

Attack Tactic Labeling for Cyber Threat Hunting

SHENG-XIANG LIN, ZONG-JYUN LI, TZU-YANG CHEN, DONG-JIE WU

*Cybersecurity Technology Institute, Institute for Information Industry, Taiwan, R.O.C

leafin@iii.org.tw, tomlee@iii.org.tw, tzuyangchen@iii.org.tw, dongjiewu@iii.org.tw

Abstract— Recently, the cyber attack has become more complex and targeted, making traditional security defense mechanisms based on the “Indicator of Compromise” ineffective. Furthermore, fail to consider attack kill chain may lead to a high false-positive rate for attack detection. To trace hackers’ behaviors and footprints, it is crucial to provide additional information such as attack tactics, techniques, and procedures in detecting attacks. In this study, we propose a mechanism for labeling attack tactics of network intrusion detection system (NIDS) rules on the basis of text mining and machine learning. The proposed approach can help security experts determine the current attack state and infer its purpose, making it possible to detect complex attacks (e.g., APT). Besides, we refer to the ATT&CK framework developed by MITRE (a leading organization in information security) to strengthen the reliability of labeling results. The experiment result shows that the accuracy of our proposed mechanism can effectively boost the performance of the labeling attack tactic. The experimental result shows that the F1 score of our approach is more than 90% and up to approximately 96%, which can effectively assist cyber security experts in tactic labeling and provides a solid base for further alert correlation. Moreover, we also compare our approach with one of the well-known TTP labeling tools, rcATT; the result shows that our approach’s accuracy, precision, recall, and F1 score are all significantly better than rcATT.

Keywords—Cyber threat hunting, Cyber kill chain, Tactic labeling, APT detection, Machine Learning, Malicious Traffic Detection

I. INTRODUCTION

In recent years, information security incidents have occurred more and more frequently, how to detect and respond to multi-stage APT attacks is an important and challenging issue. Traditional detection approaches based on signatures or indicators of compromise (IoC) are challenging to cope with attacks composed of advanced techniques, tools, and exploits. Furthermore, such approaches may also cause a high false-positive rate [1] [2], which makes it hard for security operators to respond correctly in time.

Many researchers suggest correlating alerts raised by various security applications such as IDS, IPS, and firewall to track attack footprints instantly. Also, they try to utilize the concept of kill chain to illustrate attack progress and find hackers’ intentions. Labeling attack tactics and techniques is the most critical thing for kill chain analysis, some researches [3] [4] focus on this part based on the kill chain definition proposed by a white paper [5]. Another study [6] carries out the labeling work according to the ATT&CK framework proposed by

MITRE. The ATT&CK framework defines various attack “tactics, techniques, and procedures (TTP)” according to actual incident cases in the past. It has become an industry standard; most security applications have adopted the MITRE ATT&CK framework, which makes it very authoritative. In short, TTP labeling of attack detection rules is essential for further alert correlation and cyber threat hunting. However, with the rapid changes of attack techniques, there is no systematic and continuous labeling mechanism for the detection rules. It would be difficult to identify new types of attacks, and there will be breakpoints in the hackers’ footprints.

TTP labeling is one of the most effective ways to assist alert correlation and the APT attack footprints tracing. This allows alerts from different security applications to have a common language to describe attack behaviors. To ensure the quality of alert correlation, it is necessary to have numerous security expert-labeled rulesets. However, different security applications have their rulesets, which makes TTP labeling time-consuming and labor-intensive. For the purpose of improving the efficiency of TTP labeling, we propose a solution that can efficiently assist security experts in labeling work.

In this study, we observe that the field information in detection rule is meaningful. Take “Suricata” NIDS as an example, the threat description is usually in the “msg” field. As stated above, that information can be linked to the description of TTP in MITRE’s ATT&CK framework. According to this observation, we propose a mechanism for attack tactic labeling. The mechanism is composed of two parts, “similarity-based labeling” and “machine learning-based labeling.”

We aim to improve the accuracy through machine learning algorithms. However, it requires numerous datasets to train model; since there are few public TTP labeled datasets for Suricata rules, the purpose of “similarity-based labeling” is to accumulate lots of labeled datasets for subsequent machine learning use. The experiment result shows that the accuracy, precision, recall, and F1 score are reached to more than 90% and up to 96%, significantly better than similar research [7].

This study shows that our proposed mechanism can make tactic labeling of detection rules much more efficient and contribute to kill chain analysis for cyber threat hunting.

The primary contributions of this study are given as follows:

(1) providing a mechanism to help security experts quickly and correctly label attack tactics of intrusion detection rules, making it easier to trace behaviors and footprints of attackers.

(2) our approach keeps flexibility which can be further applied in various cyber security applications and different object labeling such as techniques and kill chain stages.

(3) providing a solid base for further alert correlation.

The rest of this paper is organized as follows: Section II provides some related works; Section III gives an overview of the proposed mechanism; Section IV illustrates the experiment results; Finally, we make our conclusion in Section V.

II. RELATED WORK

Researchers from Lock Martin first propose “Intrusion Kill Chain” [8], which divide intrusion into several stages to grasp attack status effectively and mitigate cyber attacks. Each stage of the kill chain is mapped to courses of action for mitigation measures. MITRE company propose the ATT&CK framework [9] [10] [11], which further expand the kill chain into 12 tactics, and each tactic has different techniques to describe the actions of attacks for specific purposes.

A. Kill Chain

Most studies have applied kill chain and ATT&CK framework to detect cyber attacks. Ru Zhang et al. mention that the accuracy of attack detection depends on the integrity of the kill chain [3]. When adopting fuzzy clustering to correlate alerts, they cluster alerts to APT attack sequence sets. Then, they establish the probability transfer matrix to construct APT attack scenarios which can identify APT attacks in the network. In their experiments, they classify approximately 1000 valid alerts. The APT attack scenarios constructed by the alerts can match the actual situation and be useful for APT attack detection. Florian Wilkens et al. label the kill chain stage by deducing from the network direction of each IDS alert in their research [12]. They use the finite state machine method to depict the APT scenario graphs. Approximately 446,458 IDS alerts are condensed into 700 APT scenario graphs through this method.

Besides, kill chain is also used in researches [1] [2] [4] [6] [13] for cyber attack detection. Thus, it can be seen that the kill chain is usually used to assist in attack detection. Nowadays, the ATT&CK framework is the most commonly used to describe the cyber kill chain. However, the rise in the number of attack techniques and corresponding detection rules will limit the detection coverage rate. Therefore, to rapidly expand the detection coverage, developing a mechanism for continuously labeling amounts of rules with TTP is urgent. The related literature about tactic labeling will be explained in the next paragraph.

B. Tactic Labeling

To understand the purpose of threat actions and to hunt threat actors more efficiently, Ghaith Husari et al. develop automated context-aware analytics of cyber threat intelligence [14]. Their study can assist experts in mapping threat actions to attack patterns (TTPs) and taking cyber defense actions timely. The previous research continues and publishes again after two years [15]. They apply the natural language processing (NLP) approach to characterize the temporal relationship among APT attack behaviors to enhance the accuracy of TTP labeling. Valentine Legoy et al. [7] also utilize the ATT&CK framework

for training classifiers through machine learning algorithms. They choose “Binary Relevance” and “Classifier Chain” approaches to solve the multi-label classification problem and use various classification algorithms such as the Bernoulli Naive Bayes classifier, K-Nearest Neighbors. Finally, an automatic TTP labeling tool for cyber threat reports named reATT [16] is published and compared with the open-source tool, Unfetter Insight [17]. The tool can help cyber security experts easily obtain threat actions and indicators of compromise from threat reports.

III. PROPOSED APPROACH

In this section, we propose an “Attack Tactic Labeling” mechanism for further cyber threat hunting considering labeling efficiency and accuracy. The mechanism consists of two parts. The first part is similarity-based tactic labeling, which aims to efficiently assist experts with tactic labeling and accumulate the dataset quickly. The second part is machine learning-based tactic labeling, learning features of different tactics from the expanded dataset. In the following subsections, the complete architecture is described in detail.

A. Similarity-based Tactic Labeling

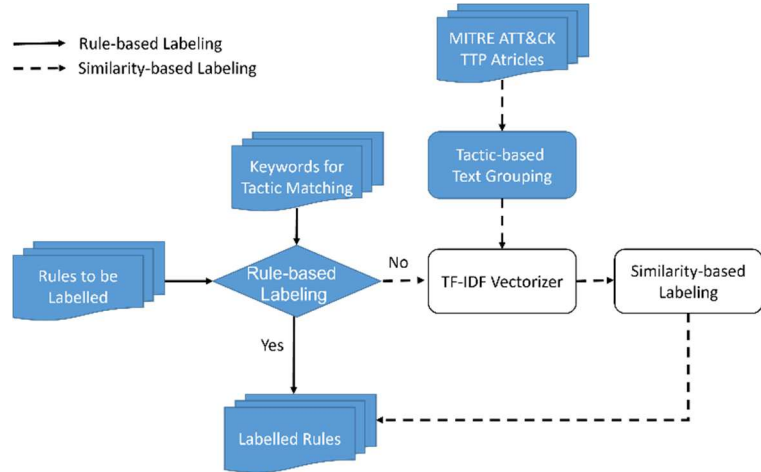


Figure 1. The Architecture of Similarity-based Tactic Labeling

The architecture of “Similarity-based Tactic Labeling” is shown in Figure 1. In the beginning, we regard tactic labeling for Suricata rules as an information retrieval problem because of the limited dataset. We consider the description of the “msg” field in Suricata rules to be query text and compare them with matching rules defined by experts in the “Keywords for Tactic Matching” function. If no matching rules can be met, the Suricata rule to be labeled will then be passed to the TF-IDF Vectorizer in the similarity-based labeling phase. Afterward, the vectorizer calculates the importance of terms in each tactic through TF-IDF algorithm. Finally, according to the similarity between the Suricata rule and each tactic group, the tactic group with the highest similarity is taken as the labeling result.

1) Rule-based Labeling: To accelerate the efficiency and accuracy of the labeling process, we develop matching rules by information security experts for tactic labeling. Since some specific keywords or information (e.g., protocol, threat type,

and malware family) can be directly used to determine the tactic group of the Suricata rule, experts define keywords for each tactic group. For example, “remote code execution” can be labeled as “lateral movement” tactic, and the terms related to OWASP Top 10 vulnerabilities can be labeled as “initial-access” tactic.

2) Similarity-based Labeling: The Suricata rules which cannot be labeled by matching rules will be passed to the similarity-based labeling stage. The stage includes three parts:

a) Tactic-based Text Grouping: Considering the information in the Suricata rules is limited, it may significantly affect the effectiveness of machine learning approaches. Therefore, we refer to the ATT&CK framework [11] proposed by the MITRE organization for the tactics, techniques, and tools used by attackers. These definitions are now the mainstream reference for the industry, and we believe that they will be helpful for tactic labeling. We collect each technique definition (including attack scenarios) and categorize all terms contained therein into their respective tactic for subsequent information retrieval and classification.

b) TF-IDF Vectorizer: Before analyzing the text information, we need to pre-process the data to filter terms that may potentially affect the accuracy of the results. First, we remove any non-text characters, including punctuation marks and numbers. Besides, to prevent the English tense from affecting the result of TF-IDF, we lemmatize each vocabulary and convert it to its base form. Furthermore, we delete meaningless stop words such as ‘he’ (pronoun), ‘a’ (article). Moreover, we remove the tag prefix that often appears in the “msg” field of the Suricata rule. By doing so, we reduce the influence of irrelevant noise for the prediction model. Finally, many attacks and professional terms in information security usually consist of two or three words and are presented in abbreviations. Thus, we convert the abbreviations into complete terms at this stage. Then, we use the original unigram and combine it with 2-gram to retain the semantic and contextual information.

When doing text analysis, TF-IDF is one of the most important methods used to evaluate the importance of a term in a document of the document set. TF-IDF counts the frequency of terms appearing in a document (Term Frequency) and uses Inverse Document Frequency to reduce the weight according to the number of times a term appears in all documents. Therefore, we use TF-IDF to analyze the article and calculate a vector to evaluate the similarity between the content of the “msg” field in the Suricata rule and each tactic group. The set of all index text represents as $V = \{w_1, \dots, w_{|V|}\}$, while w_i is the i^{th} index term and d_j represents j^{th} document. The inverse document frequency is computed as follows:

$$idf_i = \log\left(\frac{N + 1}{|\{j : w_i \in d_j\}| + 1}\right)$$

With N is the number of documents, and idf_i is the number of documents that contain w_i . Thus, TF-IDF is defined as:

$$S_{i,j} = \text{TF-IDF}_{i,j} = tf_{i,j} \times idf_i$$

A term gets a higher weight if it occurs many times in a small number of documents; on the other hand, the weight will be lower when it occurs a few times in a document or appears in many documents.

c) Similarity-based Labeling: After TF-IDF Vectorizer calculates the respective feature vectors for MITRE articles of each tactic and the “msg” field in Suricata rules, we use different similarity algorithms (including Cosine Similarity, Pearson Correlation Coefficient, and Euclidean Distance) to label the tactic of the detection rule. The introduction of similarity methods we used is as follows:

- **Cosine Similarity:** The cosine similarity is evaluated by calculating the angle between two vectors in space; when the angle between the two vectors is smaller, the cosine value tends to 1, reaching the maximum value of 1 at 0 degrees. The cosine similarity between document d_j and query q is defined as

$$\text{sim}(q, d_j) = \frac{\sum_{w_i \in V} S_{i,q} \times S_{i,j}}{\sqrt{\sum_{w_i \in V} S_{i,q}^2} \times \sqrt{\sum_{w_i \in V} S_{i,j}^2}}$$

- **Pearson Correlation Coefficient:** The Pearson correlation coefficient subtracts an average calculated from the vectors in all dimensions and then computes the cosine similarity. The result for the Pearson correlation coefficient has been proved to accord with the actual condition while keeping the original characteristics.

$$p(q, d_j) = \frac{\sum_{w_i \in V} (S_{i,q} - \bar{S}_q) \times (S_{i,j} - \bar{S}_j)}{\sqrt{\sum_{w_i \in V} (S_{i,q} - \bar{S}_q)^2} \times \sqrt{\sum_{w_i \in V} (S_{i,j} - \bar{S}_j)^2}}$$

- **Euclidean Distance:** Euclidean distance is a common method in geometry to calculate the linear distance between two points. The following equation calculates the Euclidean distance:

$$d(q, d_j) = \sqrt{\sum_{w_i \in V} (S_{i,q} - S_{i,j})^2}$$

B. Machine Learning-based Tactic Labeling

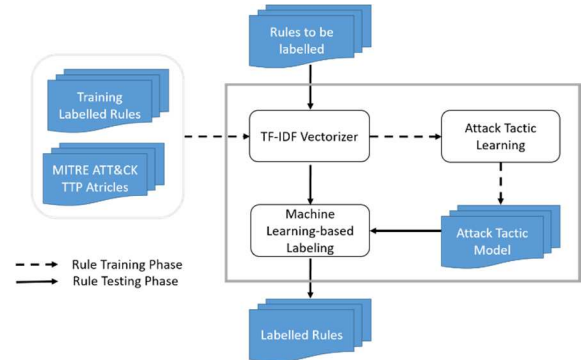


Figure 2. The Architecture of Machine Learning-Based Tactic Labeling

With the help of “Similarity-based Labeling,” the labeled dataset can be expanded efficiently. Next, we aim to learn the

features of each tactic to improve the accuracy of labeling through machine learning algorithms. Machine learning can be effective for subsequent prediction only when the dataset is sufficient. Thus, we use the large labeling dataset extended by Similarity-based Labeling part to train models for tactic classification in Machine Learning-based Tactic Labeling part.

The architecture of “Machine Learning-based Tactic Labeling” is shown in Figure 2. First, the labeled Suricata rule dataset is vectorized by the TF-IDF Vectorizer. Next, the attack tactic learning module trains the model with specified supervised learning algorithms. Finally, the rules that need to be labeled are vectorized into features and then predicted the tactic group with the model.

3) TF-IDF Vectorizer: Machine learning-based labeling includes the training phase and testing phase. In the training phase, TF-IDF is used to analyze the MITRE textual data, similar to the one used in similarity-based labeling. However, the difference is that the training data in machine learning-based labeling also includes the labeled result from similarity-based labeling. The terms in unlabeled Suricata rules will be converted into the feature vector for machine learning methods to identify tactics in the testing phase.

4) Attack Tactic Learning: we choose linear model classifiers and tree-based model classifiers for tactic learning.

- *Support Vector Machine (SVM):* SVM classifies data through multi-dimensional hyperplane (hyperplane, $wx + b$) segmentation. The appropriate weights (the value of w and b) will be obtained in the training phase, and the trained model will be used to predict data.
- *K-Nearest Neighbours (KNN):* KNN generally does not have an actual data training process. Instead, it calculates the distance between the test data and each training data in the prediction stage, finds the k adjacent data, and uses the category with the highest frequency as the classification result.
- *Decision Tree:* A decision tree is a flowchart-like structure. Each internal node represents a "test" on an attribute, each branch represents the test result, and each leaf node represents a class label. The paths from the root to the leaf represent classification rules.
- *Random Forest:* Random forest is an ensemble learning method for classification work that operates by constructing many decision trees at training time. The output of the random forest is the class selected by most trees.

5) Machine Learning-based Labeling: After the TF-IDF Vectorizer function extracts features of the testing dataset, the tactic to which the Suricata rule belongs is predicted with the pre-trained model.

IV. EXPERIMENT RESULT

To evaluate the proposed mechanism, we make experiments for assessing the effectiveness of tactic labeling. The applied

dataset for experiments is introduced in Sec IV-A. The evaluation metrics are described in Sec IV-B. The effectiveness evaluation of similarity-based and machine learning-based tactic labeling are in Sec IV-C and Sec IV-D, respectively.

A. Dataset

We collect two kinds of datasets; the first dataset is composed of “technique” article sets in all network-related tactics from MITRE ATT&CK framework, including 12 tactics, 137 definition articles, and 6711 words. The statistics of the articles collected for each tactic is summarized in TABLE 1.

TABLE 1. THE STATISTICS OF THE ARTICLES COLLECTED FOR EACH TACTIC

Tactic	# of articles	# of terms
Reconnaissance	7	262
Initial Access	7	487
Execution	3	331
Persistence	9	483
Defence Evasion	17	825
Credential Access	6	405
Discovery	2	240
Lateral Movement	13	567
Collection	8	368
Command and Control	37	1595
Exfiltration	14	593
Impact	14	555

The second dataset includes Suricata rules officially released by Snort (snortrules-snapshot-2983). We select four types of rules, namely “malware_cnc”, “malware_other”, “server-webapp,” and “top-100”. The “malware_cnc” is used to detect command and control activities such as file dropping, data exfiltration, and credential theft. The “malware-other” is used to detect behaviors related to malicious programs such as remote code execution and keylogger runtime. The “server-webapp” is used to detect attacks against vulnerabilities of web applications. The “top-100” includes high-risk attack techniques (e.g., Cobalt Strike and Eternalblue). In summary, the dataset contains four categories, 12,978 rules and 83,176 words. The statistics of the rule dataset is shown in TABLE 2.

TABLE 2. THE STATISTICS OF RULES DATASET

Rule Category	# of Rules	# of terms
malware-cnc	5,470	45,537
malware-other	3,566	15,302
server-webapp	3,842	21,833
top-100	100	504

B. Evaluation Metrics

We evaluate the effectiveness of tactic labeling results by following four indicators:

- *Accuracy:* An intuitive effectiveness measure and it is simply a ratio of correctly predicted observation to the total observations. The *accuracy* is defined by

$$\text{Accuracy} = \frac{1}{L} \sum_{k=1}^L \frac{TP_k + TN_k}{TP_k + FP_k + TN_k + FN_k}$$

TP, FP, TN, FN represent true positive, false positive, true negative, and false negative, respectively. L is the number of labels.

- **Macro Precision:** The ratio of correctly predicted positive observations to the total predicted positive observations. The *precision* is defined by

$$\text{Precision} = \frac{1}{L} \sum_{k=1}^L \frac{TP_k}{TP_k + FP_k}$$

- **Macro Recall:** The ratio of correctly predicted positive observations to all observations in the actual class. The *recall* is defined by

$$\text{Recall} = \frac{1}{L} \sum_{k=1}^L \frac{TP_k}{TP_k + FN_k}$$

- **Macro F1 score:** The weighted average of Precision and Recall. The score takes both false positives and false negatives into account. The *F1 score* is defined by

$$\text{F1 score} = \frac{1}{L} \sum_{k=1}^L \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

C. Similarity-based Tactic Labeling

We choose the TF-IDF algorithm as a feature weighting method for textual analysis and apply three different similarity calculation algorithms, including “Cosine Similarity,” “Pearson Correlation Coefficient,” and “Euclidean Distance.”

We use all rules dataset in the first experiment; the results are noted in Table 3.

TABLE 3. EXPERIMENT RESULTS OF DIFFERENT SIMILARITY ALGORITHM

Similarity	Accuracy	Precision	Recall	F1 score
Cosine Similarity	72.39%	80.36%	72.44%	73.38%
Pearson Correlation Coefficient	72.66%	80.21%	72.71%	73.45%
Euclidean Distance	72.41%	80.38%	72.45%	73.39%

To further assess the effectiveness of our approach in different rule categories, we conduct separate experiments for each rule category. The experiment results are shown in TABLE 4. We observe that the overall results of the malware-related category (e.g., malware-cnc and malware-other) are better than the other categories. We then further analyze the reasons and discover that the categories with better results are because the texts in the “msg” field have a higher overlap with the texts in tactic articles. For example, the “download” term and the “backdoor” term appeared in the malware-related category are the keyword in the “Command and Control” tactic and the “Persistence” tactic, respectively. However, the terms in the “server-webapp” category are mainly presented as the

“OWASP Top 10” term for web-related attacks, with very few attack-related technique terms. They have a lower overlap with the terms in the tactic articles, which is the main reason for the poor results.

TABLE 4. EXPERIMENT RESULTS OF PEARSON CORRELATION COEFFICIENT IN DIFFERENT RULE DATASET

Dataset	Accuracy	Precision	Recall	F1 score
malware-cnc	88.57%	89.92%	88.59%	88.18%
malware-other	84.97%	92.75%	85.1%	88.63%
server-webapp	38.99%	84.78%	38.99%	49.83%
top-100	57%	64.03%	57%	57.8%

Overall, the experiment results show that the F1 score of tactic labeling can reach over 70% and up to 73%, which can significantly help security experts improve the efficiency of tactic labeling. After this stage, a large number of tactic-labeled datasets are produced. Then, we use these accumulated datasets in “Machine Learning-Based.”

D. Machine Learning-based Tactic Labeling

TABLE 5 shows the experiment results of different machine learning algorithms. We apply four different methods, including “K-Nearest Neighbours (KNN),” “Support Vector Machine (SVM),” “Decision Tree,” and “Random Forest.” In the experiment, k-fold cross-validation (k=5) is used for evaluation. The experiment results show that the F1 score of tactic labeling can reach over 90% and up to 96%, which can significantly help security experts improve the effectiveness of tactic labeling. From the results, we can find that the KNN result is slightly worse because we treat keyword terms in the tactic group as features, and these features form a multi-dimension feature vector, resulting in the KNN not performing well in the data with higher dimensions.

TABLE 5. THE EXPERIMENT RESULTS OF DIFFERENT MACHINE LEARNING ALGORITHMS

Algorithm	Accuracy	Precision	Recall	F1 score
SVM	96.96%	96.84%	96.97%	96.7%
KNN	93.26%	93.38%	93.25%	93.23%
Decision Tree	96.46%	96.29%	96.30%	96.20%
Random Forest	96.57%	96.19%	96.42%	96.12%

Finally, we make the effectiveness comparison of tactic labeling between our approach and the rcATT [16]. The experiment result described in TABLE 6 shows that our approach is significantly better than the rcATT.

Although the rcATT study published by Valentine Legoy et al. [7] has good TTP labeling results for long articles, their method is limited for handling data with little information. The comparison results show that our approach can still label tactics effectively even with little information.

TABLE 6. DETECTION EFFECTIVENESS BETWEEN OUR APPROACH AND rcATT

Approach	Accuracy	Precision	Recall	F1 score
Similarity-based (Pearson Correlation Coefficient)	72.66%	80.21%	72.71%	73.45%
Machine Learning-based (SVM)	96.96%	96.84%	96.97%	96.7%
rcATT	8.14%	78.53%	8.14%	9.92%

V. CONCLUSIONS

We propose a mechanism for labeling attack tactics of network intrusion detection system (NIDS) rules on the basis of text mining and machine learning, which can assist security operators in tracing hackers' behaviors and footprints. The proposed mechanism is separated into two parts, including similarity-based tactic labeling and machine-based tactic labeling. In the similarity-based tactic labeling part, the accuracy reaches over 70% and up to 72.6%. It can assist experts in labeling tactics for numerous detection rules efficiently and accumulate the labeled dataset quickly. In the machine-based labeling part, we utilize multiple machine learning algorithms to train the dataset, which expand from the similarity-based labeling part. The experiment result shows that the accuracy of tactic labeling achieves over 90% and up to 96%. Our proposed mechanism can effectively correlate detection rules from various cyber security applications and contribute to APT attacks hunting.

ACKNOWLEDGMENT

This research is supported in part by the Ministry of Economic Affairs of Taiwan under grants number 110-EC-17-A-21-1702.

REFERENCES

- [1] W. Wang, L. Yu, H. He and X. Gong, "MAAC: Novel Alert Correlation Method To Detect Multi-step Attack," 2020.
- [2] S. Haas and M. Fischer, "GAC: graph-based alert correlation for the detection of distributed multi-step attacks," in *ACM/SIGAPP Symposium On Applied Computing (SAC)*, 2018.
- [3] R. Zhang, Y. Huo, J. Liu and F. Weng, "Constructing APT Attack Scenarios Based on Intrusion Kill Chain and Fuzzy Clustering," 2017.
- [4] I. Herwono and F. A. El-Moussa, "Automated Detection of the Early Stages of Cyber Kill Chain," in *International Conference on Information Systems Security and Privacy (ICISSP)*, 2018.
- [5] P. Pols, "The Unified Kill Chain," in *Cyber Security Academy (CSA)*, 2017.
- [6] B. D. Bryant and H. Saiedian, "Improving SIEM alert metadata aggregation with a novel kill-chain based classification model," *Computers & Security*, 2020.
- [7] V. Legoy, M. Caselli, C. Seifert and A. Peter, "Automated Retrieval of ATT&CK Tactics and Techniques for Cyber Threat Reports," 2020.
- [8] E. M. Hutchins and M. J. Cloppert, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and

Intrusion Kill Chains," in *International Conference on Information Warfare and Security (ICIWS)*, 2011.

- [9] B. E. Strom, A. Applebaum, P. D. Miller, K. C. Nickels, A. G. Pennington and C. B. Thomas, "MITRE ATT&CK: Design and Philosophy," The MITRE Corporation, 2018.
- [10] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, P. D. Miller, C. Wampler, S. M. Whitley and R. D. Wolf, "Finding Cyber Threats with ATT&CK-Based Analytics," The MITRE Corporation, 2017.
- [11] "MITRE ATT&CK®," The MITRE Corporation, [Online]. Available: <https://attack.mitre.org/>.
- [12] F. Wilkens, F. Ortmann, S. Haas, M. Vallentin and M. Fischer, "Multi-Stage Attack Detection via Kill Chain State Machines," 2021.
- [13] R. Al-Shaer, J. M. Spring and E. Christou, "Learning the Associations of MITRE ATT&CK Adversarial Techniques," in *IEEE Conference on Communications and Network Security (CNS)*.
- [14] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu and X. Niu, "TTPDrill: Automatic and Accurate Extraction of Threat Actions from Unstructured," in *Annual Computer Security Applications Conference (ACSAC)*, 2017.
- [15] G. Husari, E. Al-Shaer, B. Chu and R. F. Rahman, "Learning APT Chains from Cyber Threat Intelligence," in *Annual Symposium on Hot Topics in the Science of Security (HotSoS)*, 2019.
- [16] V. Legoy, "vlegoy/rcATT: A python app to predict Att&ck tactics and techniques from cyber threat reports," [Online]. Available: <https://github.com/vlegoy/rcATT>.
- [17] "Unfetter: Unfetter insight," 2018. [Online]. Available: <https://github.com/unfetter-discover/unfetter-insight>.



Sheng-Xiang Lin received his M.S. degree in Department of Computer Science and Information Engineering from National Ilan University, Yilan, Taiwan, R.O.C. in 2019. He is currently an engineer in Cybersecurity Technology Institute, Institute for information industry, Taiwan, R.O.C. His research interests include forensic analysis, reverse engineer, penetration testing, and machine learning skills.



Zong-Jyun Li received his B.S. degree in Department of Computer Science and Information Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C. in 2018. He is currently an engineer in Cybersecurity Technology Institute, Institute for information industry, Taiwan, R.O.C. His research interests include forensic, reversing, and red team skills.



Tzu-Yang Chen received his B.S. degree in Department of Computer Science and Information Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C. in 2020. He is currently an engineer in Cybersecurity Technology Institute, Institute for information industry, Taiwan, R.O.C. His research interests include machine learning, data mining, deep learning and security.



Dong-Jie Wu received his M.S. degree in Department of Computer Science and Information Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C. in 2012. He is currently a deputy director in Cybersecurity Technology Institute, Institute for information industry, Taiwan, R.O.C. His research interests include cyber threat hunting, incident response, penetration testing and machine learning.