




Article

VTT-LLM: Advancing Vulnerability-to-Tactic-and-Technique Mapping through Fine-Tuning of Large Language Model

Chenhui Zhang ¹, Le Wang ^{1,2,*} , Dunqiu Fan ³, Junyi Zhu ¹, Tang Zhou ¹, Liyi Zeng ²  and Zhaohua Li ⁴ 

¹ Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China; chenhui_zhang@gzhuedu.cn (C.Z.); 2112233067@e.gzhu.edu.cn (J.Z.); 2112333164@e.gzhu.edu.cn (T.Z.)

² Peng Cheng Laboratory, Shenzhen 518000, China; zengly@pcl.ac.cn

³ NSFOCUS Inc., Guangzhou 510006, China; fandunqiu@nsfocus.com

⁴ Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China; 13250240950@163.com

* Correspondence: wangle@gzhu.edu.cn

Abstract: Vulnerabilities are often accompanied by cyberattacks. CVE is the largest repository of open vulnerabilities, which keeps expanding. ATT&CK models known multi-step attacks both tactically and technically and remains up to date. It is valuable to correlate the vulnerability in CVE with the corresponding tactic and technique of ATT&CK which exploit the vulnerability, for active defense. Mappings manually is not only time-consuming but also difficult to keep up-to-date. Existing language-based automated mapping methods do not utilize the information associated with attack behaviors outside of CVE and ATT&CK and are therefore ineffective. In this paper, we propose a novel framework named VTT-LLM for mapping Vulnerabilities to Tactics and Techniques based on Large Language Models, which consists of a generation model and a mapping model. In order to generate fine-tuning instructions for LLM, we create a template to extract knowledge of CWE (a standardized list of common weaknesses) and CAPEC (a standardized list of common attack patterns). We train the generation model of VTT-LLM by fine-tuning the LLM according to the above instructions. The generation model correlates vulnerability and attack through their descriptions. The mapping model transforms the descriptions of ATT&CK tactics and techniques into vectors through text embedding and further associates them with attacks through semantic matching. By leveraging the knowledge of CWE and CAPEC, VTT-LLM can eventually automate the process of linking vulnerabilities in CVE to the attack techniques and tactics of ATT&CK. Experiments on the latest public dataset, ChatGPT-VDMEval, show the effectiveness of VTT-LLM with an accuracy of 85.18%, which is 13.69% and 54.42% higher than the existing CVET and ChatGPT-based methods, respectively. In addition, compared to fine-tuning without outside knowledge, the accuracy of VTT-LLM with chain fine-tuning is 9.24% higher on average across different LLMs.

Keywords: vulnerabilities; large language model; tactics and techniques; fine-tuning

MSC: 68T50



Citation: Zhang, C.; Wang, L.; Fan, D.; Zhu, J.; Zhou, T.; Zeng, L.; Li, Z.

VTT-LLM: Advancing Vulnerability-to-Tactic-and-Technique Mapping through Fine-Tuning of Large Language Model. *Mathematics* **2024**, *12*, 1286. <https://doi.org/10.3390/math12091286>

Academic Editor: Daniel-Ioan Curiac

Received: 29 March 2024

Revised: 18 April 2024

Accepted: 23 April 2024

Published: 24 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Common Vulnerabilities and Exposures list (CVE) [1] is the largest continuously growing repository of open vulnerabilities, which provides detailed descriptions of vulnerabilities. Cybersecurity staff can refer to this description to promptly identify vulnerabilities in their networks and further find the cause and scope of the vulnerabilities. However, CVE just helps identify and enumerate the vulnerabilities in enterprise networks and does not analyze the relevant attack behaviors that vulnerabilities cause. The Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) [2] serves as a comprehensive repository of tactics and techniques, which models the behaviors of attackers in a multi-step attack. A tactic is a high-level concept that describes the goals or intentions of the attacker, while

techniques describe in detail the implementation of the tactics, such as the explicit methods, tools, or behaviors. Intuitively, it is possible to establish a mapping between a Vulnerability in a CVE and an ATT&CK's Tactic and Technique (VTT), where the latter could exploit the former. This is valuable for security staff to re-conceptualize vulnerabilities from an attacker's perspective in enterprise networks, so that they can develop effective defense strategies to block hidden attacks[3,4].

Figure 1 illuminates the scenario. The defender first identifies CVE-2020-3330 in their enterprise network. Then, they infer through VTT that the corresponding tactic of the potential attack is "Initial Access" as well as the technique is "Valid Accounts". Consequently, they can adopt targeted mitigation such as "Account Locking" to block hidden attacks.

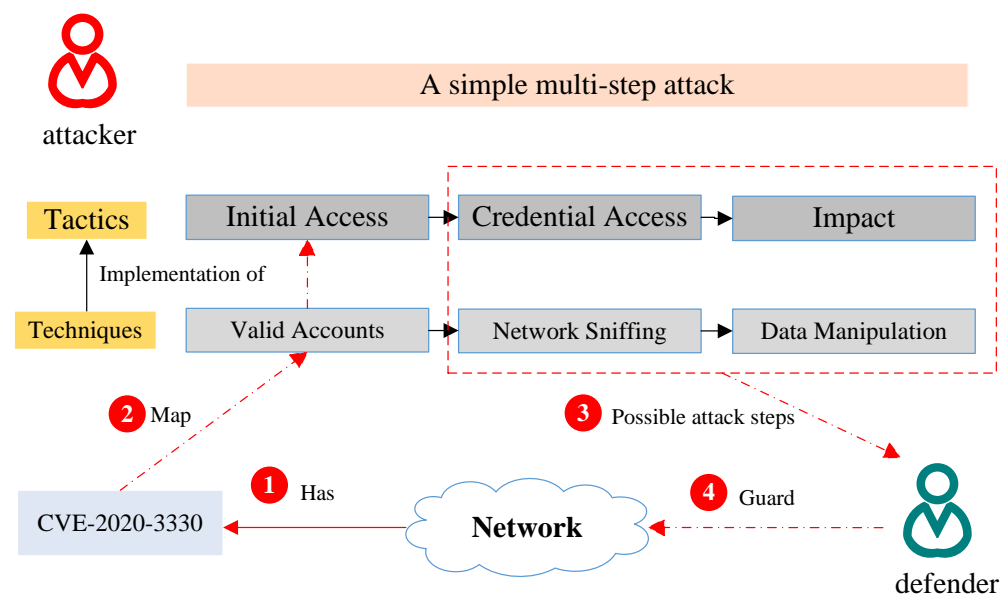


Figure 1. Defensive perspective: correlating CVE vulnerabilities with ATT&CK tactics and techniques for targeted mitigation.

However, manually building a VTT mapping is impractical in the face of emerging vulnerabilities and an ever-expanding category of tactics and techniques, so it is crucial to research automated methods to build a VTT mapping. The core challenge of automating VTT mapping is to extract semantics from lengthy vulnerability descriptions and then map them to multiple categories.

Prior researchers commonly employed language models to establish a VTT mapping, with vulnerability descriptions as input and categories of tactics or techniques as output. Benjamin Ampel and Chen [5] proposed CVET, which employed a self-knowledge distillation method alongside a fine-tuning process on RoBERTa. CVET was designed to identify key information within CVE descriptions and classify them into one of ten categories of ATT&CK tactics. This method utilized the power of knowledge distillation to achieve a 71.49% tactical accuracy but did not utilize technical and tactical descriptions. Abdeen et al. [6] introduced SEMT, which automatically mapped CVE entries to ATT&CK techniques based on their textual similarity. SEMT extracted attack vectors from detailed vulnerability descriptions using semantic role labeling and a BERT-based model, then mapped them to 1 of the 41 ATT&CK techniques with a logistic regression model trained by ATT&CK technique descriptions. SEMT utilized technical descriptions but required attack vectors as an intermediate step, which may lose some information. Past works have attempted to establish a static mapping from vulnerability descriptions to technical and tactical categories. The static mapping has to be reconstructed in the face of a category update in ATT&CK, which is generally inefficient.

Large language models (LLMs) are expected to fuel advances in VTT mapping because of their language understanding and generation capabilities. Nevertheless, Liu et al. [7] states that applying ChatGPT directly to VTT mapping does not work well, achieving a 32.76% tactical accuracy. This suggests that it is difficult for simple queries to utilize the potential of LLMs in solving domain-specific tasks such as the VTT mapping. Therefore, we fine-tuned an LLM to enhance its effectiveness in VTT applications. Many fine-tuning and prompt engineering methods such as Low-Rank Adaptation (LORA) [8] and chain of thought (CoT) [9] can unleash the potential of LLMs in the specialized domain. These methods help LLMs improve performance in VTT mapping by fusing more VTT-related information such as that from Common Weakness Enumeration (CWE) [10] and Common Attack Patterns Enumeration and Classification (CAPEC) [11]. CWE is a list of software and hardware vulnerability types, while CAPEC catalogs common attack patterns that exploit these types of vulnerabilities. Specifically, many of the attack patterns documented in CAPEC correspond to tactics and techniques outlined in ATT&CK.

In this paper, to automate VTT mapping, we propose a two-step framework called VTT-LLM, which aims to realize the potential of LLMs in VTT mapping, as shown in Figure 2. Firstly, to leverage the LLM, we extract and craft a chain template to organize instructions from four cybersecurity databases, namely, CVE, CWE, CAPEC, and ATT&CK. These instructions are used to fine-tune the LLM to the specialized domain of the VTT mapping to complete the mapping from vulnerability descriptions to attack technique descriptions. Secondly, we match the attack technique description with the ATT&CK technique description based on text embedding to complete the VTT task. Since the techniques of ATT&CK are implementations of their tactics, mapping vulnerabilities to techniques implies that the VTT task has been implemented. Our contributions are summarized as follows.

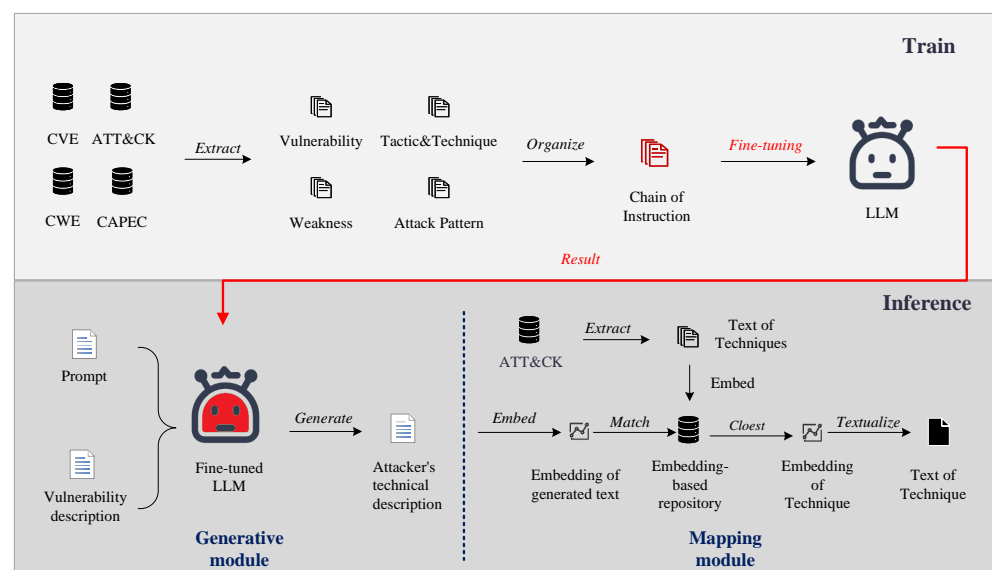


Figure 2. Workflow of VTT-LLM.

- We propose a framework called VTT-LLM, which automatically maps vulnerabilities in CVE to techniques in ATT&CK, including mapping to their tactics, thus enabling the mapping of vulnerabilities to attack categories as well as behavior patterns. VTT-LLM significantly improves the accuracy of VTT mapping by 13.69% over the traditional state-of-the-art method, CVET, and 54.42% over the native application of ChatGPT.
- To maximize the potential of LLMs in VTT mapping, we propose a novel chain template for generating fine-tuning instructions that combine a priori knowledge of CWE and CAPEC. The method yields an average accuracy increase of 9.24% across different LLMs.

- To support future research on VTT mapping and related tasks, we extract knowledge from CWE, CAPEC, and ATT&CK and compile a comprehensible instruction dataset for fine-tuning the LLM, whose effectiveness is promoted in the vertical domain. This dataset includes descriptions of vulnerabilities, weaknesses, attack patterns, and techniques, along with their mappings.

The organization of this paper is as follows. The related work is described in Section 2. This paper starts with a detailed introduction of the proposed VTT-LLM framework in Section 3 and verifies the proposed method through experiments and analyses in Section 4. Limitations are given in Section 5. Section 6 summarizes our research and discusses potential future directions. Our fine-tuned model, code, and dataset are available on request for the common research and use of the academic and industrial communities.

2. Related Work

Vulnerability to Tactics and Techniques: Vulnerability to tactics and techniques is important in vulnerability management and cyberhunting [12–15]. Few vulnerabilities can map to tactics and techniques based on explicit database information [2,11]. Automatically establishing a mapping from vulnerability to tactics and techniques commonly utilize a description of the vulnerability [5,6,16,17]. These automated methods enable language models to better extract semantics from descriptions of vulnerabilities and techniques, such as information summarization [6], model distillation [5], and model fine-tuning [16]. However, they overlook the benefit of introducing additional information beyond CVE and ATT&CK. VTT-LLM introduces more information including descriptions of weaknesses and attack patterns.

Large language model: General-purpose LLMs exhibit comparatively limited performance when solving complex problems within the professional domain such as the VTT task [17]. To make LLMs adapt to a professional domain, fine-tuning methods have been proposed [8,18–20]. To improve the ability of LLMs to solve complex problems, researchers transform complex problems into multi-step problems, employing techniques like generating intermediate steps or adopting the chain-of-thought (CoT) approach [21,22]. Moreover, fine-tuning with chain-of-thought (CoT) data would be a more effective approach for addressing complex problems in professional domains [9,23–26]. Due to the need for CoT data, these CoT-related fine-tuning methods require the implementation of emerging CoT capabilities from fairly large models like GPT-3 (175B) [25]. In contrast, on the VTT task, we handcrafted a chain template to obtain CoT data, which allowed us to achieve improved results on relatively small LLMs.

3. Method: VTT-LLM Framework

To achieve the goal of VTT mapping, we propose the VTT-LLM framework consisting of a generative module and a mapping module. It should be noted that a vulnerability may correspond to more than one tactic and technique in practice, and our approach aims to identify a correct tactic and technique. The generation module generates the corresponding attack technology description based on the vulnerability description. The mapping module then uses the semantic embedding matching method to map the generated content to the appropriate technology categories. The standard embedding library is built based on the name and description of the category in ATT&CK. In Section 3.1, we use a concrete example to illustrate how we differ from previous LLM-based studies and elicit our insight. Then, we provide detailed introductions of two modules in Sections 3.2 and 3.3, respectively.

3.1. Motivation

As shown in Figure 3, previous studies on building a VTT mapping focused on mapping detailed vulnerability descriptions to concise categories, often ignoring or making only limited use of the descriptions of the categories. However, with the introduction of powerful LLMs, there is now the potential to perform more sophisticated long-to-long mappings that capitalize on the categories' descriptions. Intuitively, allowing the model

to see more correct semantic mappings can help improve the accuracy of the VTT task. Furthermore, this inspired us to let the LLM fuse more VTT-related information to improve the effectiveness of the VTT task.

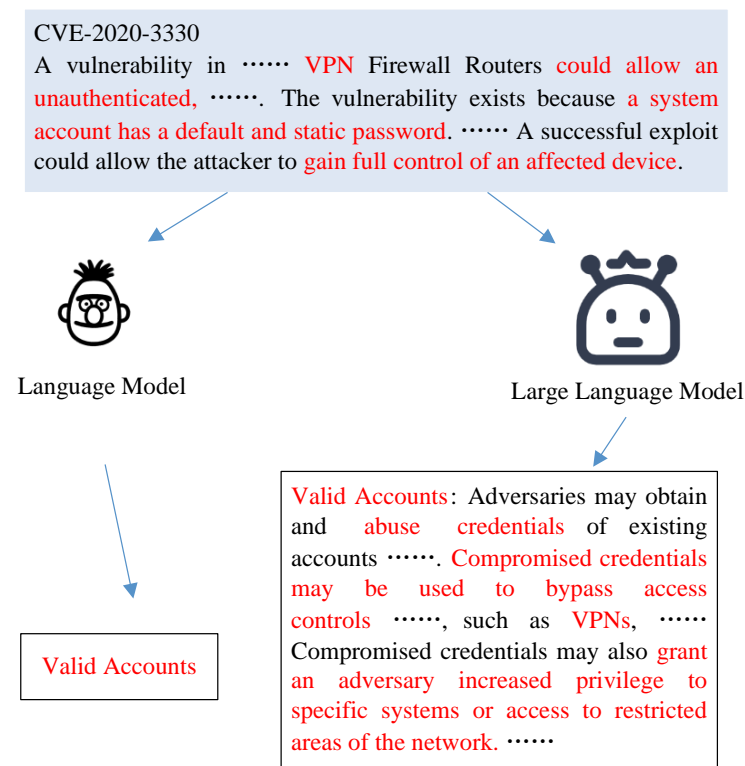


Figure 3. Comparison of our LLM method with previous approaches, illustrating key differences. The red color represents the semantic information that the model can learn to map.

3.2. Generative Module

The generation module is the core component of VTT-LLM. The LLM learns grammar, semantics, and contextual information during the pre-training phase. Consequently, it can fully leverage the semantic information within categories. Moreover, fine-tuning with appropriate instructions and data can unlock the potential of the LLM for VTT mapping. We devised three methods to organize data for fine-tuning instructions to prove the superiority of chain templates.

(a) No template

The primary objective of fine-tuning an LLM is to align the model with the specific requirements of the task at hand and the data distribution characteristics of the dataset. Intuitively, we can enhance the performance of the LLM on the VTT task through supervised fine-tuning. This is achieved by organizing instructions and data in a structured manner, as demonstrated in Table 1. The fine-tuning method without a template can be formulated as F_{VT} .

Table 1. “Instructions” and “Inputs” serve as inputs for training, while “Label” provides supervision. Bold within the curly braces represent the variables in the template.

Key	Value
Instruction	Tell me which attack technique is most likely to use the vulnerability
Input	Vulnerability description: { vulnerability description }
Label	{ ATT&CK technique name }: { technique description }

(b) Intermediate steps' template

The direct mapping from vulnerabilities to attacker techniques exhibits a stark transition, potentially impeding the inferential capabilities of the LLM. To address this issue, we propose the intermediate steps' template to create a smoother mapping relationship that incorporates two additional intermediary stages. They are weakness and attack patterns. Weakness come from CWE, which is a community-developed list of common software and hardware weakness types that have security ramifications. Attack pattern is a concept in CAPEC, which is a comprehensive dictionary of known attack patterns that an adversary use to exploit weaknesses. We define a series of functions to articulate the steps' template.

- Step 1:

$$F_{VW} : \text{Vulnerabilities} \rightarrow \text{Weaknesses}$$

This function identifies the specific nature and characteristics within the vulnerability description, mapping them to an appropriate weakness type.

- Step 2:

$$F_{WA} : \text{Weaknesses} \rightarrow \text{Attack Patterns}$$

This function translates identified weaknesses into specific attack patterns that adversaries might employ.

- Step 3:

$$F_{AT} : \text{Attack Patterns} \rightarrow \text{ATT\&CK Techniques}$$

This function connects attack patterns to precise tactic and techniques within the ATT&CK framework.

- Step 4:

$$F_{VT} : \text{Vulnerabilities} \rightarrow \text{ATT\&CK Techniques}$$

This function is to inform the LLM about which tasks to map first and implicitly connect the fine-tuning data knowledge injected into the LLM in steps 1, 2, and 3.

Based on the above function definition, the intermediate steps' template can be formulated as:

$$F_{step} = F_{VW} \circ F_{WA} \circ F_{AT} \circ F_{VT} \quad (1)$$

This function represents the composition of the preceding functions, where \circ symbolizes the sequential composition of these functions, creating a smooth transition from identifying vulnerabilities to determining the corresponding ATT&CK technique. The instructions for these functions are outlined in Table 2

Table 2. Intermediate steps' fine-tuning instructions

Type	Instruction
F_{VW}	Identify weaknesses based on vulnerability descriptions
F_{WA}	Tell me possible attack patterns based on weaknesses
F_{AT}	Identify attack technique based on the attack pattern
F_{VT}	Tell me which attack technique is most likely to use the vulnerability

(c) Chains' template

The intermediate steps' template only implicitly captures the relationship between vulnerabilities and attacker techniques. To more effectively leverage the reasoning capabilities of the LLM, utilizing explicit expressions of these relationships could prove more advantageous. The chains' data structure is shown in Table 3, which can be formulated as F_{VWAT} .

Table 3. Instruction data structures for fine-tuning when CWE and CAPEC data are explicitly added. Bold within the curly braces represent the variables in the template.

Key	Value
Instruction	Tell me which attack technique is most likely to use the vulnerability
Input	Vulnerability description: { vulnerability description }
Label	Let us think step by step. According to the vulnerability description, its corresponding weakness is { Weakness }. According to the weakness, its corresponding attack pattern is { Attack Pattern }. According to the attack pattern, its corresponding attack technique is { Technique }.

In summary, during training, the LLM undergoes fine-tuning using data organized through the three mentioned methods. This results in the development of a fine-tuned LLM. Consequently, in the inference stage, by inputting instructions and data, the fine-tuned LLM can generate a description of the attacker’s action.

3.3. Mapping Module

The mapping module serves as the auxiliary component of VTT-LLM. This module utilizes a pre-trained semantic embedding model to embed generated text. It then identifies the most similar category with the cosine similarity in the category-embedding library, which contains vectorized technique descriptions from the ATT&CK framework. Then, the module converts the embedding into its corresponding technique in textual form based on an index. By executing these steps, the module effectively matches the generated content with ATT&CK techniques based on their shared technical features of attackers. This process enhances the flexibility of VTT mapping by allowing for the modification and review of both generated text and categories’ definitions. This dynamic approach to technique category management is crucial for adapting to the evolving ATT&CK framework over time.

4. Evaluation

4.1. Setup

4.1.1. Environment

We performed fine-tuning and evaluation tasks on a single GeForce RTX 4090 GPU. We utilized the default settings of efficient fine-tuning tools [27] to fine-tune llama-7B [28]. For semantic matching, we use a pre-trained embedding model, bge-large-en-v1.5 [29], to map text to a low-dimensional dense embedding and build embedding-based repository using faiss [30].

4.1.2. Dataset

In our study, we used two datasets. We constructed Dataset 1 to obtain high-quality data for fine-tuning, while Dataset 2 was used for both fine-tuning and evaluation due to its abundant quantity of samples.

Dataset 1: This dataset was extracted from CWE, CAPEC, and ATT&CK. It encompassed the name and description of vulnerabilities, weaknesses, attack patterns, and techniques, as well as the mapping between them. In detail, we extracted weakness descriptions and corresponding vulnerability descriptions from CWE. We also extracted attack pattern descriptions and the weaknesses and techniques associated with each attack pattern from CAPEC. Furthermore, we extracted technology descriptions and their corresponding tactics from ATT&CK. Finally, these mappings were combined as described in method (a) [No template](#), (b) [Intermediate steps’ template](#), (c) [Chains’ template](#) to form an instruction dataset. The number of entries is shown in Table 4.

Table 4. Use of the functions defined in the method to express the quantity of data in the dataset and the mapping between them.

Fuction Type	Number
F_{VW} (data)	1931 \rightarrow 445
F_{WA} (data)	445 \rightarrow 128
F_{AT} (data)	128 \rightarrow 188
F_{VT} (data)	598 \rightarrow 35
F_{VWAT} (data)	598 \rightarrow 35

Dataset 2: This dataset [17] was derived from prior research (Liu et al. [7]) and is named ChatGPT-VDMEval, which served as the benchmark dataset, facilitating direct comparisons. The dataset was constructed based on BRON [14], and it included 25439 CVE vulnerability descriptions and corresponding ATT&CK techniques.

4.1.3. Evaluation Metrics

In our evaluation, unless specified otherwise, we utilized the most recent year of data from Dataset 2 as the evaluation dataset, i.e., the vulnerability mapping data for the year 2021. The evaluation metrics included two components: technique accuracy and tactic accuracy. A vulnerability can be associated with multiple techniques, and as long as VTT-LLM could accurately identify any of them, it was considered the correct output and included in the accuracy statistics. We examined the effectiveness of VTT-LLM by addressing the following research questions (RQs):

- RQ1: Can VTT-LLM effectively map a vulnerability to a tactic and a technique? How does it compare with previous methods?
- RQ2: Does the chain template perform best?
- RQ3: Which kind of chain structure performs best?
- RQ4: Is VTT-LLM still effective on different LLMs and different years of vulnerability?

4.2. Results and Analysis

To answer these research questions, we conducted experiments on the VTT-LLM on different fine-tuned data organization methods, different years of evaluation data, and different LLMs.

4.2.1. RQ1: Comparison with Previous Research

To compare with previous studies [5,7], we utilized F_{VWAT} (data) from Dataset 1 for fine-tuning and the entire Dataset 2 for evaluation, which included data from 1999 to 2021. As shown in Table 5, it can be observed that VTT-LLM exhibited superior performance compared to existing methods.

Table 5. The tactical accuracy of previous studies in the table is cited from Liu et al. [7].

Type	Method	Tactic
Classical machine learning	Random forest	37.67%
	SVM	46.34%
Fine-tuned models	GPT-2	64.56%
	BERT	69.41%
Self-distillation	CVET	71.49%
ChatGPT	Prompt	32.76%
Fine-tuning	VTT-LLM (chain)	85.18 %

4.2.2. RQ2: Effect of Template Type

We used Dataset 1 and Dataset 2 to complete our method as follows:

- As described in method (a) **No template**, our fine-tuning data were derived from two sources: F_{VT} (data) from Dataset 1, part of 2020 and 2019 CVE data, and their mapping technique (2993 pieces) from Dataset 2.
- According to the method (b) **Intermediate steps' template**, we added F_{VW} (data), F_{WA} (data), and F_{AT} (data) from Dataset 1 to implement F_{step} .
- Following the (c) **Chains' template** method, we used F_{VWAT} (data) from Dataset 1 for fine-tuning.

The result of employing the three methods is shown in Table 6. To illustrate clearly, we give an example in Appendix A.

Table 6. The impact of different template types on the accuracy of mapping to techniques and tactics.

Method	Technique	Tactic
No template	41.47%	67.85%
Intermediate steps	38.05%	77.08 %
Chain	45.87%	76.56 %

Based on the information provided in Table 6, it is evident that the chains' template achieved optimal performance in the accuracy of the mapped technique compared to the other two methods. Although it slightly lagged behind the intermediate steps' fine-tuning in the accuracy of the mapped tactic, the difference was marginal. It proved the superiority of explicitly expressing relations. We also noted that the intermediate steps achieved high performance in terms of tactical accuracy but was less effective in terms of technical accuracy. This may be because implicit relational expressions map vulnerability descriptions to similar but different techniques that implement the same tactic.

4.2.3. RQ3: Chain Structure

To study what type of chain is best in fine-tuning, we designed three types of organized chain data to fine-tune the LLM. They were F_{VWAT} , F_{VAT} , and F_{VWT} . The data organization form of F_{VWAT} is as shown in Table 3. Different from F_{VWAT} , F_{VAT} lacks weakness as a transition and maps directly from a vulnerability to an attack pattern, and F_{VWT} lacks an attack pattern. The result is shown in Table 7.

Table 7. The impact of different chain organization methods on accuracy

Method	Techniques	Tactics
F_{VWAT}	45.87%	76.56%
F_{VAT}	39.60 %	76.3 %
F_{VWT}	44.73 %	75.32%

It can be seen that F_{VWAT} had the highest accuracy rate in techniques and tactics' prediction, proving the necessity of introducing CWE and CAPEC.

4.2.4. RQ4: Robustness

We conducted experimental evaluations on vulnerabilities in different years and different LLMs, proving the robustness of VTT-LLM.

(a) Timely performance evaluation

We used the F_{VWAT} (data) from Dataset 1 to fine-tune llama and evaluate the vulnerability data of all years in Dataset 2 by year.

It can be seen in Figure 4 that our method had relatively stable and excellent performance in different years. Regarding the overall trend, there was an observed decrease in the accuracy from 2019 to 2021. This could be attributed to the evolving and diverse tactics employed by attackers in recent years. Security experts have responded by developing more refined models of attacker behaviors, thereby increasing the difficulty of classification.

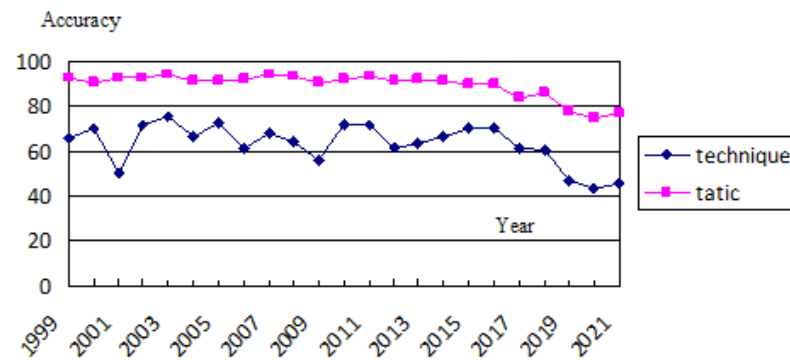


Figure 4. Years of evaluation.

(b) Different types of LLM

We tested on llama-7B, baichuan-7B, bloom-7b1, and bloomz-7b1 models with little difference in parameters to eliminate the impact of scale and verify the effectiveness and stability of the fine-tuning method in VTT-LLM. Simultaneously, we conducted a comparative experiment with Flan-T5-XXL(11B), which employs a chain-of-thought fine-tuning, thereby demonstrating robust inferential capabilities across multiple tasks. However, the accuracy of the VTT tasks was lower than when using the chain template we designed for fine-tuning. The result is shown in Figure 5.

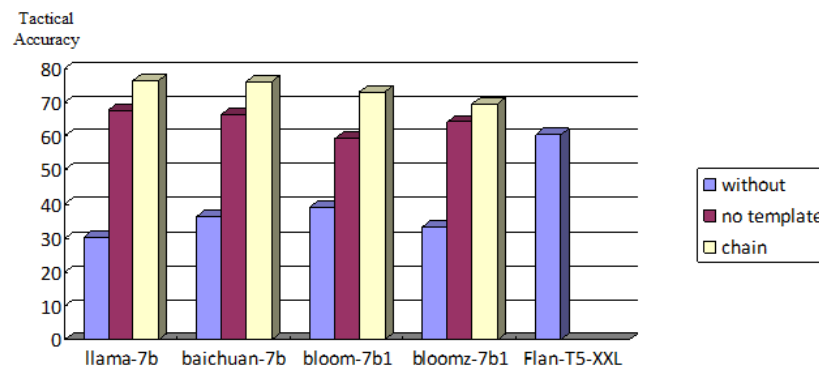


Figure 5. Tactical accuracy of different LLMs in VTT-LLM. “without” represents LLM without fine-tuning. “no template” represents LLM employing (a) No template. “chain” represents an LLM employing (c) Chains’ template.

Accordingly, VTT-LLM demonstrated stable performance on different LLMs. Compared to the Flan method which has the powerful ability to multitask, VTT-LLM fine-tuned with F_{VWAT} (data) performed better in solving the VTT problem. On these LLMs, the chains’ template method continued to outperform the method with no template, exhibiting an average improvement of 9.24%.

(c) Different scales of LLMs

To explore the impact of scale on our method, and thanks to Bloom’s multi-scale models, we chose to perform comparisons on Bloom-560m, Bloom-1b1, Bloom-1b7, Bloom-3b, and Bloom-7b1, employing the chains’ template method.

From Table 8, it is evident that our method demonstrated the best tactical accuracy at the scale of 7b1, but even at the scale of 1b1, our method still performed well. This lowers the threshold for adopting our approach.

Table 8. Effect of different scales. “m” represents a model scale with millions of parameters. “b” represents billions of parameters.

Scale	Techniques	Tactics
bloom-560m	27.21 %	60.8%
bloom-1b1	38.25 %	69.36 %
bloom-1b7	36.28 %	71.85%
bloom-3b	31.67 %	67.8%
bloom-7b1	35.51 %	73.14%

5. Limitations

We illustrate this paper’s limitations from the following two aspects: Limited by computational resources, we restricted the application of our method to models below 7.1 billion (7b1) parameters. Larger LLMs are left for future work to test. Due to dataset limitations, the most recent year for the public dataset was 2021, and it was not possible to determine the impact of the last three years on the proposed model. Further, although the number of datasets used for fine-tuning differed significantly from those used for evaluation (598 vs. 25,439), they still intersected, which may have affected the model’s performance in a small way.

6. Conclusions and Future Work

In this paper, we proposed VTT-LLM to establish a mapping between CVE and ATT&CK by using a pre-generation and post-classification framework. By incorporating the core concepts (weakness, attack pattern) and existing mapping relationships in CWE and CAPEC, we improved the model’s prediction performance. In the future, regarding chain design, we aim to explore the use of graph neural network learning or reinforcement learning methods to identify potentially superior chains, moving away from manual design. In terms of data expansion for fine-tuning, we plan to incorporate unstructured threat reports into the chain to take full advantage of LLMs’ advanced capabilities.

Author Contributions: Conceptualization, C.Z. and Z.L.; methodology, C.Z.; software, C.Z. and T.Z.; validation, C.Z. and J.Z.; formal analysis, L.Z.; investigation, J.Z. and T.Z.; resources, L.W.; data curation, C.Z.; writing—original draft preparation, C.Z. and L.Z.; writing—review and editing, Z.L., L.W. and L.Z.; supervision, D.F.; project administration, D.F.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Guangdong Basic and Applied Basic Research Foundation (2023A1515011698), Guangdong High-level University Foundation Program (SL2022A03J00918), Major Key Project of PCL (PCL2022A03), and National Natural Science Foundation of China (Grant No. 62372137).

Data Availability Statement: The data presented in this study are available on request.

Conflicts of Interest: Author Dunqiu Fan was employed by the company NSFOCUS Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

CVE-2021-44524 can be mapped to techniques such as “T1505”, “T1134”, “T1557”, “T1550” in ChatGPT-VDMEval. The vulnerability description of CVE-2021-44524 is “A vulnerability has been identified in SiPass integrated V2.76 (All versions), SiPass integrated V2.80 (All versions), SiPass integrated V2.85 (All versions), Siveillance Identity V1.5 (All versions), Siveillance Identity V1.6 (All versions < V1.6.284.0). Affected applications insufficiently limit the access to the internal user authentication service. This could allow an unauthenticated remote attacker to trigger several actions on behalf of valid user accounts”. The instruction is “Tell me which attack technique is most likely to use the vulnerability”.

Input is a combination of instructions and description. The output of the different methods is shown in Table A1.

Table A1. Output example. Bold within the curly braces represent the variables in the template.

Method	Output Description	Output Category
No template	Impair defenses: { description }	T1562
Intermediate steps	Adversary-in-the-middle: { description }	T1557
chain	Let us think step by step. According to the vulnerability description, its corresponding weakness is Improper Authorization; according to the weakness description, its corresponding attack pattern is Accessing Functionality Not Properly Constrained by ACLs; according to the attack pattern, its corresponding attack technique is Services File Permissions Weakness. Therefore, the final answer is Services File Permissions Weakness:{ description }.	T1574

In this example, intermediate steps obtain the correct answers when evaluated with ChatGPT-VDMEval.

References

1. CVE. Common Vulnerabilities and Exposures. 2023. Available online: <https://www.cve.org/> (accessed on 15 August 2023.).
2. Strom, B.E.; Applebaum, A.; Miller, D.P.; Nickels, K.C.; Pennington, A.G.; Thomas, C.B. Mitre att&ck: Design and philosophy. In *Technical Report*; The MITRE Corporation: Bedford, MA, USA, 2018.
3. Zhang, Z.; Wang, L.; Chen, G.; Gu, Z.; Tian, Z.; Du, X.; Guizani, M. STG2P: A two-stage pipeline model for intrusion detection based on improved LightGBM and K-means. *Simul. Model. Pract. Theory* **2022**, *120*, 102614. [[CrossRef](#)]
4. Kaloroumakis, P.E.; Smith, M.J. *Toward a Knowledge Graph of Cybersecurity Countermeasures*; The MITRE Corporation: Bedford, MA, USA, 2021.
5. Benjamin Ampel, Sagar Samtani, S.U.; Chen, H. Linking Common Vulnerabilities and Exposures to the MITRE ATT&CK Framework: A Self-Distillation Approach. In Proceedings of the 2021 ACM Conference Knowledge Discovery and Data Mining (KDD'21) Workshop on AI-enabled Cybersecurity Analytics, Online, 14–18 August 2021; pp. 1–5.
6. Abdeen, B.; Al-Shaer, E.; Singhal, A.; Khan, L.; Hamlen, K. SMET: Semantic Mapping of CVE to ATT&CK and Its Application to Cybersecurity. In Proceedings of the Data and Applications Security and Privacy XXXVII, Sophia-Antipolis, France, 19–21 July 2023 ; Atluri, V.; Ferrara, A.L., Eds.; Springer: Cham, Switzerland, 2023; pp. 243–260.
7. Liu, X.; Tan, Y.; Xiao, Z.; Zhuge, J.; Zhou, R. Not The End of Story: An Evaluation of ChatGPT-Driven Vulnerability Description Mappings. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 3724–3731. [[CrossRef](#)]
8. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the International Conference on Learning Representations, Virtual, 25 April 2022.
9. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv* **2023**, arXiv:2201.11903.
10. MITRE. Common Weakness Enumeration. 2023. Available online: <https://cwe.mitre.org/> (accessed on 20 August 2023).
11. MITRE. Common Attack Pattern Enumerations and Classifications. 2023. Available online: <https://capec.mitre.org/> (accessed on 15 August 2023).
12. Nair, A.; Ray, A.; Reddy, L.; Marali, M. Mapping of CVE-ID to Tactic for Comprehensive Vulnerability Management of ICS. In Proceedings of the Inventive Communication and Computational Technologies, Online, 22–23 May 2023; Ranganathan, G., Fernando, X., Rocha, Á., Eds.; Springer: Singapore, 2023; pp. 559–571.
13. Upadhyay, D.; Sampalli, S. SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations. *Comput. Secur.* **2020**, *89*, 101666. [[CrossRef](#)]
14. Hemberg, E.; Kelly, J.; Shlapentokh-Rothman, M.; Reinstadler, B.; Xu, K.; Rutar, N.; O'Reilly, U.M. Linking Threat Tactics, Techniques, and Patterns with Defensive Weaknesses, Vulnerabilities and Affected Platform Configurations for Cyber Hunting. *arXiv* **2021**, arXiv:2010.00533.
15. Santos, J.C.; Tarrit, K.; Sejfia, A.; Mirakhorli, M.; Galster, M. An empirical study of tactical vulnerabilities. *J. Syst. Softw.* **2019**, *149*, 263–284. [[CrossRef](#)]

16. Grigorescu, O.; Nica, A.; Dascalu, M.; Rughinis, R. CVE2ATT&CK: BERT-Based Mapping of CVEs to MITRE ATT&CK Techniques. *Algorithms* **2022**, *15*, 314. [CrossRef]
17. Dstsmallbird. ChatGPT-VDMEval. 2023. Available online: <https://github.com/dstsmallbird/ChatGPT-VDMEval> (accessed on 22 October 2023).
18. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. *arXiv* **2021**, arXiv:2101.00190,
19. Liu, X.; Ji, K.; Fu, Y.; Du, Z.; Yang, Z.; Tang, J. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. *arXiv* **2021**, arXiv:2110.07602,
20. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv* **2023**, arXiv:2305.14314.
21. Nye, M.; Andreassen, A.; Gur-Ari, G.; Michalewski, H.; Austin, J.; Bieber, D.; Dohan, D.; Lewkowycz, A.; Bosma, M.; Luan, D.; et al. Show Your Work: Scratchpads for Intermediate Computation with Language Models. *arXiv* **2021**, arXiv:2112.00114.
22. Kojima, T.; Gu, S.S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large Language Models are Zero-Shot Reasoners. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; Curran Associates, Inc.: Glasgow, UK, 2022; Volume 35, pp. 22199–22213.
23. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling Instruction-Finetuned Language Models. *arXiv* **2022**, arXiv:2210.11416.
24. Li, L.H.; Hessel, J.; Yu, Y.; Ren, X.; Chang, K.W.; Choi, Y. Symbolic Chain-of-Thought Distillation: Small Models Can Also “Think” Step-by-Step. *arXiv* **2023**, arXiv:2306.14050.
25. Ho, N.; Schmid, L.; Yun, S.Y. Large Language Models Are Reasoning Teachers. *arXiv* **2023**, arXiv:2212.10071.
26. Zhang, Z.; Zhang, A.; Li, M.; Smola, A. Automatic Chain of Thought Prompting in Large Language Models. *arXiv* **2022**, arXiv:2210.03493.
27. hiyouga. LLaMA Efficient Tuning. 2023. Available online: <https://github.com/hiyouga/LLaMA-Efficient-Tuning> (accessed on 18 October 2023).
28. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971
29. Xiao, S.; Liu, Z.; Zhang, P.; Muennighoff, N. C-Pack: Packaged Resources To Advance General Chinese Embedding. *arXiv* **2023**, arXiv:2309.07597.
30. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* **2019**, *7*, 535–547. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.