 (E. Kiesling); 0000-0003-3682-1364 (A. Ekelhart)

CEUR Workshop Proceedings (CEUR-WS.org)

cies of cybersecurity professionals and their capacity to effectively utilize semantic technologies in security processes. The challenge of enabling security professionals to access, query, and leverage knowledge graphs without requiring in-depth expertise in the semantic domain has long been an obstacle. This represents a significant gap, as formal knowledge representations have the potential to alleviate several critical challenges facing cybersecurity professionals, including (i) the global shortage of skilled cybersecurity personnel[1,2], (ii) a steep learning curve [8], (iii) the overwhelming volume of data leading to analyst fatigue and overload[3,4], and (iv) the rapidly evolving threat landscape requiring constant vigilance and adaptability[5].

With the advent of Large Language Models (LLMs) such as ChatGPT, LLama, Claude, and Gemini [9, 10], researchers have started to explore how these models can improve cybersecurity tasks [11]. However, these models face serious limitations. They often struggle with complex reasoning tasks and maintaining consistency across multiple queries, making it difficult to rely on them for complicated cybersecurity challenges [12]. In addition, they often fail when dealing with factual data sets, sometimes introducing irrelevant or incorrect information — a problem commonly referred to as "hallucination", where the model generates content that is not supported by evidence [13]. This raises significant trust issues, as ensuring the quality and reliability of information generated by LLMs remains a formidable challenge, especially in a domain as critical as cybersecurity.

To address these issues in the cybersecurity context, we investigate the integration of domain-specific knowledge graphs with LLMs. Our goal is to support security analysts in tasks such as log analysis and attack detection. Among the various strategies for integrating knowledge into LLMs, including fine-tuning, prompt engineering, and knowledge editing, we focus on Retrieval-Augmented Generation (RAG) [14]. This technique enables LLMs to answer user queries using data sets without the need for additional model training. RAG was originally designed for scenarios where answers are embedded in specific regions of text. However, it often struggles to synthesize insights from disparate but related pieces of information. Such volumes of text can exceed the limits of LLM context windows, and the expansion of such windows may not be sufficient, as information can be "lost in the middle" of longer contexts [15]. Furthermore, RAG encounters difficulties when tasked with understanding and summarizing semantic concepts across large data sets or voluminous documents (e.g., logs), as is often the case in cybersecurity. These challenges underscore the need for formal knowledge representations, which are well suited for defining structured concepts, their relationships, and associated metadata.

In this paper, we present CyKG-RAG, a RAG approach that integrates cybersecurity knowledge represented as graphs. Our framework leverages the rich semantic relationships and structured data within KGs to provide contextually relevant information. We validate our approach through real-world use cases.

---

[1] https://www.isc2.org/Insights/2023/11/ISC2-Cybersecurity-Workforce-Study-Looking-Deeper-into-the-Workforce-Gap

[2] https://www.csoonline.com/article/657598/cybersecurity-workforce-shortage-reaches-4-million-despite-significant-recruitment-drive.html

[3] https://www.forbes.com/sites/edwardsegal/2021/11/08/alert-fatigue-can-lead-to-missed-cyber-threats-and-staff-retentionrecruitment-issues-study/

[4] https://www.openaccessgovernment.org/fighting-alert-fatigue-and-building-resilient-cybersecurity-strategies/139904

[5] https://www.weforum.org/agenda/2024/01/cybersecurity-ai-frontline-artificial-intelligence/

## 2. Related Work

Two strands of research are of particular interest in the context of this paper: (i) modeling of knowledge in the cybersecurity domain, and (ii) graph-informed RAG approaches.

**Knowledge Graphs/Ontologies in Cybersecurity** Formal knowledge representations have been topic of extensive research in the cybersecurity domain. Early research in this field focused on high-level conceptualization of information security knowledge in ontologies (e.g., [16, 1, 17, 18, 19, 20, 21]). These ontologies typically comprise a set of fundamental concepts, such as *asset*, *threat*, *vulnerability*, and *countermeasure*. In addition, many ontologies specialized for particular application domains (e.g., risk management, incident management) have been released. A comprehensive survey and classification of security ontologies can be found in [22].

More recently, a number of initiatives have aimed at developing security ontologies that cover information sharing standards such as CVE, CVSS, and CAPEC. These standards serve as a vital source for contemporary cybersecurity knowledge and enable exchange of vulnerability information. As part of a research project (STUCCO), [23] presents an approach for a cybersecurity knowledge graph that integrates information from both structured and unstructured data sources. The objective in [6] is to integrate heterogeneous knowledge schemas from various cybersecurity systems and standards and to create a Unified Cybersecurity Ontology (UCO) that aligns CAPEC, CVE, CWE, STIX, Trusted Automated eXchange for Indicator Information (TAXII), and Att&ck. [2] took a more comprehensive approach, presenting not only a cybersecurity ontology but also addressing the dynamic nature of the cybersecurity domain and introducing an ETL workflow that updates the knowledge graph as new information becomes available. Moreover, it can be easily linked to locally available data and incorporated into operational scenarios. Open-access Cybersecurity Knowledge Graphs (CSKGs) include CSKG [4], Open-CyKG [5], ATT&CK-KG [24] and the SEPSES-CSKG [2, 3]. For further details on CSKGs, we refer to recent surveys [25, 26, 27, 28].

**LLM and Retrieval Augmented Generation** In Retrieval-Augmented Generation (RAG), relevant information is retrieved from external sources and incorporated into the context window of large language models (LLMs). The framework presented in [29] outlines three distinct RAG paradigms: Naïve RAG, Advanced RAG, and Modular RAG. Naïve RAG approaches are typically based on the conversion of documents into text, the splitting of text into chunks, and the embedding of these chunks into a vector space. Advanced RAG systems have undergone further development to incorporate pre-retrieval and post-retrieval strategies to enhance retrieval quality. Modular RAG systems introduce iterative and dynamic cycles of interleaved retrieval and generation, thereby enhancing the flexibility and effectiveness of the retrieval process.

The intersection of graphs and LLMs in RAG represents an emerging research area with significant contributions in knowledge graph creation [30], completion [31], and causal graph extraction [32, 33]. Advanced RAG systems have also begun to use graph-based indexing [34]. [35] present the G-Retriever method, which enables users to ask questions about a graph via a conversational interface. The method can be fine-tuned to enhance graph understanding via soft prompting. Another approach to add graph reasoning abilities to existing LLMs is

Graph-ToolFormer [36], which aims to enable graph learning. Furthermore, SURGE [37] supports context-relevant and knowledge-grounded dialogues with a KG. In comparison to other knowledge-grounded methods, it employs contrastive learning to guarantee that the generated texts exhibit a high degree of similarity to the retrieved subgraphs. An important approach for our work is FABULA [38], a system to automatically generate intelligence reports for events utilizing contextual narrative features found in OSINT. GraphRAG [39] addresses the challenge that RAG fails on global questions directed at an entire text corpus and hence, introduces an approach that builds a graph-based index from text corpora, and then partitions the graph into hierarchical community structures. For query-focused summarization of an entire corpus, a map-reduce approach is utilized, which first answers the query independently based on each community summary, and then summarizes all relevant partial answers into a final global answer. In the domain of question answering and multi-document question answering (MD-QA) [40], [40] use Knowledge Graph Prompting (KGP) to formulate the right context in prompting LLMs for MD-QA, which consists of a graph construction module and a graph traversal module.

## 3. CyKG-RAG Framework

The objective of this research is to enhance existing LLMs with a domain-specific RAG approach that leverages symbolic representations to address cybersecurity issues pertaining to private and dynamic security-related information such as log data, security events and IT infrastructure. Additionally, we seek to utilize existing cybersecurity information including information about vulnerabilities, weaknesses and attack patterns to assist cybersecurity analysts in their daily tasks.

In this section, we present our approach for a RAG system for cybersecurity. We discuss the RAG architecture including KG construction, vector embedding of an existing cybersecurity KG, and the retrieval mechanism. Finally, we present our prototype implementation.

### 3.1. The RAG Architecture

The architecture illustrated in Figure 1 gives an overview of our RAG system that is specifically tailored for cybersecurity and leverages knowledge graphs. It can be broken down into the following components:

**(i) Knowledge Graph Construction** Our CyKG-RAG approach leverages knowledge graph representations for domain-specific knowledge related to cybersecurity, including e.g., log events from hosts, network and infrastructure descriptions, as well as cybersecurity information. This information is essential for security analysts in answering cybersecurity-related questions, and hence, will be integrated into our RAG system.

As depicted on Figure 1, the KG is constructed upon a dual type source of information i.e., (i) the Log & Event Graphs are constructed from local log data and events, which represent private organization-specific information related to cybersecurity. Furthermore, (ii) external data from publicly available cybersecurity information, such as Common Vulnerabilities and Exposures

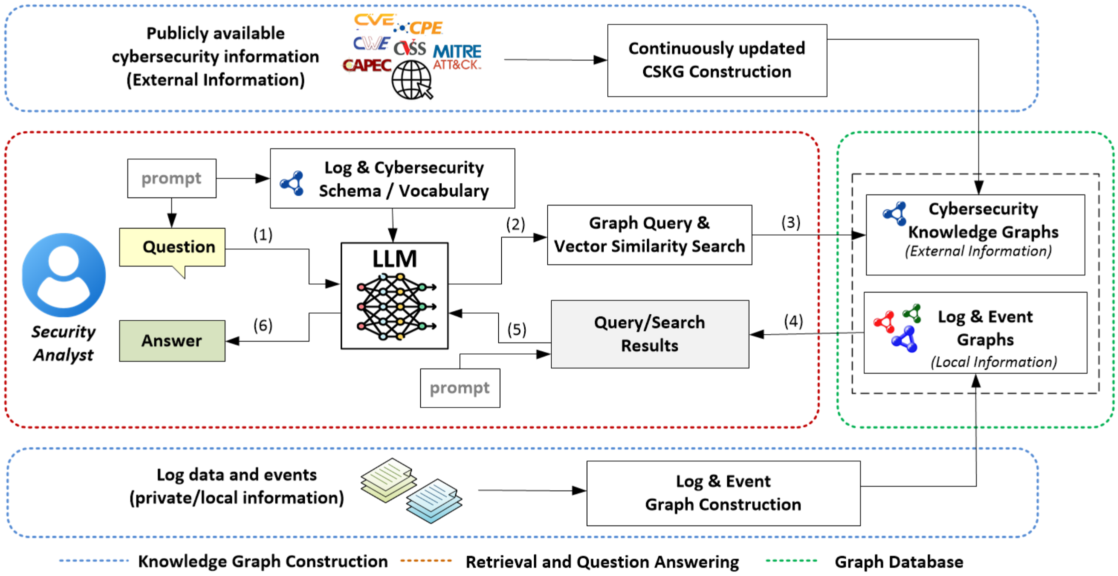


**Figure 1:** CyKG-RAG Architecture

(CVE), Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC), MITRE ATT&CK, and other threat intelligence are included. This information is continuously updated to ensure that the KG reflects the most recent landscape in cybersecurity threats and defenses. This constructed KG can then be used to contextualize local security events with external cyber threat intelligence to provide a more comprehensive understanding of potential threats.

The construction of KG can be done in two ways: (i) *LLM-Based KG Construction.* This approach transforms semi-structured and unstructured information into a KG representation by means of LLMs. (ii) *Rule-Based KG Construction.* This approach transforms structured information into a KG by using a declarative language, e.g., RML that maps the data based on an ontology/schema and transform them into a KG.

**(ii) Vector Embedding in the Existing Cybersecurity KG** The advantage of incorporating vector embeddings into a KG lies in their ability to enable flexible full-text search within unstructured elements of the KG, particularly for security-related data that is typically unstructured (e.g., log message, security events, etc). Vector embeddings in KG nodes helps to identify relevant nodes and their relationships to other entities. This capability is particularly helpful compared to traditional graph queries (e.g., SPARQL, Cypher) which may struggle to retrieve information due to mismatched and incomplete data within the KG.

The goal of vector similarity search is to locate elements (i.e., nodes) in the KG based on the similarity of their vector embeddings. These embeddings are encoded and stored as properties within the nodes. Typically, a node represents a chunk of data that is part of a larger document. In a cybersecurity context, a document could be a security log source, and a chunk might be a log line containing a message about particular events in the system.

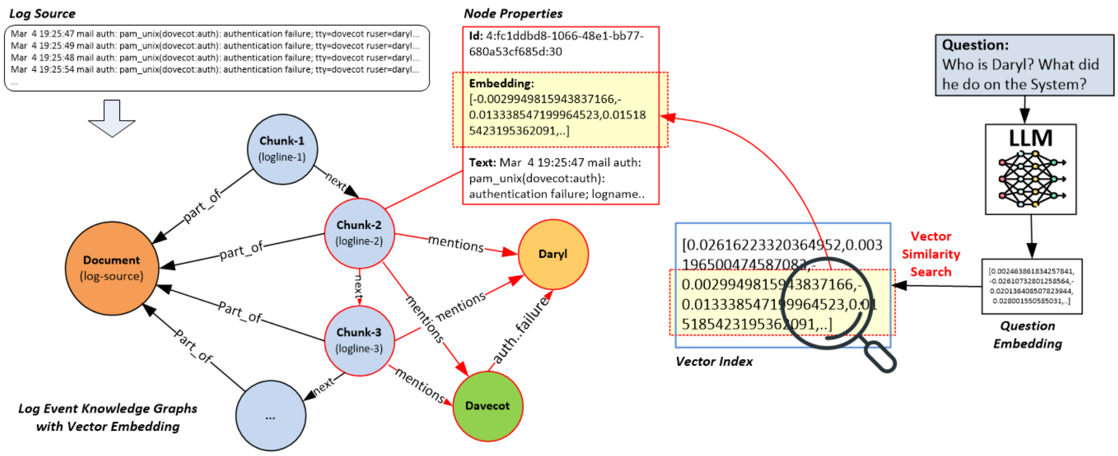


**Figure 2:** RAG question answering through vector similarity search in a KG

Figure 2 shows an example of similarity search using vector embeddings in log KG. As shown in the top-left section of the figure, the process starts with KG construction that transforms entries (e.g., *"Mar 4 19:25:47 mail auth: pam_unix(dovecot): authentication failure; logname..."*) from a log source (e.g., *Authlog*) into a log graph. The log graph is then fragmented into smaller units called *"Chunks"* (e.g., *Chunk-1, Chunk-2, Chunk-3*, etc.). These chunks are nodes in the KG and represent individual log lines associated with specific nodes that capture relevant entities (such as user *"Daryl"* and the *"Davecot"* mailserver) and their relationships (e.g., mentions, authentication failure). Each node in the KG, which represents a chunk of log data, is associated with an embedding – i.e., a numerical representation that captures the semantic meaning of the text within the chunk. These embeddings are then stored in a *Vector Index*. The embeddings allow the system to perform a vector similarity search to retrieve the most contextually relevant chunks in response to a query. For instance, the node property section shows an embedding corresponding to *Chunk-2*, which is linked to an *authentication failure* event involving *Daryl*. The chunk-wise embedding enables the system to match the query against the most relevant log lines based on their semantic similarity rather than just keyword matching.

**(iii) Retrieval and Question Answering Mechanism** As depicted in Figure 1, the retrieval mechanism of our RAG approach consists of the following steps:

- **Security Analyst's Question** ① The process starts with a *Security Analyst* that asks a cybersecurity question in natural language. This question is integrated into a prompt and sent to the LLM.
- **Graph Query & Vector (Semantic) Search** ② Based on the given question and contextual knowledge (i.e, KG schema, vocabulary), the LLM then generates a graph query — i.e., either a SPARQL query for Resource Description Framework (RDF) KGs or a Cypher query for Labelled Property Graphs (LPGs) — that is executed against the KG. Note that the generated queries may not always answer the given question – e.g., because the LLM may produce an invalid query or the query may yield an empty result. To address

this limitation, our framework combines query generation with vector (semantic) Search based on similarities between the vector representation of the question and the vector representation of the embeddings in the KG. As depicted in Figure 2, the question *"Who is Daryl? What did he do on the System?"* is first encoded into a query embedding - a vector that represents the semantic meaning of the questions - via an LLM. This query embedding is then compared against the *Vector Index* containing the embeddings of log data chunks. Through vector similarity search, the system identifies the most relevant chunks and their associate nodes in the KG. The identified nodes can then be used as a starting point to traverse other paths (e.g., via graph queries) and retrieve other information relevant to the question.

- **Cybersecurity Knowledge and Log Event Graphs** ③ The generated queries target two primary data sources:
  - *Log & Event Graphs*: These graphs represent *private/local-information* constructed from cybersecurity-related data such as logs, events and infrastructure information.
  - *Cybersecurity Knowledge Graphs*: These graphs represent *external-information* that contain broader cybersecurity knowledge constructed from publicly available data sources – such as vulnerabilities, weaknesses, attack patterns, etc.
- **Results Retrieval** ④ The results retrieved (i) from the Log & Event and Cybersecurity Knowledge Graphs based on a Graph query and (ii) via Vector (similarity) search are passed to the LLM together with the prompt ⑤.
- **Synthesizing Results** ⑥ The LLM processes and synthesizes the retrieved information to generate a final answer that is presented to the Security Analyst.

### 3.2. Prototypical Implementation

**(i) KG Construction** Our approach to KG construction is based on two fundamental techniques. The first is *rule-based construction*, which we leverage to ingest structured cybersecurity information. For this purpose, we reuse the existing CSKG[2] which employs RDF Mapping Language (RML) to map and transform structured information into a KG. We used Ontotext GraphDB[6], an Open Source graph database to store the RDF data.

Second, we utilize *LLM-based construction* to transform semi-structured and unstructured information from private or local data (e.g., log sources) into a KG. In particular, we use *Langchain LLM Graph Transformers*[7], a *Python* based tool that converts documents into graph-based formats using an LLM. In our experiments, we used ChatGPT 3.5-turbo[8] to perform these tasks.

To create vector embeddings for the constructed KGs, we leverage *OpenAIEmbedding*[9], a service provided by *OpenAI* that generates vector embeddings for text data. The generated vector embeddings are then stored in a *Neo4J vector store*[10].

[6]https://www.ontotext.com/products/graphdb/?ref=menu
[7]https://python.langchain.com/v0.1/docs/use_cases/graph/constructing/
[8]https://platform.openai.com
[9]https://python.langchain.com/v0.2/docs/integrations/text_embedding/openai/
[10]https://neo4j.io/

**(ii) Simple Chatbot Interface for Q&A** To facilitate the interaction of security analysts with the CyKG-RAG system, we developed a simple web-based chatbot system interface as shown in Figure 3. We used *Streamlit* [11], an open-source *Python* framework to rapidly build web apps. Within the resulting app, security analysts can ask security-relevant questions in natural language and get the response back together with an explanation (e.g., the actual query and its results from the KG).

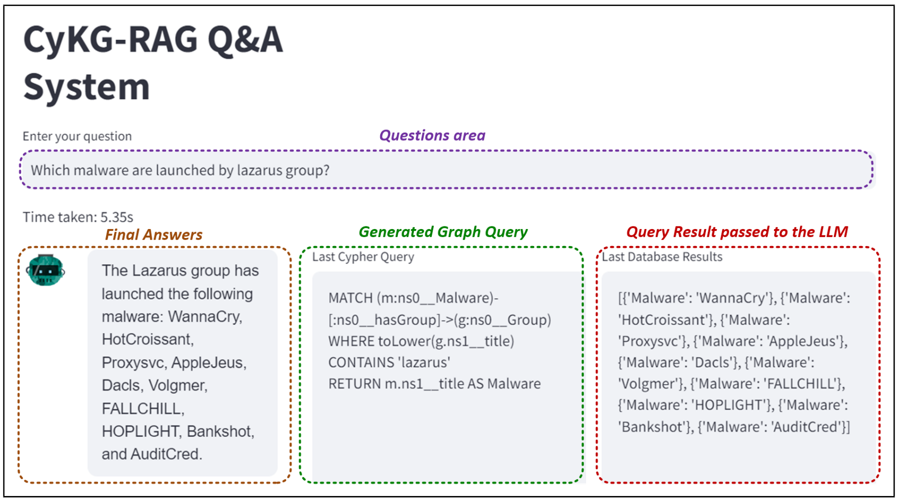


**Figure 3:** CyKG-RAG Q&A System Interface

The interface consists of several elements, (i) a Question Area where the security analyst can input queries, (ii) a Final Answers Box that displays the answers generated by the system, (iii) a Generated Graph Query section that shows the actual query created by the LLM, which could be either a SPARQL or Cypher query, and (iv) a Query Result section that provides the data passed to the LLM, including results obtained from the graph database or vector embeddings.

## 4. Experiment and Use Case Application

In this section, we describe a series of experiments that we conducted to demonstrate the feasibility of our RAG approach for cybersecurity-related question-answering and analysis tasks. We started with general cybersecurity questions related to cyber threat intelligence and vulnerability assessment using continuously updated CVE information, followed by log analysis of a specific log dataset. Different colors are used to distinguish the part of our retrieval process: (i) the blue text box represents the natural question asked by the security analyst, (ii) the green box represents the LLM's generated graph query (e.g., SPARQL, Cypher), (iii) the red box represents the query results obtained from the graph database or retrieval results from vector similarity, and (iv) the brown box represents the final answer to the question.

[11] https://streamlit.io/

**(i) General Questions on Cyber threat Intelligence** In this demonstration, we leverage our existing SEPSES CSKG[12] as a knowledge source for our RAG system. In this case, the system can be used to answer questions related to cyber threat intelligence (e.g., MITRE ATT&CK[13]) including queries about attack techniques, tactics, etc.

**Question**

Which MITRE ATT&CK techniques are used by attackers to escalate their privileges within a network?

**Generated SPARQL Query**

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX attack: <http://w3id.org/sepses/vocab/ref/attack#>

SELECT ?technique ?title WHERE {
  ?technique a attack:Technique .
  ?technique dcterms:description ?description .
  FILTER(CONTAINS(?description, "escalate")
         && CONTAINS(?description, "privileges"))
  ?technique dcterms:title ?title .
}
```

**Query Result passed to the LLM**

| | technique | title |
|---|---|---|
| 1 | res:attack/technique/boot-or-logon-initialization-scripts | "Boot or Logon Initialization Scripts" |
| 2 | res:attack/technique/token-impersonation-theft | "Token Impersonation/Theft" |
| 3 | res:attack/technique/windows-service | "Windows Service" |
| 4 | ... | "..." |

**Final Answer**

```
Attackers may use several MITRE ATT&CK techniques to escalate their
privileges within a network. These techniques include:
1. **Boot or Logon Initialization Scripts**
2. **Token Impersonation/Theft**
3. **Windows Service**
...
```

**(ii) Vulnerability Assessment over Continuously Updated Vulnerability Information** With the same knowledge source we used in use case (i), we demonstrate how our RAG system can effectively handle the dynamic retrieval of rapidly evolving cybersecurity information. For instance, the system can assess the current vulnerability of a specific component or product identified by its CVE ID. This would not be possible by an LLM alone, since it would require frequent updates (e.g., fine-tuning) whenever CVEs are released.

[12] https://w3id.org/sepses/
[13] https://attack.mitre.org/

### Question

What is the CVE identifier and its description that contains the vulnerability of C-MORE EA9 HMI? Mention three of them!

### Generated SPARQL Query

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?identifier ?description
WHERE {
  ?cve a <http://w3id.org/sepses/vocab/ref/cve#CVE> ;
       dcterms:identifier ?identifier ;
       dcterms:description ?description .
  FILTER regex(?description, "C-MORE EA9 HMI", "i")
} LIMIT 3
```

### Query result passed to the LLM

| | identifier | description |
|---|---|---|
| 1 | "CVE-2024-25138" | " In AutomationDirect C-MORE EA9 HMI, credentials used by the platform .. " |
| 2 | "CVE-2024-25137" | "In AutomationDirect C-MORE EA9 HMI there is a program that copies a .." |
| 3 | "CVE-2024-25136" | "There is a function in AutomationDirect C-MORE EA9 HMI that allows an .." |

### Final Answer

Three CVE identifiers related to vulnerabilities in AutomationDirect C-MORE EA9 HMI are CVE-2024-25138, CVE-2024-25137, and CVE-2024-25136.
CVE-2024-25138: In AutomationDirect C-MORE EA9 HMI, credentials used by the platform are stored as plain text on the device.
CVE-2024-25137: In AutomationDirect C-MORE EA9 HMI, there is a program that copies a buffer of a size controlled by the user into a limited sized buffer..
CVE-2024-25136: There is a function in AutomationDirect C-MORE EA9 HMI that allows an attacker to send a relative path in ..

**(iii) Security Log Analysis** This scenario demonstrates the use of our RAG system to facilitate question answering against private or local information (e.g., log sources). For this experiment, we used a dataset derived from the AIT dataset[41] and constructed a KG from it using an LLM. Additionally, we generated vector embeddings to enable full-text search against the constructed KG.

### Question

Who is Daryl? What did he do on the system? What time he did it?

Graph and vector data passed to the LLM

**Graph data:**

```
Dovecot - USES -> Daryl
Dovecot - AUTHENTICATES -> Daryl
Dovecot - AUTHENTICATION_FAILURE -> Daryl
Dovecot - AUTHORIZES -> Daryl
Dovecot - FAILED_AUTHENTICATION -> Daryl
```

**Vector data:**

```
text: Mar  4 19:25:49 mail auth: pam_unix(dovecot:auth): authentication failure;..
text: Mar  4 19:25:48 mail auth: pam_unix(dovecot:auth): authentication failure;..
text: Mar  4 19:28:35 mail auth: pam_unix(dovecot:auth): authentication failure;..
text: Mar  4 19:25:54 mail auth: pam_unix(dovecot:auth): authentication failure;..
```

Final Answer

Daryl is a user on the system who attempted to authenticate multiple times but failed each time. He tried to log in at 19:25:49, 19:25:48, 19:28:35, and 19:25:54.'

## 5. Conclusions and Future Work

Cybersecurity is a domain characterized by complex and fast-evolving domain knowledge as well as highly dynamic environments and information flows in various forms (such as logs). Furthermore, the relational nature of such information requires means to connect chunks of disparate information effectively and reliably. These characteristics pose a challenge for LLMs, which have shown significant potential in the cybersecurity domain, but cannot easily be retrained on updated information, have difficulties operating on factual information reliably, and are prone to effects like hallucination.

In this paper, we proposed to combine multiple retrieval-augmented generation techniques to tackle such inherent limitations and leverage LLMs in cybersecurity analysis workflows. To this end, we introduced CyKG-RAG, a RAG-based question-answering system that combines symbolic representation of security knowledge in (knowledge) graph structures with query generation and vector embedding-based semantic similarity search to retrieve relevant information in real-time and synthesize meaningful answers. Our preliminary evaluation of CyKG-RAG in example use cases ranging from general cybersecurity knowledge to answering questions on dynamic knowledge such as emerging vulnerabilities and on real-time log data show highly promising results. In our future work, we will evaluate the approach more comprehensively in a broader set of experimental settings and investigate the relative effectiveness of various RAG techniques and different approaches to combine them in different scenarios.

**Acknowledgements.** This work has been partially supported and funded by the Austrian Research Promotion Agency (FFG) via the Austrian Competence Center for Digital Production (CDP) under the contract number 881843. SBA Research (SBA-K1) is a COMET Centre within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMAW, and the federal state of Vienna. COMET is managed by FFG. This work is part of the TEAMING.AI project which receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 957402 (www.teamingai-project.eu).

## References


[1] S. Fenz, A. Ekelhart, Formalizing information security knowledge, in: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, 2009. doi:10.1145/1533057.1533084.

[2] E. Kiesling, A. Ekelhart, K. Kurniawan, F. Ekaputra, The SEPSES Knowledge Graph: An Integrated Resource for Cybersecurity, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019, volume 11779, Springer International Publishing, Cham, 2019, pp. 198–214. doi:10.1007/978-3-030-30796-7_13, series Title: Lecture Notes in Computer Science.

[3] K. Kurniawan, A. Ekelhart, E. Kiesling, G. Quirchmayr, A. M. Tjoa, Krystal: Knowledge graph-based framework for tactical attack discovery in audit data, Computers & Security 121 (2022) 102828. doi:10.1016/j.cose.2022.102828.

[4] HoloLen, Cybersecurity Knowledge Graph, 2020. URL: https://github.com/HoloLen/CyberSecurity_Knowledge_graph.

[5] I. Sarhan, M. Spruit, Open-cykg: An open cyber threat intelligence knowledge graph, Knowledge-Based Systems 233 (2021) 107524. doi:https://doi.org/10.1016/j.knosys.2021.107524.

[6] Z. Syed, A. Padia, M. Mathews, T. Finin, A. Joshi, UCO: A Unified Cybersecurity Ontology, in: Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security, 2016.

[7] A. Ekelhart, F. J. Ekaputra, E. Kiesling, The slogert framework for automated log knowledge graph construction, in: R. Verborgh, K. Hose, H. Paulheim, P.-A. Champin, M. Maleshkova, O. Corcho, P. Ristoski, M. Alam (Eds.), The Semantic Web, Springer International Publishing, Cham, 2021, pp. 631–646.

[8] R. Shah, M. Maarek, S. Stals, L. Baillie, S. C. Chan, R. Stewart, H.-W. Loidl, O. Chatzifoti, Introducing and interfacing with cybersecurity – a cards approach, 2023. URL: https://arxiv.org/abs/2307.16535. arXiv:2307.16535.

[9] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, 2024. URL: https://arxiv.org/abs/2402.06196. arXiv:2402.06196.

[10] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, 2023. URL: https://arxiv.org/abs/2303.18223. arXiv:2303.18223.

[11] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, Y. Zhang, A Survey on Large Language Model (LLM) Security and Privacy: The Good, The Bad, and The Ugly, High-Confidence Computing 4 (2024) 100211. URL: https://www.sciencedirect.com/science/article/pii/S266729522400014X. doi:10.1016/j.hcc.2024.100211.

[12] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, 2023. URL: https://arxiv.org/abs/2203.11171. arXiv:2203.11171.

[13] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu, A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023. URL: https://arxiv.org/abs/2311.05232. arXiv:2311.05232.

[14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL: https://arxiv.org/abs/2005.11401. arXiv:2005.11401.

[15] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts, 2023. URL: https://arxiv.org/abs/2307.03172. arXiv:2307.03172.

[16] A. Ekelhart, S. Fenz, T. Neubauer, Aurum: A framework for information security risk management, in: Proceedings of the 42nd Hawaii International Conference on System Sciences, 2009. doi:10.

1109/HICSS.2009.82.
[17] A. Kim, J. Luo, M. Kang, Security ontology for annotating resources, in: On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, 2005.
[18] A. Martimiano, E. d. S. Moreira, An owl-based security incident ontology, in: Proceedings of the Eighth International Protege Conference, 2005.
[19] V. Raskin, C. Hempelmann, K. Triezenberg, S. Nirenburg, Ontology in information security: A useful theoretical foundation and methodological tool, in: Proceedings of the 2001 Workshop on New Security Paradigms, 2001. doi:10.1145/508171.508180.
[20] M. Schumacher, Toward a Security Core Ontology, 2003. doi:10.1007/978-3-540-45180-8_6.
[21] J. Undercoffer, A. Joshi, J. Pinkston, Modeling computer attacks: An ontology for intrusion detection, in: Recent Advances in Intrusion Detection, 2003.
[22] A. Souag, C. Salinesi, I. Comyn-Wattiau, Ontologies for security requirements: A literature survey and classification, in: Advanced Information Systems Engineering Workshops, 2012.
[23] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, J. Goodall, Developing an ontology for cyber security knowledge graphs, 2015. doi:10.1145/2746266.2746278.
[24] K. Kurniawan, A. Ekelhart, E. Kiesling, An att&ck-kg for linking cybersecurity attacks to adversary tactics and techniques, in: The Semantic Web – ISWC 2021, 2021, p. 5. URL: https://ceur-ws.org/Vol-2980/paper363.pdf.
[25] K. Liu, F. Wang, Z. Ding, S. Liang, Z. Yu, Y. Zhou, Recent Progress of Using Knowledge Graph for Cybersecurity, Electronics 11 (2022) 2287. doi:10.3390/electronics11152287.
[26] K. Liu, F. Wang, Z. Ding, S. Liang, Z. Yu, Y. Zhou, A review of knowledge graph application scenarios in cyber security, 2022. URL: http://arxiv.org/abs/2204.04769, arXiv:2204.04769 [cs].
[27] L. F. Sikos, Cybersecurity knowledge graphs, Knowledge and Information Systems 65 (2023) 3511–3531. doi:10.1007/s10115-023-01860-3.
[28] X. Zhao, R. Jiang, Y. Han, A. Li, Z. Peng, A survey on cybersecurity knowledge graph construction, Computers & Security 136 (2024) 103524. doi:10.1016/j.cose.2023.103524.
[29] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, 2024. URL: https://arxiv.org/abs/2312.10997. arXiv:2312.10997.
[30] M. Trajanoska, R. Stojanov, D. Trajanov, Enhancing knowledge graph construction using large language models, 2023. URL: https://arxiv.org/abs/2305.04676. arXiv:2305.04676.
[31] L. Yao, J. Peng, C. Mao, Y. Luo, Exploring large language models for knowledge graph completion, 2024. URL: https://arxiv.org/abs/2308.13916. arXiv:2308.13916.
[32] T. Ban, L. Chen, X. Wang, H. Chen, From query tools to causal architects: Harnessing large language models for advanced causal discovery from data, 2023. URL: https://arxiv.org/abs/2306.16902. arXiv:2306.16902.
[33] Y. Zhang, Y. Zhang, Y. Gan, L. Yao, C. Wang, Causal graph discovery with retrieval-augmented generation based large language models, 2024. URL: https://arxiv.org/abs/2402.15301. arXiv:2402.15301.
[34] J. Baek, A. F. Aji, A. Saffari, Knowledge-augmented language model prompting for zero-shot knowledge graph question answering, 2023. URL: https://arxiv.org/abs/2306.04136. arXiv:2306.04136.
[35] X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, B. Hooi, G-retriever: Retrieval-augmented generation for textual graph understanding and question answering, 2024. URL: https://arxiv.org/abs/2402.07630. arXiv:2402.07630.
[36] J. Zhang, Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt, 2023. URL: https://arxiv.org/abs/2304.11116. arXiv:2304.11116.
[37] M. Kang, J. M. Kwak, J. Baek, S. J. Hwang, Knowledge graph-augmented language models for knowledge-grounded dialogue generation, 2023. URL: https://arxiv.org/abs/2305.18846.

arXiv:2305.18846.
[38] P. Ranade, A. Joshi, Fabula: Intelligence report generation using retrieval-augmented narrative construction, in: Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, ASONAM '23, ACM, 2023. URL: http://dx.doi.org/10.1145/3625007.3627505. doi:10.1145/3625007.3627505.
[39] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, J. Larson, From local to global: A graph rag approach to query-focused summarization, 2024. URL: https://www.microsoft.com/en-us/research/publication/from-local-to-global-a-graph-rag-approach-to-query-focused-summarization/.
[40] Y. Wang, N. Lipka, R. A. Rossi, A. Siu, R. Zhang, T. Derr, Knowledge graph prompting for multi-document question answering, 2023. URL: https://arxiv.org/abs/2308.11730. arXiv:2308.11730.
[41] M. Landauer, F. Skopik, M. Wurzenberger, W. Hotwagner, A. Rauber, Ait log data set v1.1, 2020. URL: https://doi.org/10.5281/zenodo.4264796. doi:10.5281/zenodo.4264796.