SMET: Semantic Mapping of CVE to ATT&CK and its Application to Cybersecurity

Basel Abdeen[1], Ehab Al-Shaer[2], Anoop Singhal[3], Latifur Khan[1], and Kevin Hamlen[1]

[1] University of Texas at Dallas, TX, USA
[2] Carnegie Mellon University, PA, USA
[3] National Institute of Standards and Technology, MD, USA

Abstract. Cybercriminals relentlessly pursue vulnerabilities across cyberspace to exploit software, threatening the security of individuals, organizations, and governments. Although security teams strive to establish defense measures to thwart attackers, the complexity of cyber defense and the magnitude of existing threats exceed the capacity of defenders. Therefore, MITRE took the initiative and introduced multiple frameworks to facilitate the sharing of vital knowledge about vulnerabilities, attacks, and defense information. The Common Vulnerabilities and Exposures (CVE) program and ATT&CK Matrix are two significant MITRE endeavors. CVE facilitates the sharing of publicly discovered vulnerabilities, while ATT&CK collects and categorizes adversaries' Tactics, Techniques, and Procedures (TTP) and recommends appropriate countermeasures.

As CVE yields a low-level description of the vulnerability, ATT&CK can complement it by providing more insights into that vulnerability from an attacking perspective, thereby aiding defenders in countering exploitation attempts. Unfortunately, due to the complexity of this mapping and the rapid growth of these frameworks, mapping CVE to ATT&CK is a daunting and time-intensive undertaking. Multiple studies have proposed models that automatically achieve this mapping. However, due to their reliance on annotated datasets, these models exhibit limitations in quality and coverage and fail to justify their decisions. To overcome these challenges, we present SMET — a tool that automatically maps CVE entries to ATT&CK techniques based on their textual similarity. SMET achieves this mapping by leveraging ATT&CK BERT, a model that we trained using the SIAMESE network to learn semantic similarity among attack actions. In inference, SMET utilizes semantic extraction, ATT&CK BERT, and a logistic regression model to map CVE entries to ATT&CK techniques. As a result, SMET has demonstrated superior performance compared to other state-of-the-art models.

1 Introduction

The amount, robustness, and impact of cyber attacks have significantly increased in the past few years. The cost of cybercrime was estimated to be around $8.4

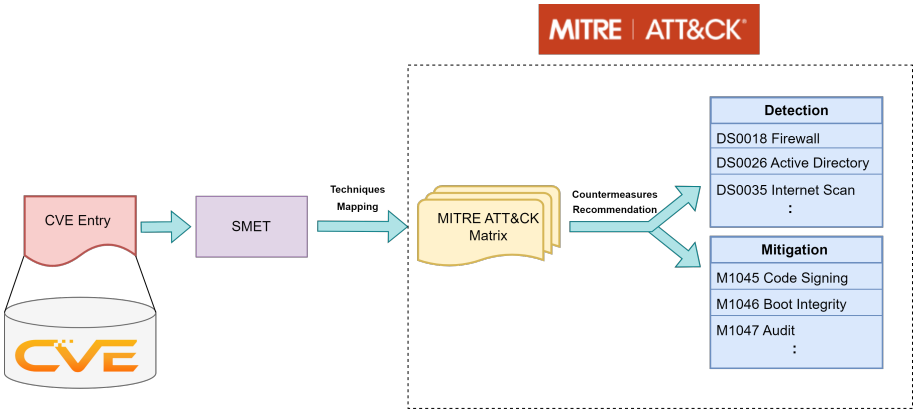

Fig. 1. CVE to ATT&CK mapping importance

trillion globally in 2022, and it is predicted to reach $23.84 trillion annually by 2027 [14]. As organizations and governments strive to combat cybercrime, their efforts are being met with increasingly complex attacks and hindered by the prevalence of software vulnerabilities. A notable example of such an attack is WannaCry, a malware that leveraged a vulnerability in the Windows system to spread globally and cause an estimated financial loss of $4 billion [12].

A vulnerability is a software design or implementation flaw that attackers can exploit to adversely impact confidentiality, integrity, or availability. Efforts have been made to counter vulnerability threat, primarily by identifying vulnerabilities in public software and sharing critical information about them. MITRE has led these efforts and introduced a list of publicly disclosed vulnerabilities known as Common Vulnerabilities and Exposures (CVE). Each entry of CVE consists of a unique ID number and a description of a discovered vulnerability alongside other critical information. Furthermore, the National Institute of Standards and Technology (NIST) introduced the National Vulnerability Database (NVD), which enriches CVE entries with more information, such as an estimation of the vulnerability's severity and classification of the vulnerability type. CVE is leveraged by organizations to monitor newly discovered vulnerabilities and ensure the security of their systems and networks.

In addition to CVE, multiple comprehensive knowledge bases and ontologies have been constructed by various organizations to aid security analysts in understanding and categorizing information regarding attackers' behaviors and objectives. ATT&CK is another knowledge base developed by MITRE that categorizes attack techniques that adversaries have been observed using in the real world. ATT&CK categorizes attack techniques into different tactics, where each tactic represents a goal that the attacker tries to achieve, such as initial access, execution, or defense evasion. A tactic contains multiple techniques that an attacker can each of which includes a textual description of the attacker's behaviors and examples of real-world uses of the technique by known malicious

groups and software. Moreover, each technique contains detection and mitigation recommendations to aid security teams in countering the technique. ATT&CK's high-level attack categorization and practical defense recommendation made it a useful resource for security analysts to secure their systems and networks.

As CVE and ATT&CK offer distinct perspectives on threat information, linking them enriches both frameworks and enables defenders to gain more insight into vulnerabilities and leverage information from ATT&CK to set their defense measures accordingly, as shown in Figure 1. However, achieving this link is a difficult task that requires experts with a thorough understanding of both frameworks. Moreover, as the size of CVE grows daily, manually mapping all of its entries to techniques becomes an infeasible task. Previous studies have tackled this problem and introduced machine learning models that automatically map a CVE entry to an ATT&CK technique [20, 16, 6, 23]. In these studies, researchers utilized an annotated dataset of CVE-ATT&CK mapping and trained a text classification model to classify entries to a limited number of techniques using the entry's description. However, the performance and coverage of these models are constrained by the annotated dataset that they rely on and cannot adapt to the dynamic nature of these frameworks. Moreover, these approaches lack explainability due to the black-box nature of deep learning models.

Unlike previous studies, SMET maps CVE to ATT&CK techniques without relying on any annotated dataset. SMET maps a CVE entry to techniques by first leveraging a semantic role labeling (SRL) model to extract attack vectors from the entry description. Attack vectors are textual descriptions of malicious actions that an attacker can perform. Some examples of attack vectors include "exploit a vulnerability to gain access to a network," "execute code on a victim machine," or "send data to a C2 server." SMET then extracts the embedding of these attack vectors using our developed ATT&CK BERT model. ATT&CK BERT is a transformer model that we fine-tuned using the SIAMESE network over ATT&CK Matrix data. As a result, ATT&CK BERT extracts a semantically meaningful embedding of attack vectors. Thus, extracted embeddings of similar attack vectors are close in the embedding space. Finally, SMET uses a logistic regression model trained using the ATT&CK Matrix to estimate the probability of an attack vector belonging to each ATT&CK technique and rank techniques based on the estimated probability. SMET[4] is publicly available on GitHub.

In this paper, we made the following contribution. First, we studied the relationship between CVE and ATT&CK and identified criteria to map any CVE entry to a corresponding ATT&CK technique. Second, we introduced a dataset that we manually annotated based on our criteria. We only used this dataset for evaluation purposes. Third, we developed SMET, a tool that maps any CVE entry to ATT&CK techniques. This paper is organized as follows. Section 2 presents the motivation and the problem statement. Section 3 presents previous related work.Section 4 introduces SMET. Section 5 evaluates SMET. Section 6 contains the conclusion and future work.

[4] https://github.com/basel-a/SMET.git

Mitigations

ID	Mitigation	Description
M1048	Application Isolation and Sandboxing	Application isolation will limit what other processes and system features the exploited target can access.
M1050	Exploit Protection	Web Application Firewalls may be used to limit exposure of applications to prevent exploit traffic from reaching the application.
M1030	Network Segmentation	Segment externally facing servers and services from the rest of the network with a DMZ or on separate hosting infrastructure.
M1026	Privileged Account Management	Use least privilege for service accounts will limit what permissions the exploited process gets on the rest of the system.
M1051	Update Software	Update software regularly by employing patch management for externally exposed applications.
M1016	Vulnerability Scanning	Regularly scan externally facing systems for vulnerabilities and establish procedures to rapidly patch systems when critical vulnerabilities are discovered through scanning and through public disclosure.[6]

Fig. 2. The exploitation for privilege escalation technique mitigation from ATT&CK

2 Motivation and Problem Statement

2.1 Motivation

Whenever a CVE entry for publicly used software is published, security analysts in organizations must be alerted and take appropriate measures to ensure the security of their system against the newly discovered vulnerability. Although the best defense against a vulnerability is applying its patch, patches take time to implement, during which, the software remains vulnerable, threatening organizations' systems and networks. Moreover, some vulnerabilities are never patched due to cost and complexity. Therefore, security analysts must take other measures to counter the vulnerability. Here is when ATT&CK proves helpful. When a vulnerability is linked to an ATT&CK technique, it can be studied from an attacker's perspective, as ATT&CK sheds insight into how an attacker can exploit the vulnerability, what an attacker can gain from the exploitation, and what mitigation and detection measures can be taken to counter the attacks. An example of mitigation measures recommended by ATT&CK to mitigate the exploitation for privilege escalation technique is shown in Figure 2.

Previous studies have proposed models that automatically map a CVE to an ATT&CK. In these approaches, experts first annotate a dataset by manually mapping a number of CVE entries to ATT&CK techniques. They then train a supervised machine learning model to link entries to techniques using the annotated data. The process of annotation is both time-consuming and labor-intensive, which leads to datasets that have limited quality and coverage. Consequently, models trained on such datasets suffer from performance limitations and are unable to accommodate the dynamic nature of both CVE and ATT&CK. In contrast, in this paper, we tackle the problem of mapping a CVE entry to an ATT&CK technique in an unsupervised manner by leveraging the textual description similarity between a CVE entry and an ATT&CK technique.

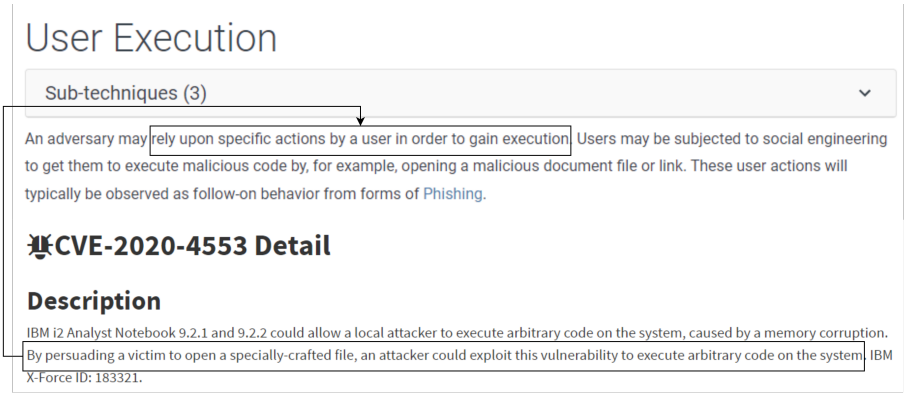

Fig. 3. Text similarity between the user execution technique description and a CV-2020-4553 description

2.2 Problem Statement

Our proposed approach achieves unsupervised mapping by leveraging text similarities between CVE and the descriptions of techniques. An example of similar text can be seen in Figure 3, where a CVE-2020-4553 description states that an attacker can exploit it by persuading the victim to open a crafted file. This attack behavior is represented in the user execution technique from ATT&CK. Identifying sentences' semantic similarities is a challenging and active area of research. Although researchers have introduced several deep learning models to extract sentence similarity, no research has investigated sentence similarity in the cybersecurity domain. The complexity of text similarity in cybersecurity stems from the lack of annotated data and the semantic gap between low-level and high-level attack vectors. For example, the sentence "An attacker runs a script in a machine" is semantically similar to "A malicious actor executes code on a system," as they represent a similar objective. However, the two sentences have no words in common. On the other hand, the two sentences "An attacker reads a file" and "An attacker deletes a file" share almost all words but are semantically different, as each attack vector corresponds to different attack objectives. Table 1 shows examples of semantically similar attack vectors.

2.3 CVE-ATT&CK Association Analysis

The description of an ATT&CK technique contains several attack vectors that the attacker performs to achieve the technique. These attack vectors can range from low-level behaviors, such as "create file" or "read registry," to high-level behaviors, such as "compromise system" or "steal information." Each ATT&CK technique consists of mainly two attack vectors: an action that an attacker takes and the objective of that action. For example, in the exploitation for client execution technique, an attacker exploits software vulnerability in client applications

Table 1. Semantically Similar Attack Vectors

Similar Attack Vectors
<Execute code, Run a script in the system >
<Read file, Retrieve the content of a file>
<Disable security tools, Delete Antivirus files >
<Identify the OS version, Collect system information >
<Crash the system, Cause machine termination >
<Take a screenshot, Capture the desktop screen >

(action) to execute code (objective). Actions are usually lower-level attack vectors, while objectives are higher-level attack goals and can represent the tactic of the attack. The action and objective can be identified for all techniques from the first sentence in the technique description.

ATT&CK techniques cover all stages of an attack life cycle, from reconnaissance and initial access to the compromising of confidentiality, integrity, and availability. Therefore, techniques can be linked to a CVE entry through various means. We investigated the CVE-ATT&CK association and identified two types of techniques for classifying a CVE entry that are inspired by [9]. First, a CVE entry can be mapped to techniques that describe an exploitation method, more specifically, a technique that an attacker needs to perform to exploit a vulnerability, such as exploiting a web browser vulnerability or tricking a user into performing an action. Second, CVEs can be mapped to techniques that describe the consequences of exploiting a vulnerability or the objectives that an attacker can accomplish after exploiting the vulnerability, such as code execution, privilege escalation, or data manipulation. In other words, one technique enables an attacker to exploit a vulnerability, and a vulnerability enables an attacker to achieve other techniques. We refer to these techniques as pre-exploit and post-exploit techniques, respectively.

We studied various ATT&CK techniques and identified techniques that correspond to each mapping type. Pre-exploit techniques are mostly part of the initial access tactic, such as drive-by compromise, exploit public-facing applications, user execution, and valid accounts. In the drive-by compromise technique, an attacker uses websites to exploit a vulnerability in users' web browsers. In the exploit public-facing application technique, an attacker exploits a website or any public-facing application, such as databases or standard services that use crafted input. In the user execution technique, the attacker depends on an action by the user to exploit a vulnerability. Finally, in the valid accounts technique, an attacker needs to compromise an account first in order to exploit the vulnerability.

Post-exploit techniques can be part of any tactic. The execution, privilege escalation, and credential access tactics all involve a technique of an adversary exploiting a vulnerability to achieve the tactic. The techniques are: exploitation for client execution, exploitation for privilege escalation, and privilege escalation for credential access, respectively. In addition, the lateral movement tactic

contains a technique — exploitation of remote services — where an attacker exploits a vulnerability in remote services to gain access to systems. Moreover, the impact tactic contains various techniques where an attacker aims to compromise the system or data's availability or integrity. One example is the application or system exploitation sub-technique, which describes the attack technique where an attacker exploits a vulnerability to compromise availability. CVE can also be mapped to other techniques that are not limited to vulnerability exploitation but describe the attacker's goals, such as data manipulation or system shutdown/reboot.In conclusion, the ATT&CK framework contains various techniques related to CVE. Some techniques focus specifically on vulnerability exploitation, while other general techniques can be achieved after exploitation.

3 Related Work

Since the ATT&CK Matrix became a vital resource for security analysts to understand and counter cyber threats, researchers have proposed multiple algorithms to automatically map various cyber information resources to ATT&CK techniques. For example, multiple studies have proposed models to map Cyber Threat Intelligence (CTI) reports to ATT&CK techniques [22, 21, 8, 18]. Other studies have proposed models to map CVE entries to ATT&CK techniques [20, 16, 6, 23]. Moreover, researchers have proposed models to map malware behavior, Linux shell commands, and threat data from smart grid systems to ATT&CK [25, 7, 19]. Other studies have focused on enriching CVE by automatically mapping it to the Common Weaknesses Enumeration (CWE) or Common Attack Pattern Enumeration and Classification (CAPEC) frameworks [10, 5]. Researchers have also proposed domain-specific language models for cybersecurity that can be applied to a wide range of downstream tasks [4, 2].

In one study, researchers studied the performance of various traditional machine learning models (e.g., Naive Bayes and SVC) and advanced deep learning models (e.g., CNN, BERT, SciBERT, and SecBERT) in linking CVE entries to the ATT&CK matrix using supervised learning [16]. They also studied the impact of data augmentation methods that help enrich the training set. They used the dataset introduced by MITRE Engenuity [13] for training and testing and enriched it with their own labeled data. In their evaluation, they only considered mapping to 31 ATT&CK techniques.

In another study, researchers proposed a neural network architecture — Multi-Head Joint Embedding Neural Network — that automatically maps CVE to ATT&CK techniques [20]. To create a training dataset, they introduced an unsupervised labeling technique with which they extracted CVE information from publicly available threat reports. Their labeling technique was able to map CVE entries to 17 ATT&CK techniques. Moreover, researchers have proposed CVET, a transformer-based model that has mapped CVE entries to 10 ATT&CK tactics [6]. CVET was trained using the self-knowledge distillation approach over the BRON dataset [17].

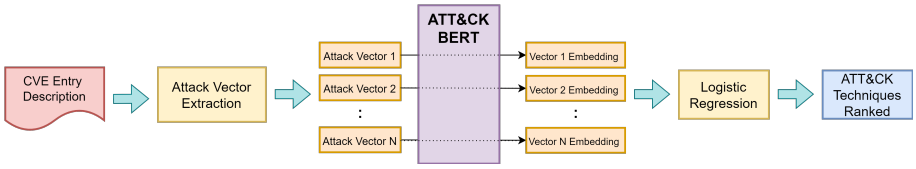

Fig. 4. SMET overview

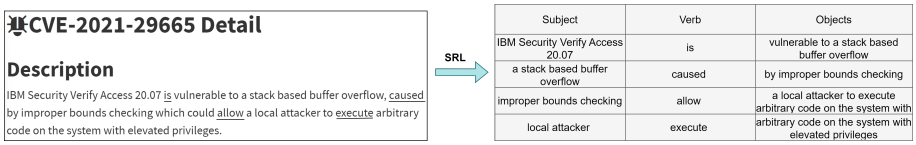

Fig. 5. Semantic role labeling (SRL) example

The aforementioned proposed models map CVE to a limited number of ATT&CK techniques due to their reliance on a manually annotated dataset. However, in a CTI mapping research, researchers introduced a tool — AttacKG — that maps CTI reports to all ATT&CK techniques [22]. AttacKG converts CTI reports and ATT&CK techniques into structured attack behavior graphs and maps reports by aligning their graphs to the techniques' template graphs. In the alignment phase, AttacKG uses character-level similarity to align nodes instead of semantic similarity. Unfortunately, character-level similarity is limited and cannot align text with similar meanings but different wording.

4 Approach

4.1 SMET Overview

SMET consists of three components: an attack vector extraction component, an attack vector representation model (ATT&CK BERT), and a logistic regression model. First, we used a semantic role labeling (SRL) model to extract attack vectors from a CVE entry description. The CVE entry description contains attack vectors alongside other vulnerability details that are irrelevant to our task. Therefore, we designed a few rules using SRL to identify attack vectors while discarding irrelevant information. Second, we proposed an attack vector embedding model — ATT&CK BERT — that extracts a semantic vector representation of attack vectors. Using ATT&CK BERT extraction, the extracted embedding of similar attack vectors are close in the embedding space. Thus, they have a high cosine similarity, while unrelated attack vectors have a low cosine similarity. Third, we used a logistic regression model to estimate the probability of CVE belonging to techniques and ranked all techniques by the estimated probability. Figure 4 shows an overview of the SMET architecture.

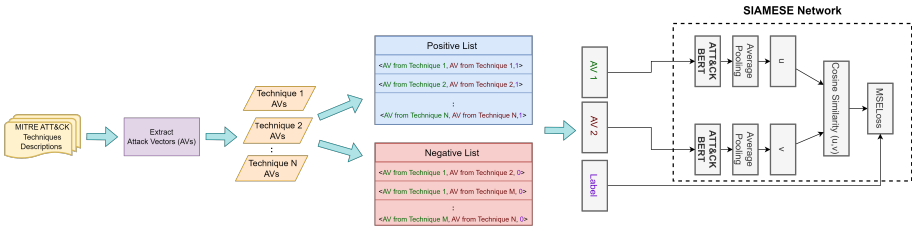

Fig. 6. ATT&CK BERT fine-tuning

4.2 Attack Vector Extraction

A CVE entry textual description contains different types of information regarding the CVE, including software name, affected versions, vulnerability type, vulnerability description, actions that an attacker can take to exploit the vulnerability, and objectives that an attacker can gain from the exploitation. Unfortunately, CVE entries do not follow a predefined structure, and not all of the information mentioned above exists for all CVE entries. Therefore, the attack vector extraction component aims to automatically identify attack vectors from the CVE description . To achieve this extraction, we leveraged an Allenlp SRL model, a state-of-the-art semantic extraction model that extracts semantic frames from unstructured text [15]. A semantic frame consists of a verb linked to its neighboring words/phrases that are tagged by their semantic relationship to the verb, such as a subject, patient, location, manner, or purpose. Figure 5 shows an example of semantic frames extracted from the CVE-2021-27032 description. For simplicity, we combined all roles other than subject and verb and assigned them an "objects" tag in the table.

Semantic frames can represent various types of information. For example, the first semantic frame in Figure 5 represents an attribute of the software using the verb "is," while the second semantic frame represents a causal relationship between "buffer overflow" (subject) and "improper bound checking" (objects) using the verb "caused." The last semantic frame represents an attack vector identified by the subject "local attacker." SMET considers a semantic frame to be an attack vector if one of these conditions is met. First, the words attacker, adversary, vulnerability, or user exist in the subject, as these subjects indicate that the frame contains an action that an attacker can do. Second, the verb is either allow, lead, or result. Our CVE analysis shows that these verbs precede an attack vector even if the subject is not an attacker. An example can be shown in the third semantic frame, where the verb is "allow," and the object is an attack vector.

4.3 ATT&CK BERT

Semantic similarity between sentences is a multifaceted concept that requires a clear definition based on the research goals at hand. In our study, we observed

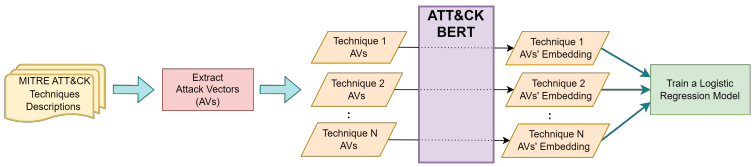

Fig. 7. Logistic regression training

that attack vectors can exhibit similarities in various aspects. For example, "create a file" and "create a registry key" are similar in the sense that both represent adding a storage unit in the system, while "create a file" and "delete a file" are similar in the sense that both apply an action to a file. Other dimensions of similarity can encompass the location of attack vectors (e.g., system or network) or privileges needed (e.g., administrator or user). Since we aim to map attack vectors to ATT&CK techniques, we consider two attack vectors similar only if they share the same objective and help achieve the same technique. For example, the three attack vectors "delete system file," "delete anti-virus file," and "delete log files" are different, although they all represent deleting a file from the system. "Delete system file "aims to interrupt the availability of the system; "delete anti-virus file" aims to disable defensive mechanisms; and "delete log files" aims to cover attack actions.

To extract semantic embedding, we propose ATT&CK BERT. ATT&CK BERT is a transformer model that aims to represent attack vectors in a semantically meaningful embedding where the embedding of attack vectors with similar meanings are close in the embedding space and thus, have high cosine similarity. Text embedding has been an active research challenge for a long time. The evolution of the transformer model, researchers have introduced multiple transformer-based sentence embedding models. One of the best-performing recent models is Sentence BERT (SBERT) [24]. SBERT introduced a method to train the BERT model using SIAMESE network architecture by taking two sentences as input, extracting each sentence embedding using BERT, and then optimizing the network weights to maximize the similarity of the two embeddings if the sentences are semantically similar and minimize it otherwise.

Since the original SBERT is trained on a general entitlement dataset, its performance on cybersecurity text is limited. However, preparing a dataset of pairs of sentences that cover all attack life cycle information and annotating them is an infeasible task. To overcome this challenge, we propose an approach to fine-tune SBERT using the ATT&CK framework as follows. First, we extract all attack vectors from each ATT&CK technique description and procedure examples using the Allenlp SRL model. Second, we create two lists that we denote as positive and negative lists. The positive list contains pairs of attack vectors that are extracted from the same technique, while the negative list contains pairs of attack vectors where each is extracted from a different technique. Finally, we train SBERT to maximize the cosine similarity between pairs from the positive

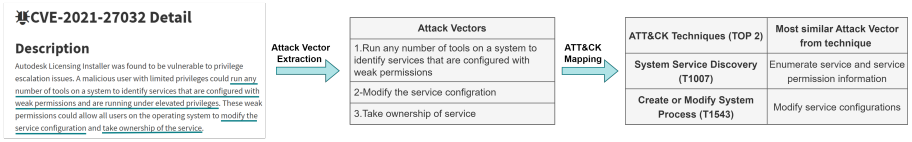

Fig. 8. SMET ATT&CK mapping

list and minimize the cosine similarity between pairs from the negative list. This approach is shown in Figure 6.

Specifically, the dataset was prepared as follows. From each technique, we extracted all attack vectors and combined each two extracted attack vectors in a list of pairs. We randomly selected 40 pairs from the list if the list size was more than 40. All selected pairs from all techniques were combined in the positive list. For each technique, we paired its attack vectors with at most six attack vectors from each other technique. We randomly sampled 160 pairs from the resulting pairings and added them to the negative list. This resulted in 38,396 pairs — 7,356 positives and 31,040 negatives. We fine-tuned the pre-trained all-mpnet-base-v2 model using our dataset. We used the Adam optimizer with a 2e-05 learning rate and fine-tuned the model for one epoch.

4.4 ATT&CK Mapping

Leveraging ATT&CK BERT, we trained a logistic regression model as follows. First, we collected all attack vectors from each technique and labeled each attack vector by its corresponding technique. We then used ATT&CK BERT to extract all attack vectors' embedding and trained a multinomial logistic regression model using this training set. This training approach is shown in Figure 7. We used logistic regression instead of a more complex model to avoid overfitting due to our dataset's high number of classes (185 techniques) and the small number of samples per class (52 median). Moreover, we used the class weights mechanism, where weights are adjusted inversely proportional to class frequencies during training to overcome the class imbalance.

When mapping a CVE entry to techniques, SMET runs the trained logistic regression model over each attack vector extracted from the entry and the entire entry description. Each run will return a probability score for each technique, representing how likely the attack vector belongs to the corresponding technique. SMET then combines all results by setting the probability score of each technique to the maximum score across all attack vectors. For example, if a CVE has two attack vectors, AV1 and AV2, we would run logistic regression three times — one time over each attack vector and a third over the entire entry text. For example, assume a technique T with probability scores of 0.2, 0.3, and .01 for AV1, AV2, and full description, respectively. SMET sets the probability score of T to 0.3. SMET does the same for all techniques and then ranks them based on the final probability scores.

Table 2. SMET and baselines' results

Model	Coverage Error	Ranking Loss	LRAP	R@5
SMET(our)	**13.96**	**0.05**	**53.77**%	**67.71**%
SBERT	32.66	0.126	24.58%	31.05%
TF-IDF	32.28	0.12	23.71%	30.90%
LLM chatbot	129.00	0.605	20.97%	33.60%

As ATT&CK BERT extracts semantic embedding of attack vectors, SMET explains its mapping by identifying the most similar attack vectors from the mapped technique's description or procedure examples. This explanation helps security analysts understand SMET's decision and gain insight into its mapping criteria. Figure 8 shows an example of SMET's mapping of CVE-2021-27032. The middle step presents a list of attack vectors extracted by SMET, and the last step shows SMET's two highest ranked techniques and the most similar attack vector from each technique to the CVE.

5 Evaluation

5.1 Dataset

To evaluate SMET, we collected 1,813 CVE entries published from 2014 to 2022 and gathered by existing research [16, 9]. We annotated the entries by mapping each entry to appropriate ATT&CK techniques. To ensure an accurate and justifiable mapping, we divide the mapping into two tasks. First, we identified attack vectors from the entry description. Second, we identified the most similar attack vector from ATT&CK for each extracted attack vector and mapped the entry to the corresponding technique. Each attack vector in an entry can be mapped to different techniques, so an entry can also be mapped to multiple techniques. Figure 3 shows an example of mapping CVE-2020-4553 to the user execution technique.

We mapped 303 CVE entries to 41 techniques from ATT&CK Matrix version 12.0. Figure 9 shows the number of CVE entries mapped to each technique. For example, the exploitation for the client execution technique had the most entries mapped to it, as many CVE entries in our dataset allow an attacker to execute code. Moreover, the four techniques with the most entry mappings all focus on vulnerability exploitation, whereas other techniques are more general.

5.2 Results

When evaluating a ranking model, we aim to rank the ground truth labels as low as possible. We evaluated SMET ranking using four multi-label ranking metrics: coverage error, label ranking loss, label ranking average precision (LRAP), and recall@k. LRAP calculates what percentage of the higher-ranked labels are true labels. LRAP values range from 0 to 1, excluding 0, and the best LRAP value is 1.

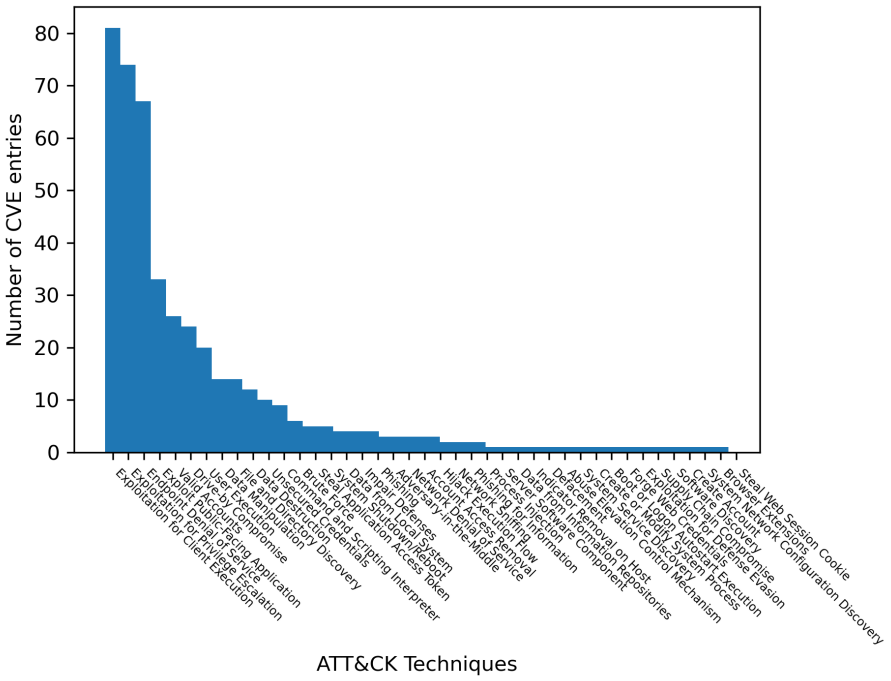

Fig. 9. Histogram of Ground Truth Techniques

Coverage error represents the average number of top-scored predictions required so that the predictions include all ground truth labels. The best coverage error value equals the average number of labels. Label ranking loss represents the average number of incorrectly ordered label pairs with respect to the number of correctly ordered labels. Label ranking loss's best value is 0. recall@K calculates the proportion of ground truth labels found in the top-k predictions. We set k to 5 to correspond with the top 2.55% of the predicted labels (185 techniques).

Below are the formulas of the first three metrics as defined by the scikit-learn python library [1], where $y \in \{0,1\}^{n_{samples} \times n_{labels}}$ is the ground truth labels and $\hat{f}$ is the score associated with each class.

$$LRAP(y, \hat{f}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \frac{1}{\|y_i\|_0} \sum_{j:y_{ij}=1} \frac{|\mathcal{L}_{ij}|}{rank_{ij}}$$

$$\mathcal{L}_{ij} = \left\{k : y_{ik} = 1, \hat{f}_{ik} \geq \hat{f}_{ij}\right\}, rank_{ij} = \left|\left\{k : \hat{f}_{ik} \geq \hat{f}_{ij}\right\}\right|$$

$$ranking\_loss(y, \hat{f}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \frac{1}{\|y_i\|_0 \left(n_{labels} - \|y_i\|_0\right)}.$$

$$\left\{(k,l) : \hat{f}_{ik} \leq \hat{f}_{il}, y_{ik} = 1, y_{il} = 0\right\}$$

$$coverage(y, \hat{f}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \max_{j:y_{ij}=1} rank_{ij}$$

We first compared SMET performance to the following two unsupervised text similarity models: SBERT, and TF-IDF. SBERT is a sentence embedding model introduced by [24], where they used the SIAMESE network to train transformer models on natural language inference datasets. In our experiments, we used the pre-trained all-mpnet-base-v2 model, which was trained on over 1 billion training pairs and designed for general-purpose text similarity. The all-mpnet-base-v2 model achieves the highest average score on over 20 NLP datasets [3].

TF-IDF is a traditional information retrieval model that computes the similarity of documents based on a statistical analysis of word distribution across documents. TF-IDF's robustness and simplicity make it the first go-to method in text similarity tasks.

In each case, we extracted the embeddings of ATT&CK technique descriptions and CVE entries and then ranked all techniques based on their cosine similarity to the entry. As shown in Table 2, our tool greatly outperforms all baselines in all metrics. We attribute the superior performance of SMET to three main reasons. First, instead of using the entries and technique descriptions for mapping, SMET uses attack vectors as a key feature, which reduces noise by eliminating irrelevant information. Second, SMET uses ATT&CK BERT, which we fine-tuned using the ATT&CK matrix and can thus extract embedding that better represents attack vectors. Third, instead of cosine similarity, SMET uses a logistic regression model that we trained to predict attack vectors' techniques.

LLM chatbots Recently, large language models (LLMs) chatbots have been used to solve various NLP tasks. These chatbots have knowledge of the web and can answer complicated questions regarding various topics without any fine-tuning. We studied the performance of one of the most popular chatbots in mapping CVE to ATT&CK by providing it with a CVE entry description and asking it to map that description to techniques. Due to government policy, we are unable to reveal the name of the chatbot. We conducted three experiments by providing the chatbot with the following three prompts:

1. Please identify MITRE ATT&CK techniques that can be associated with the CVE description below: "cve_des". Please return a list of identified ATT&CK techniques IDs.
2. Please identify what MITRE ATT&CK techniques an attacker can perform using the CVE description below: "cve_des". Please return a list of identified ATT&CK techniques IDs.
3. Please identify MITRE ATT&CK techniques that are mentioned in the CVE description below: "cve_des". Please return a list of identified ATT&CK techniques IDs.

We used regular expressions to extract technique IDs from each response.

Table 3. SMET components analysis

Modification	Model	Coverage Error	Ranking Loss	LRAP	R@5
-	SMET	**13.96**	**0.05**	**53.77**%	**67.71**%
Extraction Component	LLM chatbot	15.24	0.55	48.59%	61.61%
Embedding Model	MPNET	15.84	0.058	46.93%	61.36%
	SecBERT	29.51	0.115	34.22%	44.31%
	$BERT_{Base}$	30.38	0.117	33.31%	43.41%

Although the chatbot showed a basic understanding of ATT&CK, its mapping performance was inferior. Although the prompts were semantically similar, the chatbot showed inconsistent behavior and responded with a different set of techniques for each prompt. For example, the number of techniques the the chatbot extracted across all responses were 976, 993, and 486 techniques for the first, second, and third prompts, respectively. In only six instances, all three prompts provided the same techniques, and in 41 instances, the three prompts agreed on at least one technique.

To compare the chatbot to SMET, we combined the prediction of the three prompts by making each prompt vote for its predicted techniques and ranked the techniques based on the number of votes. Techniques with the same number of votes were ranked randomly. We presented the results in Table 2. The chatbot performance was inferior in all metrics except R@5, where it performed slightly better than the TF-IDF and SBERT models. We attribute these results to the fact that the chatbot is not a ranking model. Even though we ranked techniques based on prompt voting, most techniques had no votes and were ranked randomly. However, R@5 is a classification metric that gives a more accurate indication of the chatbot's performance.

5.3 SMET Component Analysis

We investigated the performance of SMET after replacing its components with other existing alternatives from the literature. In the first experiment, we replaced the attack vector extraction model with a LLM chatbot by using the following prompt: "Please identify actions that can be done by an attacker from the following description inclusively: "cve_des." Return each action along with all its objects in a line with the format subject—verb—objects." We then used a regular expression to extract the attack vectors from the chatbot's responses. In the second experiment, we replaced ATT&CK BERT with different transformer models introduced by researchers. We used $BERT_{Base}$, which was introduced by the original BERT paper [11]. Moreover, we used the sentence transformer model all-mpnet-base-v2, which was trained on over 1 billion training pairs to achieve meaningful semantic embedding. Lastly, we used SecBERT, a domain-specific transformer model trained over cybersecurity data, such as CTI reports, articles, and databases [2].

As shown in Table 3, SMET achieved the highest score across all metrics. On average, $BERT_{Base}$ exhibited the weakest performance, as BERT is not a domain-specific model, and its embedding is not semantically meaningful. Although SecBERT is a domain-specific model, its performance was comparable to BERT's, as its embedding is not semantically meaningful either. MPNET (all-mpnet-base-v2) achieved a higher performance as it was trained on a text similarity task. Thus, its embedding is semantically meaningful. Finally, SMET achieved the highest performance as it uses ATT&CK BERT, which was trained on cybersecurity data (ATT&CK Matrix), and its embedding is semantically meaningful.

When we used the LLM chatbot for extraction, the performance of SMET decreased. As discussed in the previous section, the chatbot output is inconsistent and sensitive to prompts. Moreover, unlike SRL models, the chatbot responds with unstructured text, which requires further processing using regular expressions to extract attack vectors. This unstructured response introduces noise that might affect the quality of extracted attack vectors. More investigation needs to be conducted to study the possibility of using LLM chatbots to accurately extract attack vectors from unstructured text. We leave this challenge for future work.

6 Conclusion and Future Work

In this paper, we introduced SMET — a tool that automatically maps CVE to ATT&CK to assist security analysts in understanding vulnerabilities and gaining more insight into appropriate countermeasures. SMET utilizes a semantic extraction model to identify information regarding attack actions from CVE and ATT&CK descriptions and leverages ATT&CK BERT, a model that we propose to extract semantically meaningful embedding of unstructured attack description text. We evaluated SMET using a ground truth that we manually created based on our analysis of the CVE-ATT&CK association. SMET achieved a robust performance compared to existing state-of-the-art models. Moreover, we studied the importance of SMET's components by replacing them with existing alternatives from the literature. Although this paper focused on CVE mapping, SMET has the potential to be applied to other forms of cybersecurity resources (e.g., CTI reports, data breach reports, cybersecurity articles and news). Mapping these sources to ATT&CK can also be very helpful to security analysis in understanding current attack trends and applicable countermeasures. We leave this challenge to future work due to the complexity of processing such unstructured documents and the need for additional robust components to achieve accurate mapping.

Disclaimer Certain equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

1. 3.3. metrics and scoring: Quantifying the quality of predictions, https://scikit-learn.org/stable/modules/model_evaluation.htmlmultilabel-ranking-metrics
2. Jackaduma/secbert · hugging face, https://huggingface.co/jackaduma/SecBERT
3. Pretrained models, https://www.sbert.net/docs/pretrained_models.html
4. Aghaei, E., Niu, X., Shadid, W., Al-Shaer, E.: Securebert: A domain-specific language model for cybersecurity. In: Security and Privacy in Communication Networks: 18th EAI International Conference, SecureComm 2022, Virtual Event, October 2022, Proceedings. pp. 39–56. Springer (2023)
5. Aghaei, E., Shadid, W., Al-Shaer, E.: Threatzoom: Hierarchical neural network for cves to cwes classification. In: Security and Privacy in Communication Networks: 16th EAI International Conference, SecureComm 2020, Washington, DC, USA, October 21-23, 2020, Proceedings, Part I. pp. 23–41. Springer (2020)
6. Ampel, B., Samtani, S., Ullman, S., Chen, H.: Linking common vulnerabilities and exposures to the mitre att&ck framework: A self-distillation approach. arXiv preprint arXiv:2108.01696 (2021)
7. Andrew, Y., Lim, C., Budiarto, E.: Mapping linux shell commands to mitre att&ck using nlp-based approach. In: 2022 International Conference on Electrical Engineering and Informatics (ICELTICs). pp. 37–42. IEEE (2022)
8. Ayoade, G., Chandra, S., Khan, L., Hamlen, K., Thuraisingham, B.: Automated threat report classification over multi-source data. In: 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC). pp. 236–245. IEEE (2018)
9. Center-For-Threat-Informed-Defense: Center-for-threat-informed-defense/attack_to_cve: A methodology for mapping mitre att&ck techniques to vulnerability records to describe the impact of a vulnerability., https://github.com/center-for-threat-informed-defense/attack_to_cve
10. Das, S.S., Halappanavar, M., Tumeo, A., Serra, E., Pothen, A., Al-Shaer, E.: Vwc-bert: Scaling vulnerability–weakness–exploit mapping on modern ai accelerators. In: 2022 IEEE International Conference on Big Data (Big Data). pp. 1224–1229. IEEE (2022)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
12. Editor, C.C., Cooper, C., Editor, C., the AuthorCharles CooperConsulting EditorCharles Cooper has covered technology, A., business for more than 25 years. He is now assisting Symantec with our blog writing, managing our editorial team., Author, A.t.: Wannacry: Lessons learned 1 year later, https://symantec-enterprise-blogs.security.com/blogs/feature-stories/wannacry-lessons-learned-1-year-later
13. Engenuity, M.: Mapping attamp;ck to cve: Threat-informed defense project (Jan 2023), https://mitre-engenuity.org/blog/2021/10/21/mapping-attck-to-cve-for-impact/
14. Fleck, A., Richter, F.: Infographic: Cybercrime expected to skyrocket in coming years (Dec 2022), https://www.statista.com/chart/28878/expected-cost-of-cybercrime-until-2027/
15. Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N.F., Peters, M., Schmitz, M., Zettlemoyer, L.S.: Allennlp: A deep semantic natural language processing platform (2017)

16. Grigorescu, O., Nica, A., Dascalu, M., Rughinis, R.: Cve2att&ck: Bert-based mapping of cves to mitre att&ck techniques. Algorithms **15**(9), 314 (2022)
17. 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 preprint arXiv:2010.00533 (2020)
18. Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., Niu, X.: 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 (2017)
19. Izzuddin, A.B., Lim, C.: Mapping threats in smart grid system using the mitre att&ck ics framework. In: 2022 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES). pp. 1–7. IEEE (2022)
20. Kuppa, A., Aouad, L., Le-Khac, N.A.: Linking cve's to mitre att&ck techniques. In: Proceedings of the 16th International Conference on Availability, Reliability and Security. pp. 1–12 (2021)
21. Legoy, V., Caselli, M., Seifert, C., Peter, A.: Automated retrieval of att&ck tactics and techniques for cyber threat reports. arXiv preprint arXiv:2004.14322 (2020)
22. Li, Z., Zeng, J., Chen, Y., Liang, Z.: Attackg: Constructing technique knowledge graph from cyber threat intelligence reports. In: Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I. pp. 589–609. Springer (2022)
23. Mendsaikhan, O., Hasegawa, H., Yamaguchi, Y., Shimada, H.: Automatic mapping of vulnerability information to adversary techniques. In: The Fourteenth International Conference on Emerging Security Information, Systems and Technologies SECUREWARE2020 (2020)
24. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
25. Sajid, M.S.I., Wei, J., Abdeen, B., Al-Shaer, E., Islam, M.M., Diong, W., Khan, L.: Soda: A system for cyber deception orchestration and automation. In: Annual Computer Security Applications Conference. pp. 675–689 (2021)