Enhancing CTI Awareness of Small and Medium Enterprises with LLMs and Knowledge Graphs

Francesco Panebianco, Nicolò Amigoni†, Gabriele Ruini†, Tommaso Paladini, Stefano Longari, Stefano Zanero, and Michele Carminati

Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Milano, Italy
name.surname@polimi.it, †name.surname@mail.polimi.it

Abstract

Effective Cyber Threat Intelligence (CTI) is crucial for proactive defense; however, the high costs and technical complexity of commercial solutions make them inaccessible to the majority of Small and Medium Enterprises (SMEs). Furthermore, public CTI is predominantly available as unstructured text, making manual processing unbearable for resource-constrained organizations. To address this gap, we present a unified, low-cost pipeline that ingests, processes, and presents actionable CTI to Small and Medium Enterprises (SMEs) through a Retrieval-Augmented Generation (RAG) conversational interface. The system automatically crawls public CTI sources and integrates them with up-to-date Common Vulnerability Exposure (CVE) repositories to ensure coverage of structured and unstructured threat data. To enhance scalability compared to existing solutions, we transform raw reports into a knowledge graph that facilitates efficient querying and contextual retrieval. The system finally combines graph-derived intelligence with organization-specific software configurations to generate tailored, accessible threat guidance from a Large Language Model (LLM). We evaluate the pipeline through large-scale ingestion of 49,747 public threat documents and the latest CVE records, alongside a survey of security experts and an unsupervised assessment. Results show a substantial improvement in response completeness and relevance compared to a baseline language model, achieving a median answer relevance score of 93.7%, while remaining interpretable and actionable for non-specialists. This work demonstrates that structured CTI extraction combined with lightweight generative interfaces can enable cost-effective cybersecurity awareness for SMEs.

1 Introduction

Attackers routinely exploit vulnerabilities in both software and hardware to conduct sophisticated cyberattacks, resulting in substantial economic losses and outages across public-sector organizations and private enterprises [8]. Current prevention methodologies rely on detailed, up-to-date information about existing threats [3], commonly referred to as Cyber Threat Intelligence (CTI). However, CTI is often available in an unstructured and decentralized form across the Internet [44]. Public feeds predominantly provide Indicators of Compromise (IoCs), digital artifacts (e.g., malware hashes or IP addresses associated with malicious infrastructures) that can be used to automatically detect intrusions. Because attackers can easily modify these artifacts to evade detection, CTI analysis has increasingly shifted toward higher-level intelligence, such as Tactics, Techniques, and Procedures (TTPs), which are general enough to render manipulation significantly more complex, although much harder to formalize [9]. Timely intelligence about threat actors and their TTPs is predominantly available in the form of CTI reports expressed in natural language. Extracting actionable knowledge from these reports typically necessitates sophisticated natural language processing (NLP) methods to transform unstructured text into machine-interpretable representations suitable for large-scale analysis. In response to this need, several commercial solutions have begun to offer conversational agents

designed to support security practitioners in constructing CTI-driven threat models for their software systems. Despite showing significant capabilities in identifying security threats, these solutions require expensive access fees that exclude Small and Medium Enterprises (SMEs), which in the European Union represent 99.8% of total businesses [42] as of 2024. For example, the annual cost of existing commercial offerings spans from several tens of thousands to several hundreds of thousands U.S. dollars [23, 36]. Many SMEs lack the resources to establish dedicated security teams. Such organizations are left exposed to cyberattacks with potentially devastating consequences. Additionally, the economic burden is paired with a technical one, as CTI information is not readily understandable to non-specialized personnel.

Large Language Models (LLMs) have proven to be a versatile tool for restructuring and presenting information in different forms, particularly when prompted to adopt a specific style via Zero-Shot Prompting [62]. While LLMs are generally prone to hallucination (i.e., confidently provide non-factual information), it is shown that this phenomenon is significantly reduced when the LLM is paired with a Retrieval Augmented Generation (RAG) engine [60]. Research works have proposed the use of RAG for providing CTI information tailored to enterprises [37, 7]; however, their traditional approach, based on vector index RAG, fails to scale with the large amount of CTI information present on the internet, especially with the typical hardware setup of a SME. To address this problem, methodologies have been proposed to construct knowledge graphs out of the extracted entities and relations [5]. Graphs can represent large quantities of information without the overhead and verbosity of natural language. These graphs can then be queried to obtain relevant subgraphs with the requested information.

In this work, we propose an end-to-end pipeline for ingesting, processing, and presenting CTI information to SMEs using an LLM chatbot augmented with a RAG engine. The system ingests public CTI reports and blog posts using crawlers and public repositories. The information is processed to construct a knowledge graph using the state-of-the-art framework LADDER [5]. Once the graph is populated, the RAG engine retrieves information from a knowledge graph containing CTI information and an up-to-date repository of CVEs. Information from these sources is retrieved using both the user's question and the organization's software setup, detailing the version of each package so that the responses are limited to relevant information.

We identify the requirements for each stage of our pipeline and evaluate the extent to which they are met. We crawl CTI reports and blog posts from 15 public sources, collecting 49747 documents as well as the complete records of CVEs from the official CVE Program. A survey is administered to security experts to evaluate the system's responses to properties that cannot be evaluated automatically. The results of the questionnaire indicate that the proposed pipeline generates responses that are significantly more comprehensive than those produced by the baseline language model, while remaining readily interpretable by non-expert personnel and directly actionable. RAG properties, such as answer relevance and faithfulness, are instead assessed in our unsupervised evaluation using the commonly used RAGAS framework [19]. In this evaluation, the system reaches an answer relevance score of 93.7%.

We summarize our contributions as follows:

- We introduce a low-budget conversational pipeline for CTI, designed for the needs of SMEs. The pipeline follows three stages: *ingestion*, *processing*, and *presentation*.
- We formalize the automation of presenting CTI information to SMEs into a three-stage problem and identify requirements for each of them.
- We evaluate the extent to which these requirements are met in our pipeline. The evaluation consists of realistic use cases, with parts of the evaluation including human feedback.

2 Primer on Natural Language Processing

Since our methodology employs Natural Language Processing (NLP) techniques, this section aims to familiarize the reader with key NLP concepts that will recur throughout the following sections. Part of the broader Speech and Language Processing field, NLP aims at developing software capable of understanding humans and interacting with them [27]. In the following paragraphs, we introduce specific NLP tasks and concepts mostly related to our approach. For a comprehensive treatment of this subject, we refer the reader to Jurafsky et al. [27].

Named Entity Recognition (NER). Typically formulated as a sequence labeling [29] or text-span detection problem [52], this NLP task involves the identification and categorization of entities mentioned in text into predefined classes. In CTI, this translates to identifying elements such as malware, threat actors, and vulnerabilities. For instance, given the sentence "Zeus Panda is a Trojan designed to steal banking information", a NER algorithm may identify "Zeus Panda" as a mention to a malware. This task can be solved by leveraging predefined dictionaries [43, 46], by training deep learning models [28, 20], fine-tuning language models (e.g., BERT) on cybersecurity corpora [43, 5], or by instructing generative language models [12].

Relation Extraction (RE). Often pipelined with NER, RE identifies semantic relations between entities that interact or are associated. Consider the sentence "The TALONITE activity group uses LookBack": a RE algorithm for CTI should identify the relation "uses" between "TALONITE" and "LookBack". As in NER, techniques for RE range from rule-based methods [46], to deep neural architectures and generative models [5, 12]. NER and RE are combined to annotate linguistic structures and build machine-interpretable representations of concepts (e.g., entities and relations) appearing in natural language texts, known as Knowledge Graphs (KGs) [5, 46].

Large Language Models (LLMs). LLMs are, at their core, deep learning-based NLP models capable of predicting the next word given a previous one [27]. From a broader perspective, LLMs can be defined as agents capable of interacting with humans by conversing in natural language. These models possess general language knowledge by previous extensive training procedures conducted on extremely large corpora of text (*pretraining*). Importantly, given their ability to predict words, many NLP tasks can be approximated as a word prediction problem known as Conditional Text Generation [27]; in other words, LLMs can solve tasks on natural language texts by being instructed with an input text known as "prompt". Notably, LLMs suffer from hallucinations [63]: their answers could contain erroneous, non-factual information.

Retrieval-Augmented Generation (RAG). This emerging paradigm combines the strengths of LLMs with external information retrieval mechanisms to enhance factual accuracy, contextual relevance, and explainability. Unlike traditional generative models that rely solely on the knowledge encoded within their parameters, RAG frameworks dynamically retrieve relevant documents or passages from an external knowledge base before or during the generation process. The retrieved content serves as contextual grounding, allowing the model to produce responses that are both informed by up-to-date evidence and reduce hallucination.

3 Motivation and Challenges

Cyber Threat Intelligence (CTI) analysis emerged as a paradigm for cyberattack prevention, providing organizations with contextual insights into adversarial behavior, enabling proactive defense and informed response strategies. A large fraction of CTI research involves studying methodologies for extracting intelligence from unstructured sources (e.g., online discus-

sions [43, 13] or natural language reports on cyber threats [5, 32]). CTI data by itself provides limited value unless it is systematically analyzed within the context of an organization's specific threat model and operational environment. This environment encompasses not only the organization's software systems but also its physical infrastructure and associated assets. The interpretation and application of CTI typically require specialized expertise, a function commonly performed by Security Operations Centers (SOCs). However, the highly specialized nature of SOC personnel and their corresponding compensation levels often render such resources inaccessible to many SMEs.

3.1 Motivation

While many approaches have been proposed to automate the ingestion, analysis, and presentation of CTI information [37, 22, 7, 59], and several powerful commercial solutions are available [23, 36], existing systems generally overlook the economic and expertise constraints faced by SMEs. Open-source solutions employing RAG engines have been proposed [37, 22, 7, 59]; however, none of them address the problem of data scalability. As more reports on similar topics are incorporated, the vector index query used for RAG contexts can rapidly exceed the available context window size, particularly when employing open-source models on resource-constrained hardware, which is typical of SMEs. In addition to financial constraints, SMEs also face significant technical limitations, as they often lack the resources to employ specialized personnel capable of interpreting and acting upon threat intelligence reports. Many commercial products exist that offer advanced automated threat monitoring solutions, with the likes of Microsoft Security Copilot [36] and Recorded Future [23]. These solutions are enterprise-grade and rely on high-cost cloud computing resources to deliver monitoring, alerting, and other security services customized to client-specific requirements. The associated costs can range from tens to hundreds of thousands of dollars annually, placing such systems beyond the financial reach of most SMEs. Current general-purpose commercial LLMs, such as ChatGPT [40], Gemini [24], and Claude [6], already provide users with the ability to generate responses that integrate web search and reasoning capabilities, often at a low cost or even for free. However, web search includes both reputable sources and low-quality or inaccurate materials. Consequently, these tools cannot be fully trusted for critical decision-making contexts. Moreover, being general-purpose models, they are not readily adaptable to a specific scenario. As such, we require a tailored solution for the CTI domain.

Addressing this gap in CTI research, our work proposes a cost-effective framework designed to deliver clear, actionable, and contextually relevant CTI tailored to the specific needs of SMEs.

3.2 Challenges

In this section, we formalize the key challenges associated with designing a threat monitoring platform tailored to the needs of SMEs. These challenges are articulated as a set of requirements that must be satisfied for the proposed approach to effectively achieve the objectives of this study. We identify three main stages for such a pipeline: (A) *information ingestion*, (B) *information processing*, and (C) *information presentation*. We now enumerate the requirements associated with each stage.

- **(A) Ingestion.** The ingestion stage of a threat monitoring pipeline should collect information from publicly available intelligence sources to construct a comprehensive knowledge base of global CTI data. This stage has one primary requirement: (1A) *Information Completeness*, which means that the knowledge base must contain information that is

sufficiently comprehensive for its intended users. Such information comes in two forms, structured and unstructured information. While the former is easier to parse, ignoring the latter would exclude a large corpus of information.

- **(B) Processing.** The processing stage of a threat monitoring pipeline should autonomously transform and organize incoming CTI data to facilitate efficient storage and enable optimized downstream operations, such as querying and analysis. Raw structured information collected during the ingestion stage is directly usable by the presentation stage, as it can be stored and retrieved with negligible overhead. In contrast, the unstructured textual information obtained from web sources is substantially more verbose than necessary for efficient storage and retrieval of the relevant content. We can formalize the requirements of the information processing stage as follows. (1B) *Information Factuality*, which specifies that the processed information must preserve its original truth content, and (2B) *Scalability*, which requires that the system be able to continuously incorporate additional information without burdening the retrieval stage with excessive contextual overhead.

- **(C) Presentation.** The presentation stage of a threat monitoring pipeline should support user interaction through natural language queries. Likewise, the system is expected to generate natural language responses consistent with the ingested global CTI corpus. However, because not all intelligence is contextually relevant, the system should filter this knowledge to highlight elements most pertinent to the user's organizational profile. This stage is subject to the following requirements: (1C) *Clarity*, meaning information should be conveyed in plain, accessible language, suitable for personnel with limited technical expertise, (2C) *Answer Relevance*, which requires that responses be tailored to the specific circumstances of the organization using the system, and (3C) *Actionability*, which mandates that the system's response be formulated such that it is directly actionable, suggesting straightforward translation into practical measures for threat mitigation.

4 Methodology

We propose a conversational system designed to ingest, process, and present CTI information in a manner accessible to personnel with limited technical expertise within small and medium-sized enterprises. Figure 1 makes an overview of our approach, highlighting to what stage of the pipeline each component belongs. We now examine each of the three stages: *Ingestion*, *Processing*, and *Presentation*, detailing their respective requirements (as described in Section 3.2) and the solution by which those requirements are satisfied.

4.1 Information Ingestion

The ingestion stage of our pipeline is responsible for autonomously acquiring information from designated reputable sources. Some of this information comes in structured format, for example Common Vulnerability Exposure (CVE) reports. The majority of this information however, is only publically available in unstructured form (e.g. CTI reports, blogposts, hacking forums). To retrieve unstructured information, the ingestion stage deploys autonomous crawlers to collect CTI data from these sources. On the other hand, structured information of CVE data is simply retrieved from up-to-date official repositories. This information is particularly valuable because it can be readily correlated with an organization's infrastructure to yield an accurate enumeration of unresolved vulnerabilities within its systems. By ingesting both structured and

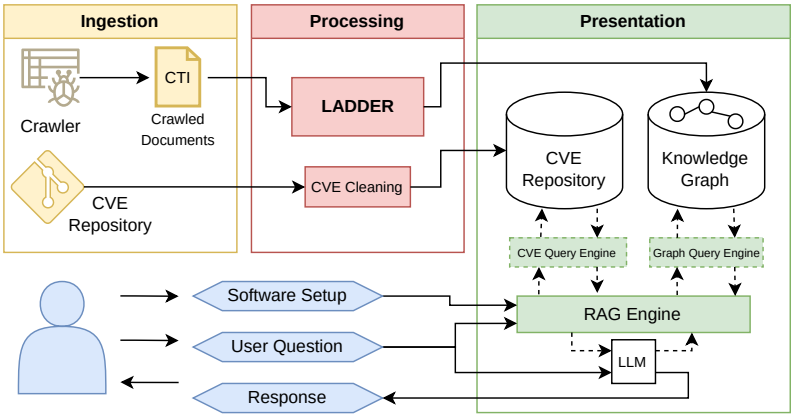


Figure 1: Overview of our approach.

unstructured CTI data, the first stage of our pipeline satisfies requirement (1A) Information Completeness.

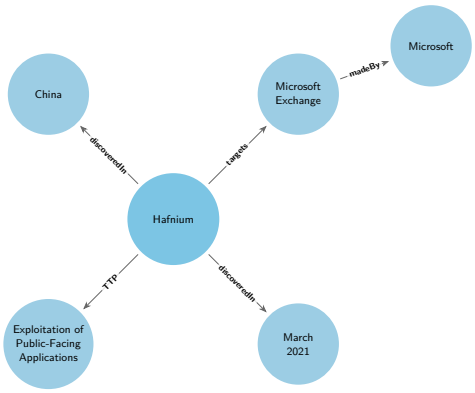
4.2 Information Processing

The processing stage of our pipeline ingests raw information collected from diverse CTI sources and stores it in a database for subsequent querying. The latest CVE data is retrieved from the repository and cleaned to remove rejected or duplicated entries and parse the standardized fields. Versions of softwares are expected as a typical dot-separated sequence of major version, minor version, and optionally a build number. The textual description of the CVE is kept as is and will be indexed in a vector database to be used directly by the RAG engine. While structured data obtained from CVE reports can be directly integrated with minimal processing, unstructured data is subjected to information extraction procedures to identify relevant entities and relationships, which are then represented in the form of a knowledge graph. The knowledge graph serves as a minimal representation of the factual content extracted from the original data, effectively eliminating the redundancy and ambiguity inherent in natural language text. Consequently, the adoption of a knowledge graph enables efficient data organization and retrieval, thereby addressing requirement (2B) Scalability in the system design.

We instead address requirement (1B) Information Factuality through the use of the state-of-the-art framework LADDER [5], which processes CTI reports with encoder-based transformers to extract TTPs, Advanced Persistent Threats (APTs) and other relevant information using NER and RE (introduced in Section 2). The entities and relationships that will form the knowledge graph are extracted using traditional encoder models (e.g. BERT [17, 34, 14]). In contrast to generative models like LLMs, classification models like BERT [17] offer the advantage of extracting only the exact tokens that were present in the text, thereby avoiding the generation of unsourced content and satisfying requirement (1B).

4.3 Information Presentation

The presentation stage of the pipeline is the component that is deployed to allow continuous user interaction with the system. It allows a user to ask a question and to follow-up with




```
Microsoft Exchange[Application](
    madeBy:(
        Microsoft[Organization]
    )
    isTargetedBy:(
        Hafnium[ThreatActor](
            discoveredIn:(
                China[Location],
                March 2021[Time]
            )
            ttp:(
                Exploitation of Public-Facing
                ↪ Application[TTP]
            )
        )
    )
)
```

Figure 2: Example of queried subgraph (left) and its textual LLM context representation (right).

additional doubts after the model has provided an answer. Due to the recent advances in language modeling, a natural candidate for this step is an LLM. Language Models can be prompted to restructure and adapt complex information into various formats and for diverse audiences [35]. Owing to their strong generalization capabilities, LLMs can reliably follow such requirement-focused prompts [62]. In our zero-shot prompting strategy, we guide the LLM into providing a clear response, targeted to an untrained user working in a SME. The LLM is asked to limit the usage of technical vocabulary, and to explain its meaining when the use is deemed necessary. The choice of using an LLM to execute this task thus fulfills requirement (1C) Clarity. We address requirement (3C) Actionability in a similar manner. In our prompt engineering strategy, we instruct the LLM to always include key takeaways at the end of their response. This paragraph has to contain clear and easy-to-follow action points that can meaningfully mitigate the described threats.

By itself, the LLM cannot be expected to provide factual answers, as such our pipeline integrates a RAG engine into the LLM conversation loop. The RAG engine queries the knowledge graph using the user question. Entities are retrieved by their semantic similarity to the request. The similarity is thresholded to discard irrelevant results. From the selected entities, the relations are traversed up to k jumps (tunable parameter). Relations are also considered backwards from the connected entity to the source to increase informational coverage. The resulting subgraph is encoded as a string. For reference, the example from Figure 2 is represented as would be returned by the RAG engine. After the extraction, LADDER provides us with entities that represent specific concepts such as malware families, dates, organizations, individuals, operating systems, to name a few. Simultaneously, it links these entities through relations such as “discoveredIn”, “targets”, and “isA”. Figure 2 (left) illustrates an example of a knowledge graph that could be produced by LADDER. The RAG engine also retrieves CVE entries by matching them to an additional list provided by the user describing the software setup of the organization. Each entry in the list contains the name of the software and its version. Only CVEs affecting the provided software versions are retrieved by the engine. The usage of the described RAG pipeline addresses requirement (2C) Answer Relevance, allowing the generated answer to contain relevant CTI information for the specific query and organizational characteristics of the user. The usage of a RAG backend also strengthens requirement (1B) Information Factuality, since the usage of RAG has been empirically demonstrated to substantially reduce LLM hallucinations when supplied with factual information [50].

After receiving an answer, the user may provide a follow-up message. In such cases, the pipeline is executed again, with prior messages incorporated into the context to enable the LLM to generate a coherent continuation of the dialogue. When using models with limited context windows, extended conversations may exceed the maximum allowable input length. In these situations, earlier messages are summarized before being reintroduced into the context, following common practice in transformer-based conversational systems [60, 30].

5 Experimental Evaluation

In this section, we go through the validation steps for our pipeline, from the setup of our experimental setting to the discussion of results.

Datasets. For a comprehensive evaluation of our pipeline, three datasets are required. The first is needed to extract real-world CTI information to populate the knowledge graph. The second consists of CVE entries to be served when matched with vulnerable versions of the user's software. The third contains usage scenarios designed to assess the quality and relevance of the overall system's responses. To assemble the first dataset, we replicate the crawling procedure described by Paladini et al. [43], collecting up-to-date blogposts, reports, and papers from 15 sources. This is one fewer than in the original study, as one source has since employed non-trivial anti-crawling protections that prevented reliable data acquisition. In total, we collected 49747 documents, whose processing resulted in a graph with 60885 entities and 221933 relationships. For a complete characterization of the knowledge graph constructed in our experimental evaluation, read Appendix A. The second database is downloaded from a cache of the official up-to-date CVE List from the CVEProject on Github [1]. CVE information in this repository is organized in a json file for each entry. We process each file to populate a SQLite dataset for easier querying. Finally, the third dataset consists of a synthetic JSON collection comprising 103 usage scenarios for the system. Each scenario includes both a software-setup field (a list of software names and corresponding versions) and a user query, which may or may not pertain to the software installed on the user's system. This enables an evaluation of the pipeline's capacity to reason over both CVE information and the CTI subgraph retrieved during inference, which we will employ in our unsupervised evaluation. For generating these scenarios, we use the state-of-the-art model Gemini 2.5 Pro [15].

Hardware and Software Specifications. All experiments are run on a machine with a Intel(R) Xeon(R) Gold 6418H 2.100GHz, 64 GB of RAM, and a Nvidia L40 (48 GB VRAM) on Ubuntu 24.04.3 LTS. The machine runs an Ollama [39] server for LLM inference, a Neo4j [38] server for the knowledge graph, and a Jupyter server for the experiment scripts. We use LangChain, LangGraph and the OpenAI library for inference on the Ollama server.

5.1 Human Evaluation

Properties like (1C) Clarity and (3C) Actionability of a textual response are not straightforward to evaluate in an automatic experimental suite. As such, we rely on the judgment of human evaluators for this purpose. We conducted a survey with a pool of 15 security-trained volunteers, reporting the system's responses to three realistic use cases. These subjects include both Ph.D. Students and professionals in the field of Information Security. For this assessment, we compare our system with the base LLM model that does not have access to a knowledge graph nor a CVE database. Having to evaluate information-agnostic aspects of a response, we

[1] https://github.com/CVEProject/cvelistV5/

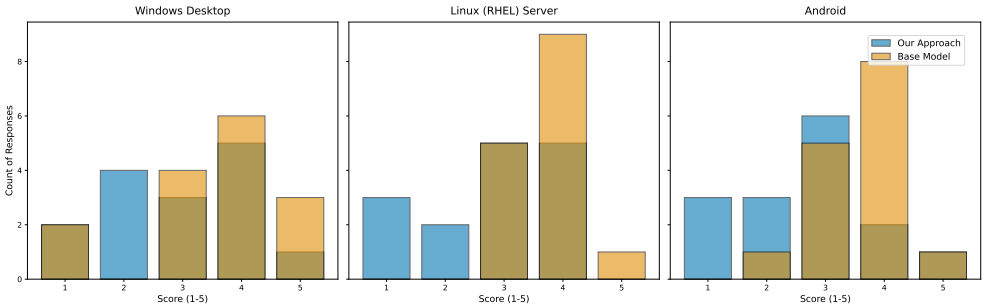


Figure 3: Scores provided by respondents on clarity of threats and prevention.

ignore the factuality of the answers for the purposes of this evaluation. Accordingly, our term of comparison will not be guaranteed to satisfy the requirement (1B) Factuality. The base model selected for both our pipeline and the baseline evaluation is Gemma 3 27.4B [58] (Q4_K_M quantized), a widely adopted open-source LLM that can be deployed on consumer-grade hardware and is therefore particularly accessible to small enterprises.

We briefly describe the three realistic use cases evaluated in the survey.

- Windows Desktop User
  This use-case includes software that is commonly present on a desktop computer. Naturally, the operating system for this use case is Windows, being the one with the largest market share at the time of writing (66.25%) [54].

- Linux Enterprise Server (RHEL)
  This use case includes software that is commonly present on an average company server hosting websites and databases. According to recent statistics on operating system market share in the server domain, Linux is the most popular choice (58.3%) among web servers for which the operating system is known [57].

- Mobile Threats (Android)
  This use case does not focus on specific software installed on a company's system. Instead, it entails a request for information on potential threats targeting an employee's Android device. Similarly to the previous use cases, Android was chosen as the most prevalent Mobile operating system (72.6%) [55]. In this case, the LLM's response is primarily conditioned by the knowledge graph information rather than the CVE database.

The survey included three questions per scenario. The questions are the same for all scenarios, except for the second question, which is not asked in the third scenario.

- **Q1.** How much do you think a non-expert user will be able to understand what could happen and what can be done to prevent exploitation?

- **Q2.** What do you think of the length of this response? (1 is too short, 5 is too long).

- **Q3.** Give a score on the completeness of this response.

In the Mobile Threats (Android) scenario, question (2) was not included, as the model's responses in this scenario are of comparable length.

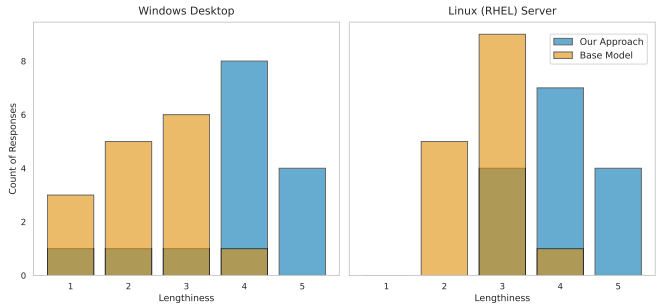


Figure 4: Scores provided by respondents on the length of the model's answer.

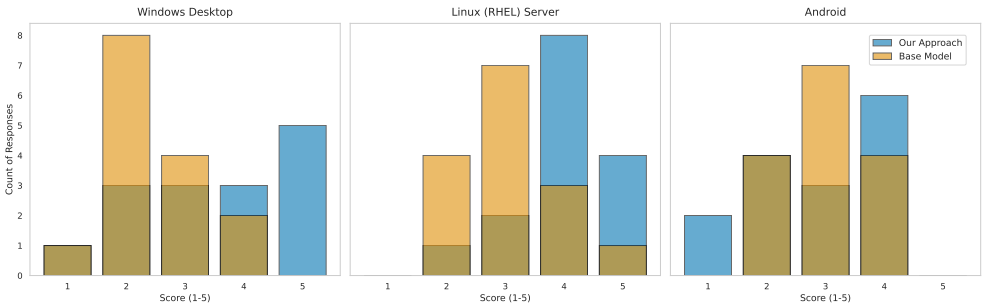


Figure 5: Scores provided by respondents on completeness of the model's answer.

Answer Clarity (1C) and Actionability (3C). Figure 3 shows the distribution of responses to the first question across all three scenarios, whereas Figure 4 shows the distribution of responses to the second question across the first two scenarios. These two questions both pertain to the clarity of the response. The distribution across the two models is similar and close to the center of the range. We can see that the base model is considered slightly more understandable than the model using the knowledge graph and CVE database. This also correlates with the length of the responses, which are shorter in the base model. The reason for this mismatch is clear: the base model does not have much pertinent information to present to the user, and thus falls back on its training data. Figure 3 also covers the property of actionability, as it asks the respondent whether a non-expert would "be able to understand what could happen and *what can be done to prevent exploitation*". In all responses, both our approach and the baseline consistently include a closing paragraph with actionable bullet points, guiding the user on mitigating the discussed threats.

Information Completeness (1A). Figure 5 presents the distribution of responses to the third question across all three scenarios. This aspect appears to be a particular strength of our pipeline, as the scores assigned by respondents are consistently higher than those of the base model in all cases. These results indicate that the experts participating in the survey judged the model's output to be sufficiently comprehensive for explanatory purposes when addressing non-expert personnel. The reason for this is readily apparent, since the base model is only capable of generic remarks, and does not have access to up-to-date CVE entries nor CTI reports.

5.2 Unsupervised Evaluation

RAGAS [19] is a framework including both unsupervised and supervised metrics for the evaluation of RAG pipelines. Metrics in the first category use a judge model to produce scores on the target of the evaluation (e.g., answer relevancy, faithfulness to the retrieved context, context recall). Metrics in the second category require a ground truth and may rely on a judge model or embedding similarity (e.g., answer semantic similarity, answer correctness). While heavily reliant on the quality of the judge model, RAGAS is extensively used by state-of-the-art RAG-based approaches [7, 37]. We focus on the two most significant unsupervised metrics, that is *answer relevancy* and *faithfulness*. The former, in particular, serves as the metric for assessing requirement 2C (see Section 4.3).

For the LLMs employed in our evaluation, we select Gemma 3 27.4B [58] (Q4_K_M quantized) as the answer generation model and GPT-4.1 [41] (gpt-4.1-2025-04-14) as the evaluation model. Gemma 3 was selected because it represents one of the highest-performing open-source models within its parameter range and is readily accessible to small and medium-sized enterprises. GPT-4.1, a larger commercial model, was chosen to provide a more reliable, robust, and unbiased assessment of the outputs produced by the generation model.

Results. The evaluation yields a median answer relevance of 0.937. Among the 103 scenarios, only 32 samples received a low relevancy score. This occurred because the model sometimes provided verbose answers or asked multiple follow-up questions instead of giving a direct, concise response to the main query. In these cases, manual verification shows that the answers are usually satisfactory despite their verbosity. Since the RAGAS evaluation is restrictive in this regard, computed relevancy for these samples yields lower scores. Faithfulness exhibits a median value close to 0.5, with a relatively uniform distribution spanning the interval $[0, 1]$. This variability reflects the dependence of faithfulness on the information content present in the provided context. Questions achieving the highest faithfulness are those for which the model can derive a complete answer directly from the supplied context. For example, in the query "What vulnerabilities could expose source code to unauthorized access?", the system accurately retrieves and reports relevant CVE information corresponding to vulnerabilities that satisfy the request. Conversely, questions with the lowest faithfulness are those for which the model relies predominantly on its parametric (training) knowledge rather than the retrieved context. An illustrative case is: "How can malware spread through file-sharing applications?". In this particular case, the model was unable to retrieve any relevant contextual information from the knowledge graph and consequently generated a response based solely on its general training knowledge.

5.3 Discussion

In Section 4, we presented our approach, which consists of three distinct stages, each characterized by specific requirements intended to address the challenges outlined in Section 3. Our experimental evaluation omits two properties of the processing stage. Requirement (1B) Information Factuality is excluded from evaluation because it is assumed to hold due to: (1) the reliability of the CTI information sources, which include reputable publishers of threat intelligence reports and blog posts; and (2) the use of extraction technologies (e.g., BERT [17]), which operate solely on existing tokens within the text and are therefore incapable of generating fabricated content ("hallucinations"). Requirement (2B) Scalability is likewise omitted, as it is considered trivially satisfied: raw textual data inherently contains the redundancy of natural language, which knowledge graphs effectively eliminate. Thus, retrieved context for a vector index database would be larger than that of a graph representing the same concepts. All

remaining properties are included in the evaluation. Table 1 highlights how each property was evaluated and the measurable results we obtained.

Still, the motivation of our work has a strong financial component. The pipeline we developed is designed to be sustainable for a SME in terms of resource investment. While we have already argued how our choices support this goal, we also include an analysis of the prospective costs of our pipeline. To prove our point, we run our model on a consumer-grade laptop with a Intel® Core™Ultra 7 Processor 155H and a Nvidia GeForce RTX 4050 Mobile. Across multiple experimental runs, the generation time ranged between 154 and 238 seconds. Although this duration is relatively long for a general-purpose model, it remains acceptable for a sparsely utilized system performing an analytical task. The process exhibited a memory footprint below 20 GB of RAM and approximately 5.64 GB of VRAM. The complete knowledge graph, stored in Neo4J, required only 1.16 GB of disk space, whereas the model itself occupied roughly 17 GB. Power consumption during operation averaged 30 W, with approximately 15 W attributed to GPU utilization. Electricity prices vary geographically and temporally; however, according to the Italian National Confederation of Artisans and Small and Medium Enterprises (CNA) [16], the national average price for Italian SMEs in June 2025 was approximately 118 €/MWh. Under these assumptions, the estimated electricity cost per query is between €0.000151 and €0.000234.

While enterprise threat intelligence platforms are generally sold via private offers negotiated on a customer basis, we can make some estimates on some existing solutions that publically disclose pricing. For example Recorded Future Threat Intelligence [23] contract is available on Amazon Web Services (AWS) [47] for $281,250.00 USD (36 month contract). On the other hand, Microsoft Security Copilot is estimated to cost between $4 and $6 USD per Security Compute Unit (SCU) per hour. Considering that the system requires a minimum of one SCU operating continuously (24 hours per day, 7 days per week), this corresponds to a baseline annual cost of approximately $35,040 USD. While such enterprise-grade services provide extensive functionality, such as continuous threat monitoring and proactive attack mitigation, these capabilities extend well beyond the operational and financial reach of most SMEs.

6 Related Work

CTI Mining. CTI mining approaches typically aim at converting unstructured information into structured representations known as Knowledge Graphs (KGs). Such a structure enables several key CTI tasks, such as structured knowledge retrieval [5, 51], or threat hunting activities [5, 46]. KG development is mostly based on a pipeline of two NLP tasks [5, 45]: NER (see Section 2), to identify text spans mentioning cybersecurity entities, and RE (see Section 2), to identify the semantic relationships between such entities (e.g., threat actor *is-author-of* a specific malware). Popular examples are *Extractor* [46] and *LADDER* [5]. *Extractor* [46] focuses on the extraction of IoCs and links them via relations representing system calls to reconstruct the behavior of a specific threat. Conversely, *LADDER* [5] connects IoCs, as well as TTPs, with cybersecurity entities (e.g., threat actors, malware), to structure CTI into a KG and support several threat hunting activities. Other examples include *TRICTI* [32], which associates IoCs to the specific phases of the attack *killchain*, CyberEntRel [2], which proposes a deep learning approach for jointly solving NER and RE, or CSKG4APT [45], which proposes a KG structure that supports conversion to the Diamond Model [11] of cybersecurity.

More recent efforts proposed to solve this task via Conditional Text Generation with Large Language Models (LLMs) [21, 26, 33, 51], i.e., by instructing a LLM to identify such entities and relations from text with a specific input *prompt*. *aCTIon* [51] performs NER and RE with

Table 1: Summary table of the evaluation of each requirement.

Pipeline Stage	**Req.**	**Evaluation Findings**
Information Ingestion	(1A)	Human Survey (Q3). Responses of our pipeline are rated as significantly more complete than the base model's.
Information Processing	(1B)	Justified by choice of model architecture. BERT-like models only extract text from the input and do not hallucinate.
	(2B)	Trivially verified. The usage of a knowledge graph significantly reduces the textual overhead of the knowledge base.
Information Presentation	(1C)	Human Survey (Q1). Good performance. Base model is slightly better rated, but correlates with shorter responses.
	(2C)	Unsupervised RAGAS evaluation. Median 93.7% answer relevancy score.
	(3C)	Human Survey (Q1). Good performance. Base model is slightly better rated, but both always include short bullet points of key takeaways.

prompt-instructed LLMs, to build STIX representations of narrative CTI reports. Similarly, *CTINEXUS* [12] performs the task using LLMs; however, they also propose a structure capable of adapting to diverse ontology schemas (e.g., MalONT), as well as a methodology for identifying novel links in the final KG structure. Other works [26, 49] propose agentic frameworks [56], or fine-tuned LLMs to build KGs. *CTIKG* [26] uses a dual-memory design and multi-temperature reasoning to mitigate hallucinations. MAD-CTI [49], instead, uses multi-agents to analyze dark web data for emerging cyberthreats. Lastly, *LLM-TIKG* [25] is a Llama2 fine-tuned on synthetic LLM-generated data to support the extraction of entities and relations from CTI reports.

Considering the vastity of methodologies proposed, and evaluation settings that hinder comparison, we refer the reader to the systematization of Büchel et al. [10] to gain a complete understanding of CTI extraction techniques. Our approach relies on *LADDER* [5], which, to the best of our knowledge, is the only fully publicly available methodology capable of extracting numerous CTI entities, among which IoCs, TTPs, and connect them with adversaries and tools in a Knowledge Graph. Furthermore, LADDER employs encoder models like BERT [17], which enables us to process large quantities of documents with state-of-the-art performance and low resource consumption.

Conversational Agents. Similarly to our work, other CTI research regards the study of frameworks that support CTI operations and natural language interactions with practitioners [22, 48, 53]. *LocalIntel* [37] integrates *global* and *local* cyber-knowledge to generate contextualized organizational threat intelligence. Their approach combines open-source and organizational data sources to improve the relevance and specificity of threat information available to analysts. IntellBot [7] is an LLM-based chatbot that leverages retrieval-augmented generation (RAG) to deliver cybersecurity knowledge. Tellache et al. [59] present an approach for automating incident response, by using RAG to incorporate CTI information into user queries. Notably, Alam et al. [4] introduce CTIBench, a benchmark suite for evaluating LLMs in cyber threat intelligence tasks. The benchmark encompasses multiple datasets and subtasks designed to measure domain knowledge, reasoning ability, and generalization of LLMs in CTI scenarios.

As the field advances, many approaches are continuously proposed; thus, we refer the reader to the work of Xu et al. [61], which survey over 180 papers to map trends in model architectures, adaptation strategies, and emerging use cases such as autonomous security agents and proactive defense mechanisms. With respect to existing solutions, our approach is tailored to the requirements of SMEs, constraining and directing the methodological choices towards affordable and efficiently scalable solutions.

Retrieval Augmented Generation (RAG). Several works introduce RAG techniques to improve the quality of LLMs' answers and reduce the likelihood of hallucination (see Section 2). In *LocalIntel* [37], RAG is used to combine *global* knowledge from public threat repositories with *local* knowledge from private databases. In this way, LocalIntel contextualizes such global information for specific infrastructures of an organization. Agrawal et al. propose CyberQ [1], an innovative model designed for educational purposes. The novelty introduced by this work lies in the fact that the same entity is queried three times with different and increasingly specific prompts to enhance ChatGPT's knowledge and context. Lekssays et al. [31] propose a retrieval-augmented framework targeted for adversarial technique annotation in CTI text. The system enhances retrieval precision via zero-shot re-ranking and domain-specific alignment, improving the reliability of LLM-generated annotations. Our retrieval module for CTI reports is instead inspired by *GraphRAG* [18], which extends the RAG paradigm by incorporating graph-structured knowledge representations. Graph RAG uses LLMs to create a knowledge graph, according to two phases: first, it extracts entities and relations from source documents, and second, it creates summaries for communities of related elements. The summaries from such communities are integrated to produce the final answer to a specific question. While GraphRAG effectively mitigates the retrieval overhead associated with individual queries, our approach further reduces it by directly retrieving subgraph information in a structured form. In this manner, the system mitigates rapid exhaustion of the context window (limited on low-end hardware), optimizing token utilization.

7 Limitations and Future Works

The performance of our approach is limited by several factors. As explained in Section 4, we rely on LADDER [5] for the Information Processing module. To a certain degree, LADDER's NER and RE models introduce noise in our pipeline, reducing the validity of the information and facts presented to the practitioner. In few cases the intelligence produced contains noise (entities might be extracted with the adjacent punctuation, e.g. "OneNote"). We attribute these errors to the intrinsic limitations of the BERT-based models and their training data. Note that LADDER [5] figures among the few works that provide complete, open access to models and data, making it fully reproducible [10]. The impact of its performances on our practical CTI application leads us to think that future works should focus on improving the performance of CTI mining tools and knowledge representation methodologies. For example, augmenting knowledge graphs with logical constraints could filter out impossible relations, improving the reliability of the extraction. Additionally, the performance of our approach is limited by the specific configuration adopted for our Graph-querying RAG and our LLM. These represent cost-performance trade-offs that reduce resource consumption and align the requirements with the limited amount of resources typically possessed by SMEs. We acknowledge that, although our dataset comprises 49,747 reports, comparable in scale to datasets used in related studies, the information contained in the Knowledge Graph may be incomplete. Specifically, it currently represents data associated with a limited subset of cyberattacks, and more recent threat intelligence may be underrepresented. Nevertheless, the collected sources provide a representative

dataset for the requirements assessed in our experimental evaluation. Future iterations of the system could enhance coverage by extending the crawling pipeline to incorporate a broader range of publicly available sources. Additionally, the current implementation does not process IoCs embedded within the crawled reports, which limits the system's ability to identify specific threats during the presentation phase. However, the primary objective of our approach is preventive CTI rather than incident response or intrusion detection. Future work may investigate methods for integrating deterministic IoC matching with the inherently stochastic characteristics of LLMs.

8 Conclusions

As the cyberthreat landscape evolves, all organizations face major challenges, and the ability to analyze, understand, and mitigate such threats is a necessity. In this paper, we explored how Large Language Models (LLMs) can support Cyber Threat Intelligence (CTI) by presenting structured intelligence collected in a Knowledge Graph (KG) through natural language conversation and support non-expert practitioners. Our goal was to provide an accessible solution tailored for Small and Medium Enterprises (SMEs), which do not have the financial and technical resources to address these needs. We proposed a framework structured in three main stages. The first stage was responsible for collecting CTI information from public sources using dedicated crawlers and repositories of structured Common Vulnerability Exposure (CVE) reports. The second module processed raw information from the sources; for this task, we relied on LADDER [5], a well-recognized state-of-the-art approach based on the BERT [17] language model. The final stage presented the information using an LLM with Retrieval Augmented Generation (RAG) to provide contextual answers by integrating information from local sources, software and versions used by the user, and global sources encoded in the KG produced by LADDER. In our experiments, we evaluated our approach with a human survey and an unsupervised evaluation based on RAGAS [19], a commonly employed framework for evaluating RAG systems. Respondents of our survey have deemed the answers of the model significantly more complete than the baseline, consistently producing clear and actionable responses. Our unsupervised evaluation yields a median answer relevancy of 93.7%, measured across numerous and diverse use-cases. Future work could improve the ingestion stage by acquiring information from a more comprehensive pool of sources, as well as ensuring real-time update of the KG content. The integration of LLMs with structured knowledge poses as a promising direction for Cyber Threat Intelligence.

Acknowledgements

This work was co-funded by the Google.org Impact Challenge - Tech for Social Good Research Grant (Tides Foundation) and by the Ministero delle Imprese e del Made in Italy. The research was conducted within the framework of the project "WASABI - Web Analysis System with Artificial intelligence on Big data", established under the Innovation Agreement signed on December 4, 2024.

References

[1] Garima Agrawal, Kuntal Pal, Yuli Deng, Huan Liu, and Ying-Chih Chen. Cyberq: Generating questions and answers for cybersecurity education using knowledge graph-augmented llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23164–23172, 2024.

[2] Kashan Ahmed, Syed Khaldoon Khurshid, and Sadaf Hina. CyberEntRel: Joint extraction of cyber entities and relations using deep learning. *Computers & Security*, 136:103579, January 2024.

[3] Scott Ainslie, Dean Thompson, Sean Maynard, and Atif Ahmad. Cyber-threat intelligence for security decision-making: A review and research agenda for practice. *Computers & Security*, 132:103352, 2023.

[4] Md Tanvirul Alam, Dipkamal Bhusal, Le Nguyen, and Nidhi Rastogi. Ctibench: A benchmark for evaluating llms in cyber threat intelligence. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 50805–50825. Curran Associates, Inc., 2024.

[5] Md Tanvirul Alam, Dipkamal Bhusal, Youngja Park, and Nidhi Rastogi. Looking beyond iocs: Automatically extracting attack patterns from external cti. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, pages 92–108, 2023.

[6] Anthropic. Claude, 2025. Accessed: 2025-11-24.

[7] Dincy R. Arikkat, Abhinav M., Navya Binu, Parvathi M., Navya Biju, K. S. Arunima, Vinod P, Rafidha Rehiman K. A., and Mauro Conti. Intellbot: Retrieval augmented llm chatbot for cyber threat knowledge delivery. In *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 644–651, 2024.

[8] Ashish Arora and Rahul Telang. Economics of software vulnerability disclosure. *IEEE security & privacy*, 3(1):20–25, 2005.

[9] David J. Bianco. The pyramid of pain, 2013. Accessed: 2025-11-27.

[10] Marvin Büchel, Tommaso Paladini, Stefano Longari, Michele Carminati, Stefano Zanero, Hodaya Binyamini, Gal Engelberg, Dan Klein, Giancarlo Guizzardi, Marco Caselli, et al. {SoK}: Automated {TTP} extraction from {CTI} reports–are we there yet? In *34th USENIX Security Symposium (USENIX Security 25)*, pages 4621–4641, 2025.

[11] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. The diamond model of intrusion analysis. 2013.

[12] Yutong Cheng, Osama Bajaber, Saimon Amanuel Tsegai, Dawn Song, and Peng Gao. Ctinexus: leveraging optimized llm in-context learning for constructing cybersecurity knowledge graphs under data scarcity. *arXiv e-prints*, pages arXiv–2410, 2024.

[13] Vanessa Clairoux-Trepanier, Isa-May Beauchamp, Estelle Ruellan, Masarah Paquet-Clouston, Serge-Olivier Paquette, and Eric Clay. The use of large language models (llm) for cyber threat intelligence (cti) in cybercrime forums. *arXiv preprint arXiv:2408.03354*, 2024.

[14] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv preprint arXiv:1911.02116*, 2019.

[15] Google DeepMind. Gemini 2.5 pro, 2025. Accessed: 2025-11-14.

[16] Confederazione Nazionale dell'Artigianato e della piccola e media impresa. Costi dell'energia elevati per le pmi, cna: "necessario riformare la bolletta", jun 2025. Accessed: 2025-12-04.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[18] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.12345*, 2024. Published: April 24, 2024.

[19] ExplodingGradients. Ragas: Supercharge your llm application evaluations. https://github.com/explodinggradients/ragas, 2024.

[20] Yong Fang, Jian Gao, Zhonglin Liu, and Cheng Huang. Detecting cyber threat event from twitter using idcnn and bilstm. *Applied Sciences*, 10(17):5922, 2020.

[21] Romy Fieblinger, Md Tanvirul Alam, and Nidhi Rastogi. Actionable cyber threat intelligence using knowledge graphs and large language models. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 100–111, 2024.

[22] Muriel Figueredo Franco, Bruno Rodrigues, Eder John Scheid, Arthur Jacobs, Christian Killer, Lisandro Zambenedetti Granville, and Burkhard Stiller. Secbot: a business-driven conversational agent for cybersecurity planning and management. In *2020 16th international conference on network and service management (CNSM)*, pages 1–7. IEEE, 2020.

[23] Recorded Future. Recorded future cti platform, 2025. Accessed: 2025-11-21.

[24] Google. Google gemini, 2025. Accessed: 2025-11-24.

[25] Yuelin Hu, Futai Zou, Jiajia Han, Xin Sun, and Yilei Wang. Llm-tikg: Threat intelligence knowledge graph construction utilizing large language model. *Available at SSRN 4671345*, 2023.

[26] Liangyi Huang and Xusheng Xiao. Ctikg: Llm-powered knowledge graph construction from cyber threat intelligence. In *First Conference on Language Modeling*, 2024.

[27] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*. 3rd edition, 2025. Online manuscript released August 24, 2025.

[28] Gyeongmin Kim, Chanhee Lee, Jaechoon Jo, and Heuiseok Lim. Automatic extraction of named entities of cyber threats using a deep bi-lstm-crf network. *International journal of machine learning and cybernetics*, 11(10):2341–2355, 2020.

[29] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[30] LangChain. LangChain Docs - Summarization Middleware, 2025. Accessed: 2025-11-20.

[31] Ahmed Lekssays, Utsav Shukla, Husrev Taha Sencar, and Md Rizwan Parvez. Techniquerag: Retrieval augmented generation for adversarial technique annotation in cyber threat intelligence text. *arXiv preprint arXiv:2505.11988*, 2025.

[32] Jian Liu, Junjie Yan, Jun Jiang, Yitong He, Xuren Wang, Zhengwei Jiang, Peian Yang, and Ning Li. Tricti: an actionable cyber threat intelligence discovery system via trigger-enhanced neural network. *Cybersecurity*, 5(1):8, 2022.

[33] Xinzheng Liu and Zhaoyun Ding. Cyberre-llm: Cyber threat intelligence relation extraction with large language model. In De-Shuang Huang, Haiming Chen, Bo Li, and Qinhu Zhang, editors, *Advanced Intelligent Computing Technology and Applications*, pages 261–271, Singapore, 2025. Springer Nature Singapore.

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[35] Paloma Martínez, Alberto Ramos, and Lourdes Moreno. Exploring large language models to generate easy to read content. *Frontiers in Computer Science*, 6:1394705, 2024.

[36] Microsoft. Microsoft security copilot, 2025. Accessed: 2025-11-21.

[37] Shaswata Mitra, Subash Neupane, Trisha Chakraborty, Sudip Mittal, Aritran Piplai, Manas Gaur, and Shahram Rahimi. Localintel: Generating organizational threat intelligence from global and local cyber knowledge. In Kamel Adi, Simon Bourdeau, Christel Durand, Valérie Viet Triem Tong, Alina Dulipovici, Yvon Kermarrec, and Joaquin Garcia-Alfaro, editors, *Foundations and Practice of Security*, pages 63–78, Cham, 2025. Springer Nature Switzerland.

[38] Neo4j. Neo4j: The world's leading graph database, 2025. Accessed: 2025-11-14.

[39] Ollama. Ollama, 2025. Accessed: 2025-11-14.

[40] OpenAI. Chatgpt, 2025. Accessed: 2025-11-24.

[41] OpenAI. Gpt-4.1, 2025. Accessed: 2025-11-14.

[42] Schulze Brock P, Katsinis A, Lagüera González J, Di Bella L, Odenthal L, Hell M, Lozar B, and Secades Casino B. Annual report on european smes 2024/2025, sme performance review. Technical Report KJ-01-25-292-EN-N (online), Luxembourg (Luxembourg), 2025.

[43] Tommaso Paladini, Lara Ferro, Mario Polino, Stefano Zanero, and Michele Carminati. You might have known it earlier: Analyzing the role of underground forums in threat intelligence. In *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '24, page 368–383, New York, NY, USA, 2024. Association for Computing Machinery.

[44] Md Rayhanur Rahman, Rezvan Mahdavi-Hezaveh, and Laurie Williams. A literature review on mining cyberthreat intelligence from unstructured texts. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 516–525, 2020.

[45] Yitong Ren, Yanjun Xiao, Yinghai Zhou, Zhiyong Zhang, and Zhihong Tian. Cskg4apt: A cybersecurity knowledge graph for advanced persistent threat organization attribution. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[46] Kiavash Satvat, Rigel Gjomemo, and VN Venkatakrishnan. Extractor: Extracting attack behavior from threat reports. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 598–615. IEEE, 2021.

[47] Amazon Web Services. Recorded future intelligence platform pricing, 2025. Accessed: 2025-12-04.

[48] Samaneh Shafee, Alysson Bessani, and Pedro M. Ferreira. Evaluation of llm-based chatbots for osint-based cyber threat awareness. *Expert Systems with Applications*, 261:125509, 2025.

[49] Sayuj Shah and Vijay K. Madisetti. Mad-cti: Cyber threat intelligence analysis of the dark web using a multi-agent framework. *IEEE Access*, 13:40158–40168, 2025.

[50] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[51] Giuseppe Siracusano, Davide Sanvito, Roberto Gonzalez, Manikantan Srinivasan, Sivakaman Kamatchi, Wataru Takahashi, Masaru Kawakita, Takahiro Kakumaru, and Roberto Bifulco. Time for action: Automated analysis of cyber threat intelligence in the wild. *arXiv preprint arXiv:2307.10214*, 2023.

[52] Mohammad Golam Sohrab and Makoto Miwa. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2843–2849, 2018.

[53] Sanchana Srikanth, Mohammad Hasanuzzaman, and Farah Tasnur Meem. Evaluating the usability of llms in threat intelligence enrichment. *arXiv preprint arXiv:2409.15072*, 2024.

[54] Statcounter. Desktop Operating System Market Share Worldwide, oct 2025. Accessed: 2025-11-20.

[55] Statcounter. Mobile Operating System Market Share Worldwide, oct 2025. Accessed: 2025-11-20.

[56] Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2023.

[57] W3Techs Web Technology Surveys. Usage statistics of Linux for websites, 2025. Accessed: 2025-11-20.

[58] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[59] Amine Tellache, Abdelaziz Amara Korba, Amdjed Mokhtari, Horea Moldovan, and Yacine Ghamri-Doudane. Advancing autonomous incident response: Leveraging llms and cyber threat intelligence. *arXiv preprint arXiv:2508.10677*, 2025.

[60] Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639:130193, 2025.

[61] Hanxiang Xu, Shenao Wang, Ningke Li, Kailong Wang, Yanjie Zhao, Kai Chen, Ting Yu, Yang Liu, and Haoyu Wang. Large language models for cyber security: A systematic literature review. *ACM Trans. Softw. Eng. Methodol.*, September 2025. Just Accepted.

[62] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *ACM Trans. Knowl. Discov. Data*, 18(6), April 2024. Place: New York, NY, USA Publisher: Association for Computing Machinery.

[63] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, pages 1–46, 2025.

A Knowledge Graph Characterization

We provide a concise description of the assembled knowledge graph to contextualize the information sources used in evaluating the answer-generation experiments. After extracting entities and relationships from the crawled documents, the final graph consists of 60 885 nodes, connected with 221 933 relationships, with an approximate density of 0.000060. As expected, the graph is sparse. Table 2 shows the distribution of the top relationship types in the graph. The most frequent relationship is "targets", which usually links a malware entity to an organization's. This is very relevant information to have, given the scope of the work. If many similar organizations are targeted by a certain malware, the user should be notified that their organization could be affected as well. The average degree of a node is 7.29, with the max degree being 4 581. As can be noted from Figure 6, there are a few hub nodes degree in the same order of magnitude. The obvious hub nodes are the dates (particularly years like 2022, 2012, 2017, ...), locations (US, Russia, China, Korea, ...), and the types ("Trojan", "malware", ...). There are a couple of less obvious insight from this. First of all, Android is the only OS in the top 20 hub nodes, indicating a higher prevalence of mobile threats in the context information. One of the hub nodes corresponds to a specific individual, a consultant who appears frequently across the reports but is not a cybercriminal. Although this person is a public figure, their name has been redacted to preserve privacy. Figure 7 shows the complete distribution of node degrees.

Table 2: Top relationship type distribution

Type	**Count**
targets	70 323
has	58 677
indicates	31 981
discoveredIn	29 046
isA	15 628
uses	9 974
madeBy	4 782
variantOf	1 058
hasAlias	463

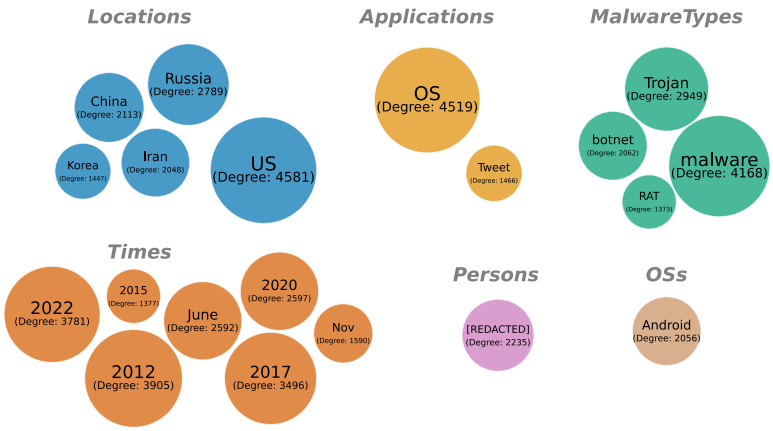


Figure 6: Top 20 entities by degree. Size of the circle is proportional to the degree of the entity.

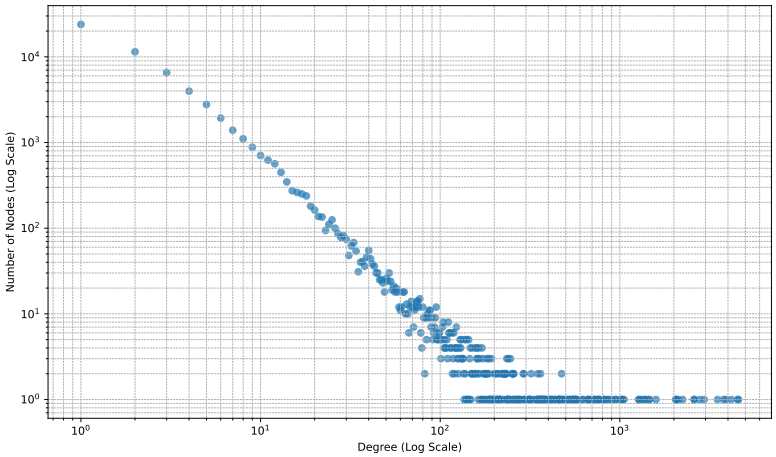


Figure 7: Full distribution of entity node degrees in logarithmic scale.