COMAT: A Cybersecurity Ontology based on MITRE ATT&CK

Yi-Ting Huang, R. Vaitheeshwari, Meng Chang Chen, Ying-Dar Lin, Ren-Hung Hwang, Po-Ching Lin, Yuan-Cheng Lai, Eric Hsiao-kuang Wu, Zi-Jie Liao, and C.K Chen

***Abstract*—This article presents COMAT, a cybersecurity ontology based on the MITRE ATT&CK framework for enhanced knowledge access and analysis. COMAT derives inference paths to identify adversarial techniques and includes forward- and backward-query modules for efficient, comprehensive Cyber Threat Intelligence (CTI) analysis.**

I. Introduction

As cyber threats evolve, we face an increasingly complex cybersecurity landscape. Advanced persistent threats (APTs) continue to endanger both individuals and organizations. For instance, in September 2020, Equinix, a large data center company, was hit by ransomware, potentially putting sensitive information and thousands of individuals' data at risk [1]. To combat these threats, companies publish cyber threat intelligence (CTI) reports that detail specific attack scenarios, sharing valuable insights. Companies like Microsoft, Google, FireEye, Trend Micro, etc, release these CTI reports to share their findings on specific threat actors, underscoring CTI's essential role in cybersecurity.

However, while CTI reports provide valuable information on specific attacks, they often lack a structured way to connect these insights to the overall strategies and actions used by adversaries. To address this, high-level models such as Lockheed Martin Kill Chain®, have been developed to outline the stages of an attack, exposing adversary goals and overarching processes. These models help understand the general objectives of adversaries, like reconnaissance and command-and-control (C&C), but they fall short in explaining the specific techniques and actions used to achieve these goals. Similarly, low-level concepts, like common vulnerabilities and exposures (CVEs), focus on individual software exploits but do not provide a complete view of the threat scenario. Though forensic indicators are useful in hunting fragmented views of threats, these indicators cannot uncover the complete threat scenario [2] [3]. Both high-level and low-level models, therefore, may not be sufficient for security experts to analyze adversarial actions comprehensively.

This gap is where the MITRE ATT&CK framework, a mid-level approach, becomes essential. It defines tactics and techniques to describe an adversary's goals and specific actions. Unlike high-level models that focus only on objectives or low-level details that cover individual indicators, MITRE ATT&CK offers a complete view of adversarial behaviors. This helps security analysts understand and map out the full scope of an attack. For instance, the adversary group admin@338 used phishing emails with malicious attachments to initiate an intrusion—a technique categorized in ATT&CK as "spearphishing attachment."

MITRE provides two main interfaces to access its knowledge base: the ATT&CK website and the Navigator tool. While the website offers comprehensive information on tactics and techniques, the Navigator helps visualize defensive coverage. However, both tools can be limited when it comes to complex knowledge queries, such as associating specific observed techniques with a known threat group during forensic analysis. This limitation highlights the need for more user-friendly and flexible systems, leading to the development of an ontology designed to infer attack techniques from CTI reports by integrating the structured knowledge of the MITRE ATT&CK framework.

An ontology is composed of key classes and their properties as a blueprint of a specific domain [4]. When building an ontology of MITRE ATT&CK as a dictionary of adversary techniques, security analysts can associate observed techniques with tactical objectives, malware, and threat groups. Existing studies on cybersecurity ontologies [2]–[10], often focus on IoC-level information [2], [6], [7], linking attack techniques to known vulnerabilities [10], while others [3]–[5], [8] emphasize information sharing, limiting their ability to reason adversary goals, lifecycles and groups. When querying the ontology, previous studies [10], [12] retrieve information through exact matching, which can miss relevant and implicit results.

To address these limitations, we present COMAT, a Cybersecurity Ontology based on MITRE ATT&CK, designed to address these gaps. COMAT builds on the existing MITRE ATT&CK schema [11] and includes an additional class, "threat action." These threat actions are extracted from procedure examples on the MITRE ATT&CK webpage using semantic

Yi-Ting Huang is with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

R. Vaitheeshwari, Eric Hsiao-Kuang Wu and Zi-Jie Liao are with the Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan.

Meng-Chang Chen is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan.

Ying-Dar Lin is with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan.

Ren-Hung Hwang is with the College of Artificial Intelligence, National Yang Ming Chiao Tung University, Hsinchu, Taiwan.

Po-Ching Lin is with the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan.

Yuan-Cheng Lai is with the Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan.

C.K Chen is with the Research Division, CyCraft Technology, New Taipei City, Taiwan

role labeling (SRL), providing a deeper understanding of how adversaries operate. In the field of cybersecurity, analysts often focus on understanding the relationships among adversary groups, malicious software, and the techniques they use. For instance, when a country or company is targeted by a threat group, security analysts examine the attack group, including the techniques and software they employed. To support this, COMAT is designed with forward- and backward-query capabilities within the MITRE ATT&CK framework, enabling analysts to associate related information and infer potential threats effectively.

Our main contributions are as follows:

- We present COMAT, an ontology built upon the MITRE ATT&CK framework, which incorporates an additional class, "threat action", serving as anchors for technique inference.
- We develop a semantic embedding approach to infer the most closely related ATT&CK Techniques based on known entities, overcoming the limitations of exact matching when querying the ontology.
- We demonstrate forward- and backward-query modules to help security analysts explore the interactions among adversary groups, software, and techniques.

II. Background and Related Work

A. MITRE ATT&CK Framework

MITRE ATT&CK, an open cyber threat knowledge base, includes adversary tactics and techniques based on real-world incident observations. Its core elements include:

- **Tactic.** Describes the purpose of the adversary, representing different lifecycle stages. For example, *TA0001 Initial Access* aims to gain access to the victim's system.
- **Technique.** Depicts the specific action taken to achieve a tactic. For example, *T1566 Phishing* is used to accomplish Initial Access by sending phishing emails.
- **Group.** Describes threat actor behaviors using Tactics, Techniques, and Procedures (TTPs). For instance, *APT29* uses *T1547.001* for persistence and *T1555.003* to steal credentials.
- **Software.** Malicious software used by adversaries, including publicly available tools (e.g.,command line interpreter (CMD)) or malware (e.g., Ragnar Locker ransomware).
- **Mitigation.** Methods to prevent a technique from succeeding. For example, *M1017 User Training* helps mitigate phishing attempts (*T1566*).

COMAT extends this framework by developing an ontology designed to infer and analyze the implicit relationships among various classes.

B. Ontology

An ontology broadly describes concepts within a domain via classes and properties and their relationship [4]. In a domain, classes and their associated properties are defined to describe the domain knowledge in a structured format. Classes contain one or more sub-classes to address specific information and comprise properties to describe features and attributes. The connection between the two classes represents a relationship between them. Instances in an ontology mean the individual example of classes have different values for object properties.

In the context of COMAT, we build upon the existing structure of MITRE ATT&CK, which already defines core entities such as Tactics, Techniques, Adversary Groups, Software, and Mitigations. COMAT extends this ontology by incorporating semantic reasoning and includes an additional class called "Threat Action." This allows COMAT to infer connections between existing knowledge and newly observed adversarial behaviors from CTI reports.

C. Natural Language Processing in Cybersecurity

To extract cybersecurity-related information in unstructured texts like CTI report, well-known Natural Language Processing (NLP) techniques are used and summarized as follows.

- **Part Of Speech (POS) tagger** is used to label the Part-of-Speech, such as verb, noun, adjective, to the corresponding word in a text.
- **Dependency parser** extracts dependency relations among words in a sentence. This represents its grammatical structure and defines the relationships between words and words.
- **Semantic Role Labeler (SRL)** annotates the semantic roles (arguments) in a sentence. Specifically, SRL investigates a given sentence with a question, "who" did "what" to "whom."

These techniques allow COMAT to transform CTI report into structured knowledge, enhancing its ability to analyze and predict adversary behavior.

D. Cybersecurity Ontology

Syed et al. [5] provided Unified Cybersecurity Ontology (UCO) based on cybersecurity standards named STIX to serve as the backbone. For example, using UCO to cover Adobe Acrobat to PDF reader and searching for related information in other open data.

Rastogi et al. [4] introduced MALOnt, an ontology to structure malware threat intelligence by integrating unstructured and diverse data sources. They later developed CyNER, an open-source library for cybersecurity named-entity recognition, to identify entities like indicators of compromise (IoCs), though their work remains focused on low-level data representation, such as specific malware family attributes.

Zhao et al. [8] proposed an ontology to unify multi-source CTI data into a JSON model, categorizing entities such as threat actors, malware, and associated IoCs. Gao et al. [2] created a threat behavior graph that models cyber threats by extracting IoCs and translating them into system-level queries for threat hunting. For instance, they extract file paths and commands, like "WRITE" to "/TMP/UPLOAD.TAR," as part of malicious activity patterns.

Mittal et al. [6] introduced CyberTwitter, a system to issue alerts by gathering and reasoning over security terms on social media, tagging elements like vulnerabilities and attack periods,

though focusing on immediate alerts rather than detailed adversarial tactics. Similarly, Akbar et al. [10] developed a cybersecurity ontology by extracting structured entities from the MITRE ATT&CK framework and linking them to CVE data for APT analysis.

Gao et al. [7] propose schemas to model relationships between infrastructure nodes (e.g., IP addresses, domains) and classify node types using a graph convolutional network. Our previous work [12] developed heuristic rules to extract valuable information from the MITRE ATT&CK framework for constructing a cybersecurity ontology, demonstrating its usefulness in cyber threat analysis, particularly for malware analysis when given techniques.

In summary, most prior work focuses on extracting surface-level clues, such as IOCs, and often lacks mechanisms to make inferences from implicit clues. COMAT addresses this gap by incorporating multi-hop inference and embedding-based reasoning capabilities, enabling the inference of tactics, techniques, threat groups, and manipulated tools from observable data.

E. Knowledge Graphs

Piplai et al. [3] developed a Cybersecurity Knowledge Graph (CKG) to consolidate extracted data for cyber-incident analysis, using Stanford NER and regular expressions to structure entities like malware, software, IoCs, and attack patterns in STIX format. Though CKG links attack patterns with associated software and IoCs, it lacks a reasoning mechanism for interpreting complex threat actions.

Husari et al. [9] introduced TTPDrill, an open-source system using a threat-action ontology to map adversarial actions to the ATT&CK framework. However, its fixed ontology limits scalability, and dependency parsing combined with BM25 often yields false positives. For example, parsing a sentence like "Dipsind encodes C2 traffic with base64" can generate broad, loosely related TTPs.

Our work introduces COMAT, an inference-based threat-action ontology designed to overcome these limitations. COMAT uses a SRL approach that interprets full sentence meaning, focusing on key elements indicative of malicious intent to accurately identify techniques (e.g., only two TTPs are matched in the example above, rather than twelve). COMAT further introduces six meaningful CTI-based query paths, enabling refined technique inference based on CTI context. Our ontology combines CKG's security knowledge mining capabilities with TTPDrill's technique inference functionality, supporting both comprehensive knowledge queries (e.g., finding groups associated with both *T1189 Drive-by Compromise* and *T1005 Data from Local System*) and context-aware threat action inference.

III. Task Definition

To address the task of organizing and analyzing cyber threat intelligence, our approach divides the problem into two main sub-tasks: ontology construction and ontology inference.

- **Ontology Construction**. A cyber ontology is a structured framework that represents knowledge about cybersecurity concepts, their properties, and the relationships between them. Fig. 2 describes the proposed ontology for COMAT, including six classes with their attributes and five relationships. An ontology O can be represented as a tuple: $O = (C, R, A, I)$, where C is classes or concepts in a domain, R denotes relations between two classes, A refers to attributes of a class, a set of values associated with a class, and I as instances of a class.
  One additonal class, *threat action* is introduced in the proposed onotlogy. A threat action is a specific and observable step performed by a threat actor, malware, or adversarial group to achieve a malicious objective. It is represented as a combination of an action (verb) and an object, such as create directory or modify registry, which collectively implements a technique within the MITRE ATT&CK framework. A threat action TA can be defined as $VO = (V, O)$, where V denotes a verb phrase representing the action (e.g., create, delete, modify), and O is an object representing the target or entity acted upon (e.g., directory, file, registry).
- **Ontology Inference**. Given a query Q, where Q can be a natural language, a single instance, or a class value, such as a sentence from a CTI report, a threat action-like phrase, a specific threat group, or software, the objective is to infer the most relevant technique(s) within the ontology. To achieve this, we present 1) a semantic similarity to match Q with potential techniques, particularly when Q is a natural language description, and 2) a multi-hop inference over O to traverse intermediate nodes and relationships, enabling the inference of related techniques at varying levels—direct (1-hop), second-degree (2-hop), and third-degree (3-hop) connections—where each "hop" represents one level of relationship in the ontology.

IV. Methodology

COMAT is composed of four modules, including ontology definition, data extraction, query mapping and technique inference. These modules work together to map user queries to the ontology and infer relevant techniques. The workflow of COMAT is shown in Fig. 1.

A. Ontology Construction

Ontology definition. The proposed ontology for COMAT includes six classes with their attributes and five primary relationships, as shown in Fig. 2. COMAT builds on the five core data models of the MITRE ATT&CK framework—Tactic, Technique, Group, Software, and Mitigation—as its foundational classes. To enhance the detail of adversary techniques, COMAT includes an additional class, *threat action*, derived from procedural examples to support technique inference and CTI analysis. Following findings from previous studies [9] [13], threat actions are defined as a combination of a verb and an object, representing actionable steps and observable objects that collectively implement a technique. Additionally, to better track potential adversary groups, COMAT includes the attributes location country and target country in the Group class.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2025.3562901

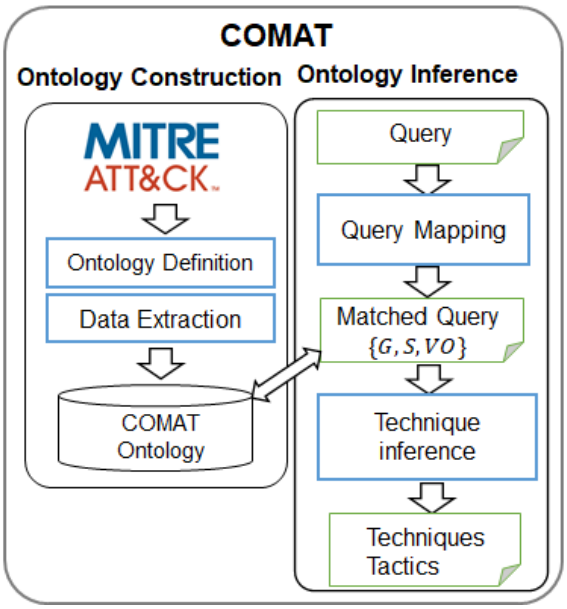

Fig. 1. The workflow of COMAT.

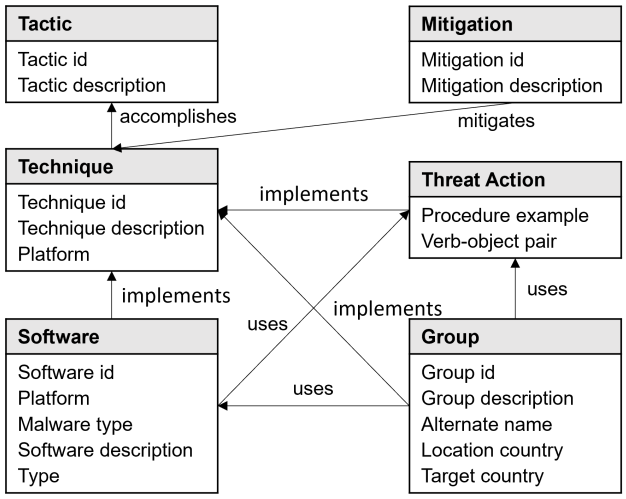

Fig. 2. COMAT ontology classes, attributes and relations

Data extraction. Based on the proposed ontology, COMAT extracts key attributes, including verb-object pairs for Threat Action, Location Country and Target Country for Group, and Malware Type for Software. Notably, COMAT incorporates automatic data extraction to fill in missing values, even though structured repositories like STIX make cyber threat knowledge accessible.

- **Verb-object pair.** Procedure examples in MITRE ATT&CK often present real-world tactics in simple structures, generally containing a subject, verb, and object. Unlike complex sentences in CTI reports, these examples are more straightforward, allowing for efficient extraction. COMAT uses regular expressions to clean up extraneous information (e.g., citations, brackets), after which an SRL approach [14] tags verb-object pairs. COMAT designates VERB and ARG1 labels as the action and object, respectively. For example, consider a procedural instance, "Axiom has used spear phishing to initially compromise victims." Here, the verb-object pairing of *"use"* and *"spear phishing"* aligns with the technique *T1566 Phishing*.
- **Location country and Target country.** Adversarial and target country information is extracted from descriptions on the Group page. The word "target" often divides descriptions, with the initial part indicating the adversary's origin and the latter part suggesting the target country.
- **Malware type.** Malware type describes what malware does on a victim machine, such as trojan, worm, and backdoor. COMAT identifies these types following the Microsoft Malware Naming Scheme.

B. Ontology Inference

The ontology inference in COMAT is achieved through a query mapping module, which aligns user input to the ontology structure and utilizes a predefined inference path, Φ, to identify the most probable techniques by traversing the mapped nodes.

Query Mapping. When a query is received, typically containing elements like Group, Software, and Threat Action, the module maps each part to a corresponding instance in the ontology. Group and Software queries are directly matched, while Threat Action may involve more nuanced interpretation with synonyms or related terms. To achieve this, COMAT employs Sentence-BERT to convert both query values and ontology terms into fixed-size vectors. The cosine similarity between the input and ontology embeddings is calculated, and if it exceeds a set threshold, the input aligns with the ontology instance.

For example, as shown in Fig. 3a, when a query is given as JSS Loader, contain, malicious file, the Software element "JSS Loader" is mapped directly via exact matching, while "contain" and "malicious Microsoft Excel attachments" are matched with Threat Action by fuzzy matching (i.e., through embeddings and cosine similarity). This allows COMAT to accurately align user queries with ontology instances.

Technique Inference. To identify the most relevant technique, COMAT employs six predefined inference paths, illustrated in Fig. 3b, to reason techniques based on the available input nodes.

- 1-Hop Paths: Three single-hop paths—Φ_1, Φ_2 and Φ_3, derive a technique when a single element, such as Verb-Object pair, Group, or Software, is provided.
- 2-Hop Paths: Two dual-hop paths—Φ_4 and Φ_5—combine Group or Software with a Verb-Object pair to enhance technique inference. For instance, in Fig. 3, when both Software and Threat Action are combined, COMAT successfully infers Technique *T1566 Phishing*.
- 3-Hop Path: For improved inference accuracy, the 3-hop path, Φ_6 integrates Group, Software, and Verb-Object pairs to deduce the most probable technique.

The fuzzy matching mechanism is applied in processes involving Threat Actions (VO), specifically in Φ_1, Φ_4, Φ_5 and Φ_6. In these processes, the Threat Action (VO) can utilize

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2025.3562901

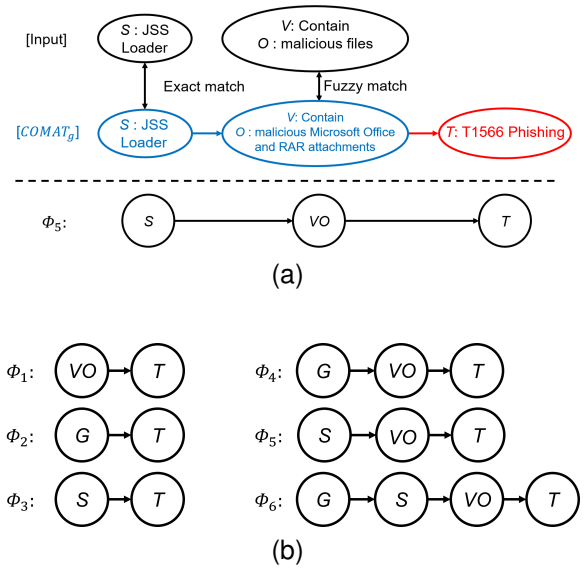

Fig. 3. Ontology Inference (a) Query mapping module and technique inference, and (b) Six inference paths.

fuzzy matching, while Group (G) and software (S) require exact matching. Once this exact match is confirmed, the fuzzy matching mechanism maps the verb and object components of the query to the corresponding Threat Action in the ontology. Thus, the semantic similarity-based inference operates under the condition that either the software or group value is an exact match. This approach ensures that the fuzzy matching results remain consistent with the underlying ontology structure and are applied in a controlled and precise manner, minimizing the risk of inconsistencies across different queries.

Once a Technique is determined, the associated Tactic is directly derived via the ATT&CK framework, offering a complete view of the adversary's tactical objective.

V. Evaluation

A. Settings

Dataset. For evaluation, we use two MITRE ATT&CK versions: v6 and v10. The ATT&CK framework for Enterprise includes 14 tactics, 566 (sub-)techniques, 539 software, 129 adversary groups, and 43 mitigations. We collected 100 CTI reports corresponding to 113 techniques in v10 and 107 techniques in v6.

Ontology Implementation. We processed the ATT&CK repository to construct the COMAT ontology, storing it in a Neo4j graph database. This resulted in 10,534 instances and 30,351 relationships, covering 9,238 procedure examples with identified Threat Actions and Verb-Object pairs. The COMAT code is released on the GitHub page, https://github.com/wmlab-MITREtreival/COMAT_Ontology.

Basesline. To evaluate COMAT's performance, we compare it with TTPDrill [9] using our dataset. Since TTPDrill is based on MITRE v6, we convert version 10 Technique IDs to their v6 counterparts (e.g., T1064 Scripting to T1059 Command and Scripting Interpreter). Techniques in v10 without v6 equivalents are omitted from the MITRE v6 dataset.

Evaluation Metrics. We consider both macro- and micro-average of precision, recall and F1 socre to evaluate the performance of Technique and Tactic inference.

Complexity Analysis. For query mapping, the similarity computation of a given query and ontology has a complexity of $O(Nd)$, where N is the number of ontology terms, and d is the dimensionality of the embeddings from Sentence-BERT. For multi-Hop inference, multi-hop reasoning involves traversing the ontology graph, with a complexity of $O(EH)$, where E is the number of edges in the graph and H is the maximum number of hops.

Execution Time. COMAT's query execution times were measured for each processing stage. The text processing took an average of 25 seconds per document, preparing data for subsequent analysis. For query node extraction, the average response time was 0.17 seconds, efficiently mapping initial inputs. During technique inference, which involves multi-hop semantic matching and reasoning across groups, software, and threat actions, the response time averaged 41 seconds.

Convergence Analysis. The proposed ontology inference achieves technique identification through convergence via multi-hop reasoning. Empirical analysis of 100 CTI reports in our dataset shows that the inference process typically converges within an average of 1.6 hops for most queries. However, 21% of the reports did not yield results, suggesting opportunities for further refinement in processing queries derived from CTI descriptions or optimizing threshold selection in semantic similarity.

B. Results on Technique and Tactic Inference

Evaluation results on both MITRE v6 and v10 datasets are shown in Table I. It indicates that COMAT is better than TTPDrill at precision and F1-scores, while the recall scores are not better than TTPDrill. COMAT leverages fuzzy matching to map ambiguous or semantically similar threat actions to ontology terms, whereas TTPDrill relies on exact keyword matches, which can limit its flexibility. For instance, consider the example: *The malware scans the list of running processes looking for outlook, iexplore, or firefox and injects the DLL into the process.* Using its semantic similarity mechanism, COMAT successfully identifies two techniques: T1057 (Process Discovery) and T1055 (Process Injection). The phrase "scans the list of running processes" is semantically matched to the threat action associated with T1057, while "injects the DLL into the process" corresponds to the threat action linked to T1055 within the ontology. In contrast, TTPDrill, which depends solely on exact keyword matches, fails to recognize T1057 (Process Discovery).

COMAT's multi-hop reasoning allows it to infer techniques by integrating relationships across adversary groups(G), software(S), and threat actions(VO). For instance, in the sentence, *After the Zebrocy Trojan is activated by unsuspecting users via spear-phishing attachments from the Sofacy group, it immediately begins to collect system information and periodically sends this data to a C2 server using HTTPS, a protocol that masks its communication within regular traffic, thus evading traditional detection mechanisms*, COMAT successfully infers T1041 (Exfiltration Over C2 Channel) by leveraging a

TABLE I
PERFORMANCE ON TECHNIQUE AND TACTIC INFERENCE

Dataset	MITRE v10 (113)						MITRE v6 (107)					
Metrics	Macro			Micro			Macro			Micro		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
TTPDrill [9]	0.13	0.65	0.20	0.14	0.64	0.23	0.12	0.58	0.17	0.14	0.66	0.23
COMAT	0.20	0.42	0.25	0.24	0.54	0.33	0.20	0.44	0.25	0.24	0.55	0.34
TTPDrill [9] (Tactic)	0.17	0.55	0.26	0.13	0.67	0.22	0.18	0.59	0.28	0.14	0.69	0.23
COMAT (Tactic)	0.29	0.54	0.35	0.32	0.64	0.43	0.30	0.56	0.36	0.33	0.65	0.43

multi-hop reasoning path involving group (Sofacy), software (Zebrocy), and threat actions including (begin, to collect system information), (collect, system information), (send, this data), and (execute, HTTPS). The fuzzy matching mechanism achieves a similarity score of 0.807 (above the threshold of 0.7) against the ontology class T1041's threat action (exfiltrates, data over the C2 channel). In contrast, TTPDrill misidentifies the techniques as T1496 (Resource Hijacking) and T1190 (Exploit Public-Facing Application), erroneously mapping isolated phrases like *system information* and *using HTTPS* to unrelated techniques.

A closer look at recall scores reveals that COMAT, while effective in capturing a wide range of tactics, faces challenges in accurately identifying either overly common or infrequently used techniques. Some techniques may only be employed by a limited number of APT groups, while some may be utilized by more than 50 APT groups. For example, techniques such as *T1057*, *T1071.001*, and *T1059.001* are associated with multiple APT groups. On the other hand, some techniques, including *T1612*, *T1580*, and *T1526*, lack any APT group association altogether. This lack of group representation for these techniques contributes to higher rates of false negatives.

C. Ontology Functionality

Here we demonstrate the benefit of Ontology inference using Cypher queries [15] on the Neo4j database.

Query Knowledge. When security analysts seek to determine which attack group is associated with observed techniques, they can use specific queries to facilitate their investigation. For example, if a victim organization is targeted using Techniques *T1189 (Drive-by Compromise)* and *T1005 (Data from Local System)*, identifying these techniques enables analysts to assess potential threat groups and anticipate further actions. Using this information, analysts can structure a query to investigate incoming attacks and linked threat groups. The retrieved results are shown in Fig. 4.

```
QUERY:
    MATCH (g:Group)-[r1]-(t1:Technique {id:"t1189"})
    MATCH (g:Group)-[r2]-(t2:Technique {id:"t1005"})
    RETURN g.name

RESULT:
    [APT38,Andariel,Windigo,APT37,Dragonfly2.0
    ,Lazarus Group,Dark Caracal,BRONZE BUTLER,
    Threat Group-3390,Patchwork,Turla]
```

Forward and Backward Query. Forward query retrieves the related Group and Software instances when given a Technique,

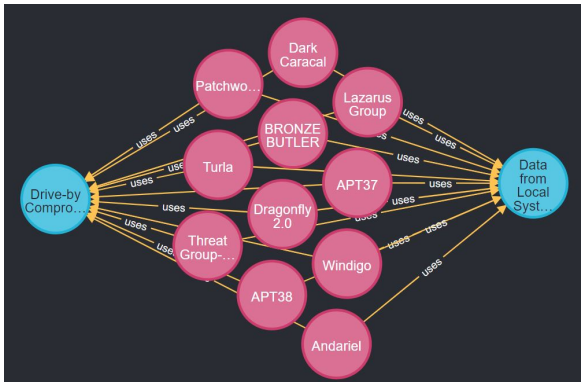

Fig. 4. Result for a given query, here red nodes represent Groups and blue nodes are Techniques.

while backward query lists the known Techniques when given either a Group or Software instance. Since some CTI reports uncover the MITRE ATT&CK Techniques, as an adversary lifecycle, at the bottom of the reports, COMAT provides more information for security analysts to understand the relations among Technique, Group, and Software to attack investigation. Here list some examples as below.

```
FORWARD-QUERY:
    { Technique, T1529 } // Name:System Shutdown/
        Reboot

RESULT:
    {Class:Software,MITRE_id:s0607,Name:KillDisk}
    {Class:Software,MITRE_id:s0582,Name:LookBack}
    {Class:Software,MITRE_id:s0449,Name:Maze}
    {Class:Software,MITRE_id:s0140,Name:Shamoon}
    {Class:Software,MITRE_id:s0368,Name:NotPetya}
    {Class:Software,MITRE_id:s0372,Name:LockerGoga}
    {Class:Software,MITRE_id:s0365,Name:Olympic
     Destroyer}
    {Class:Group,MITRE_id:g0032,Name:Lazarus Group}
    {Class:Group,MITRE_id:g0082,Name:APT38}
    {Class:Group,MITRE_id:g0067,Name:APT37}
```

```
BACKWARD-QUERY-1:
    { Group, G1039 } // Name:Suckfly

RESULT:
    {Class:Technique,MITRE_id:t1553.002,Name:Code
        Signing}
    {Class:Technique,MITRE_id:t1003,Name:OS
        Credential Dumping}
    {Class:Technique,MITRE_id:t1059.003,Name:Windows
         Command Shell}
    {Class:Technique,MITRE_id:t1078,Name:Valid
        Accounts}
```

```
    {Class:Technique,MITRE_id:t1046,Name:Network
        Service Scanning}
    {Class:Software,MITRE_id:s0118,Name:Nidiran}
```

```
BACKWARD-QUERY-2:
    { Software, S0069 } // Name:BLACKCOFFEE

RESULT:
    {Class:Technique,MITRE_id:t1553.002,Name:Code
        Signing}
    {Class:Technique,MITRE_id:t1003,Name:OS
        Credential Dumping}
    {Class:Technique,MITRE_id:t1059.003,Name:Windows
         Command Shell}
    {Class:Technique,MITRE_id:t1078,Name:Valid
        Accounts}
    {Class:Technique,MITRE_id:t1046,Namhttps://www.
        overleaf.com/project/63
        e5dc31979008015885bb2ce:Network Service
        Scanning}
    {Class:Software,MITRE_id:s0118,Name:Nidiran}
```

VI. Conclusion and Future Work

In this work, we introduced COMAT, a cybersecurity ontology designed to efficiently leverage knowledge from the MITRE ATT&CK framework. COMAT automates the extraction of critical attributes, such as target countries and malware types, to support CTI investigations. Through inference paths, it deduces Techniques and Tactics based on observable clues, while forward and backward queries offer comprehensive protection strategies against adversarial techniques.

COMAT is specifically tailored for practical threat analysis, enabling the inference of tactics, techniques, threat groups, and manipulated tools from observable data. Looking forward, with access to high-quality and up-to-date MITRE ATT&CK data, COMAT could become a foundation for advanced research initiatives, including AI-driven CTI analysis for threat prediction, stronger defenses against emerging and n-day attacks, and proactive defense strategies.

In future work, we will enhance COMAT's inference ability. Also, we plan to extend COMAT's capabilities by integrating multi-source data, such as system logs, to enhance cybersecurity insights and query quality.

Moreover, with the rapid advancement of large language models (LLMs) that can process flexible user prompts, the integration of LLMs with COMAT could significantly enhance user interaction. COMAT is designed to offer structured, high-quality cybersecurity data, and we believe that combining it with LLMs would not only improve flexibility but also mitigate issues like hallucination during inference, ensuring more accurate and reliable results. Future work could focus on developing a seamless integration between the ontology and LLMs, enabling a more user-friendly experience and accurate retrieval for threat analysis and decision support.

References

[1] I. Arghire, "Data Center Provider Equinix Hit by Ransomware," SecurityWeek, Sep. 11, 2020. [Online]. Available: https://www.securityweek.com/data-center-provider-equinix-hit-ransomware

[2] Gao, P., Shao, F., Liu, X., Xiao, X., Qin, Z., Xu, F., ... & Song, D. (2021, April). Enabling efficient cyber threat hunting with cyber threat intelligence. In 2021 IEEE 37th International Conference on Data Engineering (ICDE) (pp. 193-204). IEEE.

[3] Piplai, A., Mittal, S., Joshi, A., Finin, T., Holt, J., & Zak, R. (2020). Creating cybersecurity knowledge graphs from malware after action reports. IEEE Access, 8, 211691-211703.

[4] Rastogi, N., Dutta, S., Zaki, M. J., Gittens, A., & Aggarwal, C. (2020, August). Malont: An ontology for malware threat intelligence. In International Workshop on Deployable Machine Learning for Security Defense (pp. 28-44). Springer, Cham.

[5] Syed, Z., Padia, A., Finin, T., Mathews, L., & Joshi, A. (2016, March). UCO: A unified cybersecurity ontology. In Workshops at the thirtieth AAAI conference on artificial intelligence.

[6] Mittal, S., Das, P. K., Mulwad, V., Joshi, A., & Finin, T. (2016, August). Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities. In 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 860-867). IEEE.

[7] Gao, Y., Xiaoyong, L. I., Hao, P. E. N. G., Fang, B., & Yu, P. (2020). Hincti: A cyber threat intelligence modeling and identification system based on heterogeneous information network. IEEE Transactions on Knowledge and Data Engineering.

[8] Zhao, Y., Lang, B., & Liu, M. (2017, October). Ontology-based unified model for heterogeneous threat intelligence integration and sharing. In 2017 11 th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID) (pp. 11-15). IEEE.

[9] Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., & Niu, X. (2017, December). Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of cti sources. In Proceedings of the 33rd annual computer security applications conference (pp. 103-115).

[10] Akbar, K.A., Halim, S.M., Singhal, A., Abdeen, B., Khan, L. and Thuraisingham, B., 2023, April. The design of an ontology for ATT&CK and its application to cybersecurity. In Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (pp. 295-297).

[11] Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). MITRE ATT&CK: Design and Philosophy. (book)

[12] Huang, C.C., Huang, P.Y., Kuo, Y.R., Wong, G.W., Huang, Y.T., Sun, Y.S. and Chen, M.C., 2022, December. Building Cybersecurity Ontology for Understanding and Reasoning Adversary Tactics and Techniques. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 4266-4274). IEEE.

[13] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 755–766, 2016

[14] Satvat, K., Gjomemo, R., & Venkatakrishnan, V. N. (2021, September). EXTRACTOR: Extracting attack behavior from threat reports. In 2021 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 598-615). IEEE.

[15] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, et al.. Cypher:An Evolving Query Language for Property Graphs. SIGMOD'18 Proceedings of the 2018 International Conference on Management of Data, Jun 2018, Houston, United States. pp.1433,ff10.1145/3183713.3190657ff. ffhal-01803524f