Cybersecurity

RESEARCH

Open Access

TriCTI: an actionable cyber threat intelligence discovery system via trigger-enhanced neural network

Jian Liu[1,2], Junjie Yan[1,2], Jun Jiang[1], Yitong He[1,2], Xuren Wang[3], Zhengwei Jiang[1,2*], Peian Yang[1,2] and Ning Li[1,2]

Abstract

The cybersecurity report provides unstructured actionable cyber threat intelligence (CTI) with detailed threat attack procedures and indicators of compromise (IOCs), e.g., malware hash or URL (uniform resource locator) of command and control server. The actionable CTI, integrated into intrusion detection systems, can not only prioritize the most urgent threats based on the campaign stages of attack vectors (i.e., IOCs) but also take appropriate mitigation measures based on contextual information of the alerts. However, the dramatic growth in the number of cybersecurity reports makes it nearly impossible for security professionals to find an efficient way to use these massive amounts of threat intelligence. In this paper, we propose a trigger-enhanced actionable CTI discovery system (TriCTI) to portray a relationship between IOCs and campaign stages and generate actionable CTI from cybersecurity reports through natural language processing (NLP) technology. Specifically, we introduce the "campaign trigger" for an effective explanation of the campaign stages to improve the performance of the classification model. The campaign trigger phrases are the keywords in the sentence that imply the campaign stage. The trained final trigger vectors have similar space representations with the keywords in the unseen sentence and will help correct classification by increasing the weight of the keywords. We also meticulously devise a data augmentation specifically for cybersecurity training sets to cope with the challenge of the scarcity of annotation data sets. Compared with state-of-the-art text classification models, such as BERT, the trigger-enhanced classification model has better performance with accuracy (86.99%) and F1 score (87.02%). We run TriCTI on more than 29k cybersecurity reports, from which we automatically and efficiently collect 113,543 actionable CTI. In particular, we verify the actionability of discovered CTI by using large-scale field data from VirusTotal (VT). The results demonstrate that the threat intelligence provided by VT lacks a part of the threat context for IOCs, such as the *Actions on Objectives* campaign stage. As a comparison, our proposed method can completely identify the actionable CTI in all campaign stages. Accordingly, cyber threats can be identified and resisted at any campaign stage with the discovered actionable CTI.

Keywords: Actionable cyber threat intelligence, Campaign trigger, Indicators of compromise (IOCs), Natural language processing (NLP)

Introduction

Cyberspace security attacks (e.g., zero-day attack, advanced persistent threat) have been increasingly more sophisticated, destructive, and dangerous (Singh et al. 2019). In this situation, once a malicious attack vector is identified using actionable CTI, SOC (Security Operation Center) teams may immediately take effective measures

*Correspondence: jiangzhengwei@iie.ac.cn
[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
Full list of author information is available at the end of the article

(Samtani et al. 2020), which can contain the immediate damage.

Generally, indicators of compromise (IOCs) are the forensic artifacts of an intrusion (Liao et al. 2016), such as communication domains, downloader hashes, etc. Cyber threat intelligence (CTI) is the knowledge that provides context like what their motivation and capabilities are and what IOCs in victim's systems to look for (Zane 2021). More importantly, the CTI needs to be perceived as actionable to convey a richer context of IOCs by revealing their campaign stages. We adopt the five campaign stages of cyber threat intelligence referring to Hutchins et al. (2011), Yadav and Rao (2015): *Delivery, Exploitation, Installation, Command and Control, Actions on Objectives*. What calls for special attention is that sentences that do not clearly describe the campaign stage but introduce malicious behaviors are determined as *Malicious*. And we define the sentence describing the non-malicious behavior as *Benign*. Consequently, IOCs can be specified as attachment hash or compromised IP in *Delivery* stage, exploit site URL in *Exploitation* stage, command and control (C&C) server domain in *Command and Control* stage, and data exfiltration URL in *Actions on Objectives* stage, etc.

Actionable CTI can provide incident response teams with actionable insights and recommendations to stay nimble and precise in decision-making and taking effective actions (e.g., blocking the malicious domain or cleaning the underlying vulnerability) (Jeff 2021). Specifically, one of the essential applications of actionable CTI is that it has higher interpretability, which can guide the security professionals to distinguish the urgency of the cyber threats and promptly develop defensive countermeasures. Another instructive opinion is that if actionable CTI is integrated into intrusion detection systems, SOC teams can take appropriate mitigation actions based on contextual information of the alerts. For example, in the *Command and Control* stage, it is necessary not only to block malicious domains, but also to detect potential infection entry points (vulnerabilities) to the system. In the *Delivery* stage, on the other hand, it is sufficient to simply prevent users to access malicious phishing URLs, because at this point the attacker does not gain the initial access of the victim host.

Challenges. 1) Existing CTI feeds are significant limitations for their purported goals (Li et al. 2019; Bouwman et al. 2020). There are many security vendors in the industry that provide commercial threat intelligence services. However, open-source intelligence, such as CleanMX (2021), only provides a blocklist. Our motivation is to automatically discover actionable CTI, so the existing open or paid threat intelligence is out of the scope of our study. 2) The CTI automatically extracted is incomplete and has no actionable insights. There are many efforts focusing on how to convert unstructured texts (e.g., cybersecurity reports) into machine-readable data using automated methods. These cybersecurity reports are regularly released by security vendors, such as Kaspersky and FireEye, which proactively share the specific tactics, techniques, and procedures (TTPs) of the attacker and the associated IOCs. Some works such as Liao et al. (2016), Zhou et al. (2018), Long et al. (2019) only extract IOCs without campaign stages, others just the threat actions information (Husari et al. 2017, 2018). These efforts do not correlate with IOCs and the campaign stages they belonged to. 3) Limitations on feature engineering. Even IOCs and the corresponding campaign stages are extracted, the method adopted neural network is limited by the cumbersome feature engineering (Zhu and Dumitras 2018), which may introduce false positives. 4) For threat intelligence discovery, high-quality training data is severely scarce. Because the annotator needs to take into account the security professional background and the relevant work experience in NLP, and the annotation process is time-consuming and labor-intensive.

Our study. To solve the above challenges, we propose trigger-enhanced cyber threat intelligence (TriCTI) discovery system, which aims to automatically discover actionable CTI. Our method takes the cybersecurity report as the input raw material, and then TriCTI chooses the sentences where the candidate IOCs are located, and finally determines the campaign stage of the IOCs by classifying the campaign stage of the sentences. As shown in Fig. 1, the first sentence will be selected by TriCTI because of the candidate domain *d.heheda.tk*, then the sentence is classified as a *Command and Control* stage and thus the domain *d.heheda.tk* is determined as the *Command and Control* stage.

Intuitively, security professionals can recognize the campaign stage in a sentence based on highly explanatory words or phrases, which we define as a "campaign trigger". For the sentence "*The infection chain started with an email and an attached malicious word document*", we can infer from "*infection chain*" or "*an email and an attached malicious word document*" that the sentence describes the *Delivery* stage, which means the transmission of the cyber weapon to the targeted host. Naturally, the hash value in the sentence is determined as the *Delivery* campaign stage (see the second sentence of Fig. 1). We first train the trigger vectors of the training set, based on the principle that the trigger phrases can not only express a certain campaign stage with high interpretability but also match the keywords representing the campaign stage in the unseen sentences of the test set. When an unseen sentence is given, the most similar trigger vector can be used to enhance the weight of keywords in the unseen

Its hardcoded C2 domain is: d.heheda.tk$_{Command\ and\ Control}$.

As mentioned above, the infection chain started with an email and an attached malicious word document b98abdbdb85655c64617bb6515df23062ec184fe88d2d6a898b998276a906ebc$_{Delivery}$.

The first version of the skimmer used in this campaign is the hex obfuscated type with data exfiltration via autocapital.pw$_{Actions\ on\ Objectives}$ as seen in the decoy Rocket Loader library.

Suspected TEMP.Veles incidents include malicious activity originating from 87.245.143.140$_{Malicious}$, which is registered to CNIIHM.

Fig. 1 Examples of different campaign stages. The red font represents the IOCs, and the underlined words represent the trigger phrase in the sentence, that is, keywords that can emphasize the category of the sentence. The phrase that follows in the lower right corner of the IOC indicates the campaign stage of the IOC and the campaign stage to which the sentence belongs

sentence, thereby improving the classification performance. This approach is not limited by laborious feature engineering but instead allows the model to better absorb the features of the campaign stage.

Moreover, we manually annotate a gold-standard corpus with trigger phrases for model training, which requires considerable time and manpower. In particular, we devise a data augmentation method specifically for cybersecurity retrofitting CBERT (Conditional Bidirectional Encoder Representations from Transformers) (Wu et al. 2019) without breaking the label compatibility, modifying trigger words, and altering IOCs. Because we believe that the trigger phrase represents the emphasis of the campaign stages of the sentence, it cannot be altered. More importantly, the type of IOCs emphasized varies between the different stages of the campaign. For example, the hash is only one type of IOCs in the *Installation* stage, and the *Command and Control* stage frequently mentions IP and domain. Data augmentation can effectively alleviate the shortcomings of the small size of high-quality cybersecurity corpora and will prevent overfitting and improve the generalization of TriCTI.

Contributions. In general, we make the following contributions:

- We propose TriCTI for exploring the actionable CTI concerning the sentence contained putative IOCs from cybersecurity reports. Compared with state-of-the-art classification models, such as BERT, our model introduces an elaborate trigger vector that can best express the campaign stage and improve the weight of essential keywords in the sentence, thus the trigger-enhanced classification model has better performance.
- We present a total of 3167 sentences and 3012 trigger phrases covered five campaign stages for campaign stages classification. To alleviate the lack of annotated data in cybersecurity, we design a data augmentation method to improve the generalization performance of the model. We share the corpus[1] for in-depth study of threat intelligence community.
- We conduct a set of experiments on the annotated dataset and augmented dataset, and prove the effectiveness of our model. Particularly, compared with industry practice (VirusTotal) to verify the actionability of the discovered CTI, the experimental results demonstrate that TriCTI achieves competitive performance for completely exploring campaign stages.

Roadmap. The remainder of our paper is structured as follows: We first discuss the literature review in Section 2. Then we introduce the design and implementation of TriCTI architecture in Section 3. Section 4 gives a detailed description of experiments and results. Section 5 gives analysis and discussion of our devised data augmentation method, attention distribution exploration, comparison with industry practices, constraints for TriCTI, generality analysis, and limitation of the study and our future work. Ultimately, the paper ends with concluding remarks in Section 6.

Related works

Threat intelligence

IOCs are often used as forensic artifacts to detect attacks of threat organizations. Specifically, IOCs are hash values of malicious samples or IP addresses of C&C servers, etc. To allow IOCs to be quickly shared for defense, relevant security agencies have proposed some threat intelligence expression and transmission specifications, such as STIX (Structured Threat Information eXpression) (2021), MAEC (Malware Attribute Enumeration and Characterization) (2021). CAPEC (Common Attack Pattern Enumeration and Classification) (2021) and ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) (2021) describe the threat ontology of attack techniques and patterns. Cyber Kill Chain (Hutchins et al.

[1] https://github.com/lingren0/TriCTI.

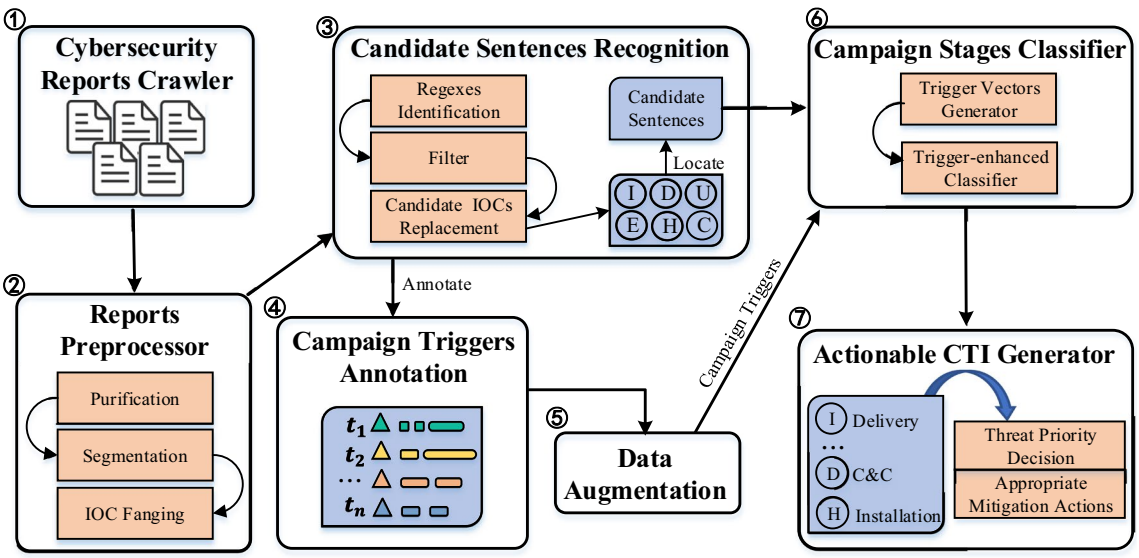


Fig. 2 The overall architecture of TriCTI system. IOC Types: IP address (I), Domain (D), URL (U), Email address (E), Hash (H), CVE number(C). ***t*** stands for campaign trigger

2011) is a model proposed by Lockheed-Martin that describes 7 attack stages, which can help defenders provide different defense and mitigation strategies for each attack stage. Although these specifications describe the context of threat intelligence in detail, few intelligence sources can provide a wealth of information about the campaign stages. For example, some public available intelligence, such as AlienVault (2021), can be provided in STIX format, but it lacks the campaign stages and shares unclassified IOCs. Instead, our TriCTI system can automatically extract IOCs and their associated campaign stages from unstructured cybersecurity reports.

NLP for cybersecurity

The application of NLP in the field of cybersecurity has many challenges. 1) Dependency vanishes. The IOCs are too far away from their associated context description. 2) The challenge of high-quality ground truth training data. It is particularly so for threat intelligence discovery. There are many advanced efforts (Dong et al. 2019; Dionísio et al. 2019) dedicating to mining reports for vulnerability information that can provide the warning. Dong et al. (2019) adopt named entity recognition and relationship extraction technology to extract the name and version number of the vulnerable software from the cybersecurity reports. Dionísio et al. (2019) develop a BiLSTM network to extract vulnerability-related information from tweets to serve security warnings. There are also many works on extracting IOCs, such as the studies of Zhou et al. (2018) and Long et al. (2019). They apply neural-based sequence labeling to extract IOCs from unstructured text, which lacks contextual semantic information. Liao et al. (2016) propose an automated IOCs extraction solution to analyze the IOCs and their contextual terms in security blogs. The identified IOCs do not involve the stages of the campaign. In addition to the basic IOCs information, there is also an effort to perform high-level threat actions discovery. Husari et al. (2018) use the metrics of entropy and mutual information to extract low-level attack actions from publicly available cybersecurity reports. Similarly, Husari et al. (2017) proposes a pre-defined ontology model to automatically extract threat actions from threat reports. This model relies heavily on custom ontology. In addition, one research (Satyapanich et al. 2020) focuses on the extraction of cybersecurity events from the text. A notable effort on discovering campaign stages is recently introduced by Zhu and Dumitras (2018) where their model relies heavily on manually predefined rules, which may introduce false positives and is limited to a small size of training sets. Our system TriCTI focuses on the campaign triggers related to IOCs and uses an advanced neural network algorithm model. This approach is not limited by laborious feature engineering but instead allows the model to better absorb the semantic information of the campaign stage. More importantly, we have adopted a well-designed data augmentation method so that the model is not limited to a small sample set.

Design and implementation

In this section, we will describe the pipeline of TriCTI with more details. As shown in Fig. 2, we first introduce the preparation of the classification model for the

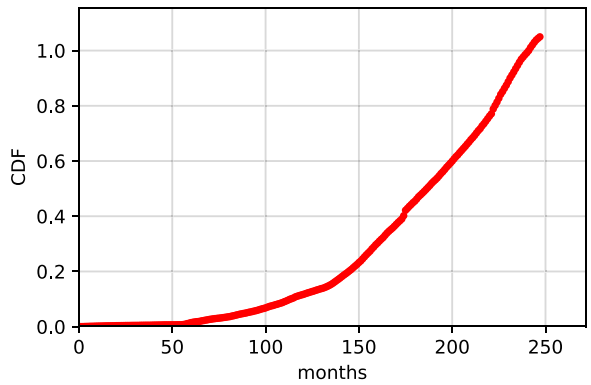


Fig. 3 The CDF of the number of cybersecurity reports published each month

campaign stages, including cybersecurity reports crawler and reports preprocessor. And then, we automatically filter candidate sentences that potentially describe the campaign stages and infrastructures (i.e., IOCs) using regexes. Subsequently, we annotate the campaign triggers and apply data augmentation to improve the generalization performance of the model. After that, we feed the annotated dataset to train the trigger vectors as well as the trigger-enhanced classification model. Finally, we will discuss the actionable CTI generator. We would then elaborate on the overall architecture of the TriCTI system.

Cybersecurity reports crawler

To highly cover the scope of cyberspace security detection, we need to crawl unstructured cybersecurity reports from a large range of high-quality security vendors. Thus, we collect a total of 29,686 cybersecurity reports covering 22 elaborately selected mainstream security vendors (such as Kaspersky, Symantec, and Fireeye) in the past 21 years (from November 24, 2000 to September 13, 2021) to support our analysis and research. We choose these security vendors because 1) their reports are related to campaigns, and 2) these reports describe IOCs used by Internet miscreants. Figure 3 depicts the cumulative distribution function (CDF) of the number of reports published each month. It can be seen that as time grows, the number of reports released every month gradually increases. It is an impossible task to extract actionable CTI from these reports only manually.

Reports preprocessor

- **Purification.** We convert the cybersecurity reports into pure text, that is to say, we remove the HTML tags that damage the performance of the model.
- **Segmentation.** Subsequently, we apply the Spacy toolkit (2021) to split the report into sentences.
- **IOC Fanging.** We convert IOCs that appear in the report from a defanged form to the normal and original form. For instance, some security professionals use "hxxp" instead of "http" in URL, and "[.]" or "(.)" instead of "." in the IP address to prevent users from clicking malicious links. Therefore, regular expressions (regex) are used to remove the anti-misclick symbols in reports.

Candidate sentences recognition

In the cybersecurity reports, there will be some normal software patch hashes or emails of the security vendor, which may be misclassified as IOCs. In real application scenarios, these benign indicators will cause a high rate of false positives when applied to intrusion detection, which is disruptive to an enterprise (Li et al. 2019). To address the above-described challenges, our intuition is that the model should strictly filter the identification of IOCs.

- **Regexes Identification.** Specifically, we first employ regular expressions to match candidate IOCs.
- **Filter.** Furthermore, we filter the Alexa (2021) top-level domain and intranet IPs to filter benign IP/Domain/URL, thereby reducing false positives.
- **Candidate IOCs Replacement.** The remainder of candidate IOCs can locate the candidate sentences containing putative IOCs. What is remarkable is that the characters of IOCs often have no linguistic meaning. Particularly, there are some special symbols in IOCs, such as ":" and "//" in URL. The above peculiarity will affect the word segmentation or feature extraction in the NLP task. Consequently, we replace IOCs with "IOC" (e.g., "8.8.8.8" → "IP").

Subsequently, the candidate sentences are fed into the campaign stage classification model. As shown in Fig. 2, we introduce 6 types of IOCs here: IP, domain, URL, hash, email address and CVE (Common Vulnerabilities and Exposures) number.

According to the contextual semantics of candidate IOCs, the trigger-enhanced classification model can classify the candidate IOCs into a different category. Notably, only when IOCs are divided as a certain campaign stage can they guide the priority of defense in intrusion detection.

Campaign triggers annotation

Campaign triggers can significantly explain the campaign stage of IOCs so that they can make our model more

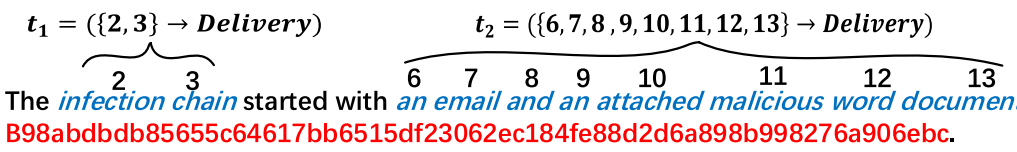


Fig. 4 An example of *Delivery* campaign stage. The red font represents the IOC, and the blue font is the campaign trigger

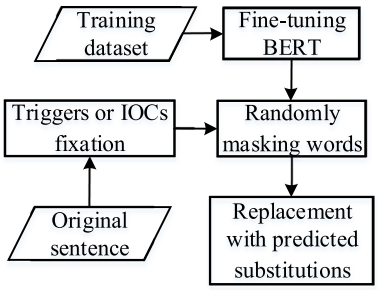


Fig. 5 Data augmentation process

generalize efficiently. We define the campaign trigger as $t = (\{w_i, w_{i+1}, ..., w_{i+q}\} \rightarrow c)$. w_i represents i-th word of the input sequence, q is the number of trigger words, and c is the campaign stage to which the trigger phrase belongs. Take Fig. 4 as an example for illustration, this sentence describes the *Delivery* campaign stage. Triggers "*infection chain*" can be described as "$t_1 = (\{2, 3\} \rightarrow Delivery)$" and "*an email and an attached malicious word document*" is "$t_2 = (\{6, 7, 8, 9, 10, 11, 12, 13\} \rightarrow Delivery)$". We annotate campaign triggers based on this criterion.

Data augmentation

We retrofit CBERT (Wu et al. 2019) to augment the training dataset without breaking the label compatibility, modifying trigger words, and altering IOCs. CBERT uses fine-tuned conditional BERT to predict label-compatible words. That is, the fine-tuned model takes the polarity of the sentence into account. Since we use the trigger to improve the performance of the model, the trigger phrases in the sentence cannot be replaced by the substitutions randomly predicted and generated by BERT. And the type of IOCs varies between the different stages of the campaign, therefore, when we introduced the conditional masked language model, we need fix the trigger phrase and IOCs, and only replace other words. Figure 5 describes the process of data augmentation using retrofitted CBERT.

Campaign stages classifier

With the recognized candidate IOCs and sentences from the reports, we next generate actionable threat intelligence by determining the campaign stages incorporating the campaign triggers. Specifically, for a sentence s, there will be a set of triggers $\bigtriangleup = \{t_1, t_2, ..., t_k\}$ (as shown in Fig. 4). To make the model training more effective, we redefine the input of the model to one sentence, one trigger, and their output y, which is expressed as (s, t_k, y). y represents the ground-truth label of the campaign stage. After training the model and the trigger vectors, we introduce the trained trigger vectors, instead of labeled trigger phrases, into our test set to verify performance.

Trigger vectors generator

In this process, we will train the vector representations of the triggers. Our method is inspired by Lin' work (Lin et al. 2020), which trains trigger vectors using bidirectional long short term memory (BiLSTM) for named entity recognition (NER). However, in the field of text classification tasks, BERT (Devlin et al. 2018) is proven to have significant experimental effects due to the multilayer bidirectional transformer (Vaswani et al. 2017). Therefore, in our paper, we apply fine-tuning BERT with elaborate design to generate trigger vectors.

We use the following heuristic to derive the trigger vectors.

- **Classification loss** L_{cls}. Generally, we introduce "campaign trigger" to strengthen the classification model because it is more explanatory for the campaign stage. Therefore, the trigger vector representation needs to be trained through the campaign stage classification model.
- **Contrastive loss** L_{sim}. In parallel, trigger representation should be similar to the unseen sentence representation, so when classifying unseen sentences, trigger vectors with strong similarity to the unseen sentences can be assimilated to enhance the classification of the model. Notably, contrastive loss (Hadsell et al. 2006) is used here to compare the similarity between sentence and trigger.

Given the sequence token of the input sentence s, the last hidden state H_l and pooler output state H_p can be obtained by the BERT model. Accordingly, the trigger vector G can be obtained from H_l based on the position of the trigger phrase in the sentence. Trigger vectors are

jointly trained by campaign stage classification and similarity calculation. Algorithm 1 describes the joint training of trigger vectors.

Algorithm 1 The joint training of trigger vectors

Input: Training set $D_T = \{(s,t,y)_i\}$, $i \in [1,n]$, n is the number of training set.
Output: Trigger vector $T = \{r_{t_1}, r_{t_2}, ..., r_{t_n}\}$.
for $i = 1$ to n **do**
  $(H_l, H_p) \leftarrow BERT(s)$;
  $G \leftarrow Position(t, H_l)$;
  $r_s \leftarrow SelfAttention(H_l)$;
  $r_t \leftarrow SelfAttention(G)$;
  $L_{cls} \leftarrow$ ClassificationLoss (r_t, y);
  $L_{sim} \leftarrow$ ContrastiveLoss (r_t, r_s);
  $L \leftarrow L_{cls} + L_{sim}$;
end for
Algorithm convergence.
return T

In Algorithm 1, the self-attention (Lin et al. 2017) is applied to generate the final sentence vectors and trigger vectors. The specific calculation formulas are as follows:

$$\begin{aligned} a_s &= softmax(W_{s_2} tanh(W_{s_1} {H_l}^T))) \\ r_s &= a_s H_l \end{aligned} \tag{1}$$

$$\begin{aligned} a_t &= softmax(W_{t_2} tanh(W_{t_1} G^T))) \\ r_t &= a_t G \end{aligned} \tag{2}$$

As for the learning of campaign stages for trigger vectors, the exact cross-entropy loss function is to evaluate the multi-stages campaign classification performance of the models.

$$L_{cls} = -\sum_{i}^{n}\sum_{j}^{c} y_i^j log(\hat{y}_i^j) \tag{3}$$

where y_i^j is the ground-truth label; $\hat{y}_i^j$ is prediction probabilities; n and c denote the number of samples and classes, respectively.

Simultaneously, the trigger vector needs to be further trained according to the similarity between trigger and sentence containing the trigger. The parameterized distance function is defined as D. Subsequently, as mentioned in Hadsell et al. (2006), we define γ as a binary label. $\gamma = 0$ if r_t and r_s are deemed similar, and $\gamma = 1$ if they are deemed dissimilar. The contrastive loss function is:

$$D = \left\| r_s - r_t \right\|_2 \tag{4}$$

$$L_{sim} = (1-\gamma)\frac{1}{2}(D)^2 + (\gamma)\frac{1}{2}\{max(0, m-D)\}^2 \tag{5}$$

where $m > 0$ is a margin for the negative examples.

Finally, the joint loss of the triggers representation learning is,

$$L = L_{sim} + \alpha L_{cls} \tag{6}$$

where α is a hyper-parameter to balance the loss functions.

Through the joint training of equation 6, the trigger vector is not only highly explanatory for the campaign stage (that is, the trigger phrase in the sentences that plays an essential role in the classification of the campaign stages has a higher attention score), but also most similar to the unseen sentence (that is, given a sentence at random, certainly, the sentence does not appear in the training set, a trigger vector can be selected according to the ranking of the similarity between the trigger vectors and the sentence). Specifically, this elaborate trigger vector is similar to an essential keywords in the unseen sentence that can best express the campaign stage and improve the weight of essential keywords in the classification process.

Trigger-enhanced classifier

So far, we have obtained a final set of trained trigger vectors T. Next, we will introduce how to use these trigger vectors to enhance our campaign stages classification model. The architecture of the trigger-enhanced campaign classification model is depicted in Fig. 6.

As mentioned in the previous section, given a sentence s, we can obtain H_l and H_p using BERT, and then the sentence vector representation r_s and trigger vector representation r_t are obtained through the self-attention mechanism. Differently, we no longer use the trigger representation r_t but instead use r_{t_k}, which comes from the set of trigger vectors we trained in the previous section and is the most similar to the sentence vector r_s. Because no labeled trigger phrase is applied in a large-scale cybersecurity report.

As described in algorithm 2, independently using equation 4, the most similar campaign triggers r_{t_k} to the sentence representation r_s can be obtained. Then, to learn alignments between H_l and r_{t_k}, they are input into dot-product attention to infer a weight vector a and create a trigger-enhanced sequence of token representations H_l'.

$$\begin{aligned} M &= tanh(W_1 H_l + W_2 r_{t_k}) \\ a &= softmax(u^T M) \\ H_l' &= a H_l \end{aligned} \tag{7}$$

Empirically, the types of IOCs contained in the sentence are also import for model, and we consider them as $IOC_{feature}$. For example, the hash is only one type of IOCs in the *Installation* stage, and the *Command and Control* stage frequently mentions IP and domain. Therefore, we

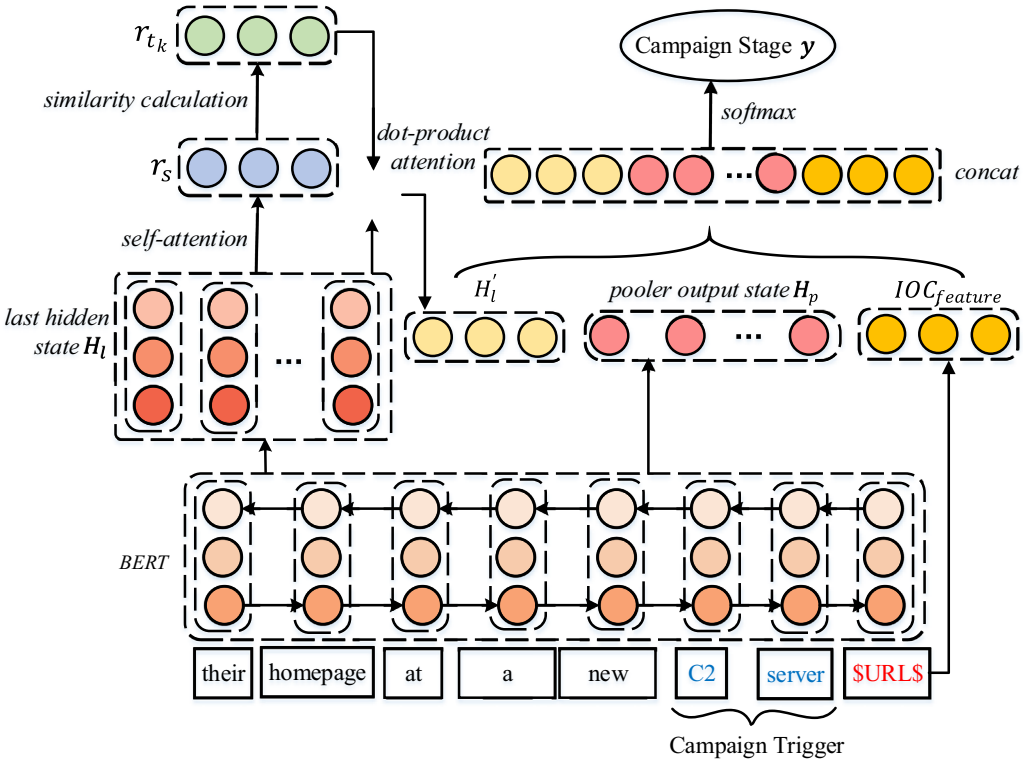


Fig. 6 The demonstration of the trigger-enhanced campaign stages classification model. r_{t_k} is the most similar trigger vector to final sentence vector r_s, which is highly explanatory for the campaign stage. The final input of the model is: the type features of the IOCs $IOC_{feature}$, weighted hidden representation H_l', and H_p. Note: the original value of "URL" is "http://5.79.66.241/index.html"

consider the types of IOCs that appear in the sentence and construct them into one-hot encoding. For example, if the token match the regex of URL, then we set the value of the URL feature to 1; otherwise, the value is 0.

Finally, we concatenate these feature vectors $[H_p, H_i', IOC_{feature}]$, and then input them into the softmax layer for training the sentence campaign stage classification. The complete pseudo-code for the trigger-enhanced campaign stage classification model is depicted as algorithm 2.

Algorithm 2 Trigger-enhanced campaign stage classification model

Input: Training set $D_T = \{(s, y)_i\}$, $i \in [1, n]$, trained trigger vectors $T = \{r_{t_1}, r_{t_2}, ..., r_{t_n}\}$.
Output: Campaign stages of sentences $\{y_1, y_2, ..., y_n\}$.
1: **for** $i = 1$ to n **do**
2: $IOC_{feature} \leftarrow$ generate a list of IOC types present in s_i;
3: $(H_l, H_p) \leftarrow BERT(s)$;
4: $r_s \leftarrow SelfAttention(H_l)$;
5: $r_{t_k} \leftarrow Distance(r_s, T)$ //get the most similar campaign trigger vector to r_s;
6: $H_l' \leftarrow Attention(H_l, r_{t_k})$;
7: $y_i \leftarrow Classify([H_p, H_l', IOC_{feature}])$;
8: **end for**
9: **return** $\{y_1, y_2, ..., y_n\}$

Actionable CTI generator

If the candidate sentence is classified into a certain campaign stage, then we will obtain the candidate IOC's classification result, and the original value of the IOC in the sentence is restored. For instance, the placeholder "$Hash$" is replaced with the hash value 791ad58d9bb66ea08465aad4ea968656c81d0b8e. And then we combine the IOC with the campaign stage to obtain actionable CTI. Applying actionable CTI to intrusion detection systems can guide security operators to make faster, better decisions.

- **Threat Priority Decision.** In Lockheed Martin's Kill Chain model (Hutchins et al. 2011), an IOC in the *Command and Control* stage is more dangerous than the IOC in the *Delivery* stage. That is, the *Delivery* stage is merely to gain access to the victim's host, while the *Command and Control* stage is closer to the attacker's final intention, which is more destructive and serious. In this case, security professionals will give priority to mitigating the cyber attack in the *Command and Control* stage. The sooner the detection is done, the less loss the organization under attack will suffer (Yadav and Rao 2015).

- **Appropriate Mitigation Actions.** Once malicious communication is detected, the C&C server address should be added to the firewall of the victim host to block the connection. Please do not take it lightly, before the attacker controls the victim host, there may exist a security breach that is compromised by the attacker. If it is not carefully cleaned up, the vulnerable host will get compromised again (De Silva et al. 2021). However, if the victim only receives a phishing email, it only needs to prevent the victim from opening the malicious link, because the attacker may not get the initial access of the victim host.

Experiments

In this section, we use the annotated corpus as the ground truth to evaluate the performance of our model. Then we apply the model to a large number of cybersecurity reports.

Datasets

Annotating security corpus is difficult due to 1) the complexity of annotation work, and 2) the security professional background and relevant work experience in NLP. And for this reason, three security professionals with NLP-related work experience help annotate the dataset. Albeit they have relevant professional experience, annotating our corpus of campaign stages still requires meticulous definition and delimitation. These three professionals, therefore, co-define the annotation specification by prior consultation and strictly comply with them. In a subsequent step, to minimize the manual annotation efforts, a semi-automated annotation method was employed. Specifically, we first randomly select reports from the collected cybersecurity reports (from 2013 to 2021) for corpus annotation. Then we use the method shown in Fig. 2, that is, regular expressions are applied to locate candidate IOCs and their sentences. As such, professionals directly categorize the sentences containing candidate IOCs and annotate the trigger words. The professionals spent a week annotating the corpus, and we also spent another week to ensure the correctness of the annotation results. Specifically, we aggregate the annotations from the three professionals and take a majority voting. For annotations that consensus was not reached (i.e., three different viewpoints), we incorporate the opinions of front-line security researchers.

We meticulously construct two different datasets, DS-1 (2013 to 2020) and DS-2 (2021) that are temporally disjoint and different window sizes to train trigger-enhanced classification models and show their generality. As shown in Table 1, a total of 2,362 sentences are annotated as ground truth in DS-1 and we use these sentences to train the campaign stages classification model. We divide the training set and the test set according to the ratio of 7:3 and only perform data augmentation for the training set. TriCTI performs better when the data is augmented to around 700 per category. After data augmentation, we obtain a total of 4,900 labeled sentences. As for DT-2, a total of 805 sentences are annotated, and more notably, no trigger phrases are annotated. We share all annotated corpus, hoping to help more security professionals in their research. Among DS-1 and DS-2, a total of 3,468 IOCs have corresponding campaign stages.

Table 1 Statistics of annotation

Category	DS-1			DS-2
	# of Sen.	# of Aug.	# of Triggers	# of Sen.
Delivery	348	700	449	111
Exploitation	304	700	392	105
Installation	316	700	371	105
Command and Control	401	700	472	125
Actions on Objectives	303	700	388	111
Malicious	351	700	413	129
Benign	339	700	527	119
Total	2362	4900	3012	805

# of Sen. indicates that the number of original annotation sentences (split into DS-1 and DS-2 datasets respectively basing on temporal order), *# of Aug.* represents the number of sentences after data augmentation (only the training set is augmented), and *# of Triggers* is the number of annotated triggers

Experiment setup

By contrast, we choose state-of-the-art text classification models as the baselines to prove the performance of the trigger-enhanced classification model, including BiLSTM (Tang et al. 2015) (we uses the last hidden state vector of LSTM to predict campaign stages), attention-based BiLSTM (BiLSTM-Att) (Lin et al. 2017) (attention mechanism will relief the burden of LSTM to capture long term dependencies), and fine-tune BERT (Devlin et al. 2018). As for BiLSTM and BiLSTM-Att, we apply the combination of 50-dimensional character-level representation from a trainable BiLSTM model and 300-dimensional pre-trained GloVe vectors (Pennington et al. 2014). And as for the TriCTI model and pure BERT, we use the uncased BERT-base English version (Devlin et al. 2018). We use Adam (Kingma and Ba 2014) as the optimizer for BERT with a learning rate of 2e-5. The TriCTI works better with a low dropout rate of around 0.4, and our batch size is 10. The experimental results of the above models are obtained by averaging the results of 3 runs. Accuracy, Precision, Recall, and Macro-Averaged F1 are adopted as the evaluation metrics.

Table 2 Comparison of campaign stages classification results

Model	Original training set				Augmentation training set			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
BiLSTM	81.23	81.45	81.39	81.27	81.27	81.54	81.55	81.38
BiLSTM-Att	80.76	81.31	80.99	80.94	82.02	82.38	82.17	82.10
BERT	82.99	83.77	82.93	82.87	83.63	84.15	83.68	83.65
TriCTI	86.38	86.55	86.57	86.43	86.99	87.16	87.10	87.02

We run the model on the original labeled training set and the augmented training set. Accuracy, Precision, Recall, and F1 score (macro) are the average value over 3 runs

Table 3 Ablation study results

Model	Precision	Recall	F1 (macro)
TriCTI	87.16	87.10	87.02
-BERT	81.00	79.83	79.98
-IOCs	87.05	86.38	86.41
-Trigger	85.80	85.41	85.42

The experimental results (average value over 3 repeated runs) are carried out on augmentation data

Overall performance

As shown in Table 2, the experimental effect on the data augmentation training set is higher than the original labeled training set in all models. This proves that data augmentation does improve the efficiency of the model. The F1 score of BiLSTM-Att gives still further improvement to 82.10% compared to BiLSTM's 81.38% in the augmentation training set. Unfortunately, it is slightly inferior to BiLSTM on the original dataset. The pure BERT outperforms all the about models in the original and augmented training set, demonstrating the power of its large pre-trained model (Devlin et al. 2018). Especially, after incorporating trigger vectors and IOCs features, the TriCTI model outperforms all these state-of-the-art neural network models and the F1 score is remarkably improved to 87.02%.

Ablation study

As shown in Table 3, we further conduct an ablation study to evaluate the level of benefit that each component of the TriCTI. '-BERT' denotes that the TriCTI replaces the BERT with BiLSTM to train sentence and trigger vectors, '-IOCs' means that TriCTI removes the IOC features, and '-Trigger' indicates that TriCTI does not introduce the pre-trained trigger vectors.

As shown in Table 3, the F1 score of '-BERT' drops to 79.98%, indicating that BERT is better at capturing richer semantic features in practice than BiLSTM. Removal of trigger features leads to performance drops considerably. This study validates that the introduction of the pre-trained trigger vectors can indeed improve model performance. Moreover, '-IOCs' has a lower score than TriCTI, which proves that adding the IOC features allows our model to be robust.

Analysis and discussion

Golden label study of augmentation

In Section 3.5, the guideline for data augmentation is that the original meaning of the sentence (i.e., the golden label) needs to be conserved. Although we fixed the trigger phrases and IOCs, we still face the possibility of changing the meaning of the sentence when the input sentence is altered. Therefore we use the test set to visualize whether the retrofitted CBERT can maintain the golden label of the input sentence. We first simply use pure BERT to train the campaign stage classification model on the original training set. Then use retrofitted CBERT to generate approximately three times the augmented test dataset, that is, one original sentence generates three augmented sentences. Subsequently, the original test set and the augmented test set are respectively fed to the trained BERT classification model to compare whether the meanings are conserved. Finally, t-SNE (Van Der Maaten 2014) is used to map the BERT model's last layer hidden-state to a 2-dimensional vector. Figure 7 is a visualization of the latent space representations for the two data sets.

It can be seen that the latent space representations of the augmented data in each campaign stage are closely adjacent to the original data. This shows that our retrofitted CBERT method can well retain the original meaning of the sentence without changing the golden label of the sentence.

Attention distribution exploration

It is enlightening to analyze which words decide the campaign stage of a sentence. As shown in Table 4, the motivating examples from the test set illustrate that the attention scores of triggers can improve classification performance. The Trigger phrase "*phishing campaign*" contributes to the *Delivery* campaign stage in the training set. Our model can calculate that the phrase "*spearphishing*" in the unseen sentence (a) is similar to trigger

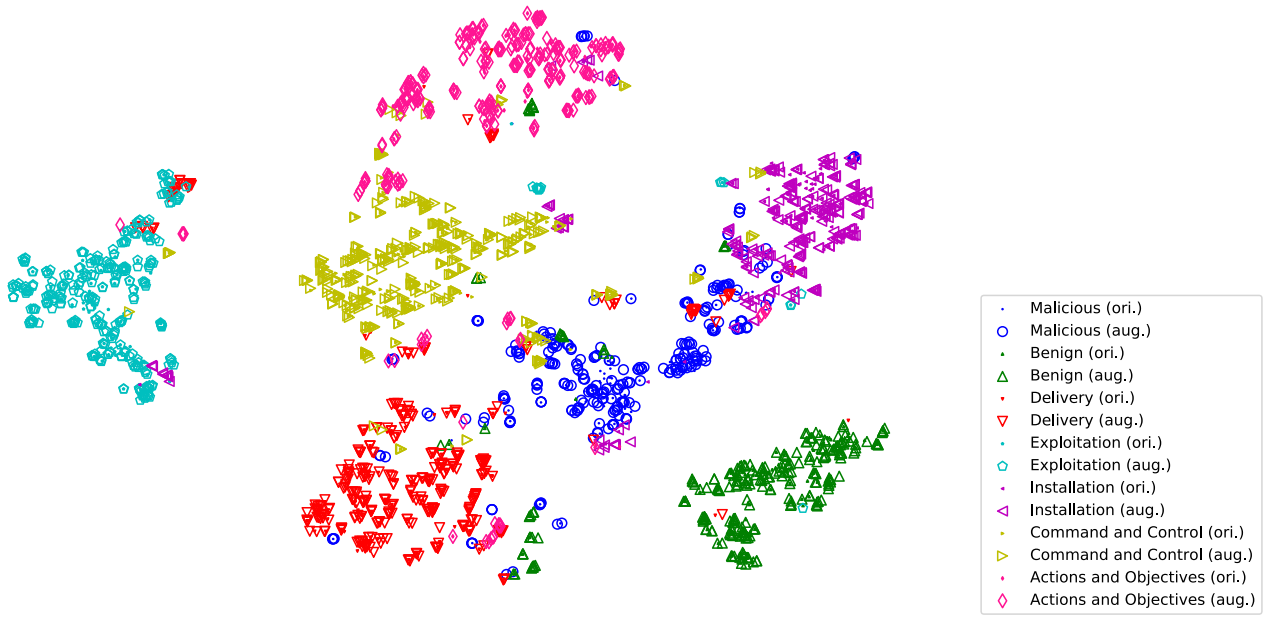


Fig. 7 Case study 1. The visualization of the latent space representations for original training set (smaller size symbol) and augmented training set (large size symbol). The same campaign stages have the same colors

Table 4 Case study 2. Attention Distribution Exploration. Darker cell color, meaning higher attention weight, demonstrates that the triggers contribute to the classification performance

	Attention visualization	Matching trigger	Label
a.	In both spear-phishing campaigns , the decoy document has been the exact same PDF file , a " US letter fax test page " 28d29c702fdf3c16f27b33f3e32687dd82185e8b .	phishing campaign	Delivery
b.	While it was serving the zero-day exploit , the IP address of ausameetings.com was 95.215.45.189	Adobe Flash exploits are	Exploitation
c.	SHA256 : 999c1d4c070e6817c3d447cf9b9869b63e82c21c6e01c6ea740fbed38b730e6e installs a Windows service called either " Microsoft Display Agent " or " Windows 10 Upgrader " .	executes it	Installation
d.	We also found several Dynamic Name Servers DNS , which at some point led to the same C&C IP address : hefklife.ddns.net fklife.ddns.net php.no-ip.biz ayalove.no-ip.biz .	C2 communications	Command and Control
e.	Shortly before then , the domain ' keybase.in ' , was registered as a homepage and online store for the KeyBase keylogger .	keylogger	Actions on Objectives
f.	Most prevalent malware files this week SHA 256 : e66d6d13096ec9a62f5c5489d73c0d1dd113ea4668502021075303495fd9ff82	indicate maliciousness	Malicious
g.	If you'd like to suggest an update or another Drupal security topic you'd like to have covered , get in touch with us at marketing@sucuri.net .	If you'd like to	Benign

"*phishing campaign*", therefore, adding trigger vector of "*phishing campaign*" to the classification model can make "*spear-phishing*" have a higher attention score to correctly distinguish the campaign stage. This situation can also be illustrated by other campaign phase examples.

Some benign indicators are incorrectly matched by regular expressions, as shown in the sentence (g). We also apply some benign triggers, such as the trigger phrase "*If you'd like to*". Similarly, this trigger matches the phrase "*you'd*" in an unseen sentence and makes the model more information to correctly determines the classification of the sentence.

Comparison with industry practices

Finally, TriCTI discovered 113,543 IOCs with their campaign stages from more than 29k cybersecurity reports

Table 5 TriCTI classification results statistics

Category	IP	Domain	URL	Hash	Email	CVE
Delivery	1359	4695	2692	767	280	–
Exploitation	314	561	593	360	–	21,698
Installation	–	–	–	3477	–	–
Command and control	2892	3675	7200	808	47	–
Actions on objectives	692	1398	1430	1754	148	–
Malicious	11,091	17,852	1498	16,301	396	–
Benign	3387	2312	1451	1037	1378	–
Total	19,735	30,493	14,864	24,504	2249	21,698

during the last 21 years. We elaborate on the statistics of classification results in Table 5.

To verify the actionability of our recognized CTI, large-scale analyses are applied to provide a quantitative measurement. In our paper, we apply VirusTotal (VT) (VirusTotal 2021) to perform verification experiments. VT is a service that users can submit suspicious files for scanning and analysis, and it provides API to query the *Relationships* (e.g., *contacted_domains* describes the domains contacted by the file) of a given IOC (e.g., malicious hash). Our *CRITERION* is to determine whether an IOC has a certain campaign stage by querying the *Relationships* of the IOC. For example, by querying whether the malware has a contacted IP (and the IP is malicious), we can determine whether it has the ability to connect back to C&C services (Other *Relationships* that can be mapped to the campaign stages, such as: *email_attachments* → *Delivery*, *dropped_files* → *Installation*, etc.). If at least one scanner in VT determines the IP to be malicious, then we consider it to be malicious.

Since the *Relationships* provided by VT are limited to support the verification of *ALL* campaign stages, we can only determine the *Delivery*, *Installation*, and *Command and Control* campaign stages of the hash, as well as the *Command and Control* campaign stages of the domain, IP, and URL using the API provided by VT to query the *Relationships* about the hash, IP, domain, and URL. Due to the high volume of our discovered IOCs and the query rate limit of VT, we intend to verify the 22,934 unique IOCs (including IP, domain, URL, and hash) and related campaign stages from 2018 to 2021. Figures 8 and 9 show the VT verification results of the discovered actionable IOCs with their campaign stages. Note that the relationship provided in VT to describe a certain campaign stage does not fully cover *ALL* malicious behaviors in that stage of the campaign. Therefore, VT can only verify *PART* of the performance of our model.

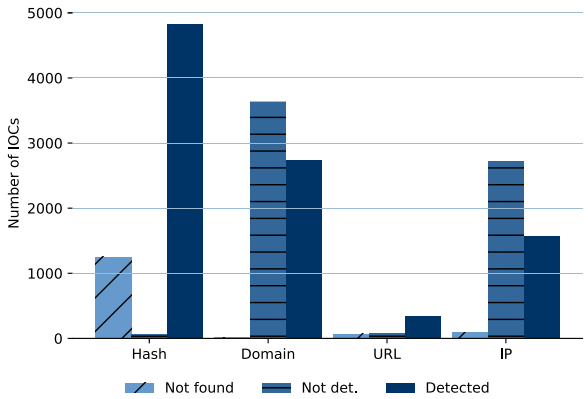
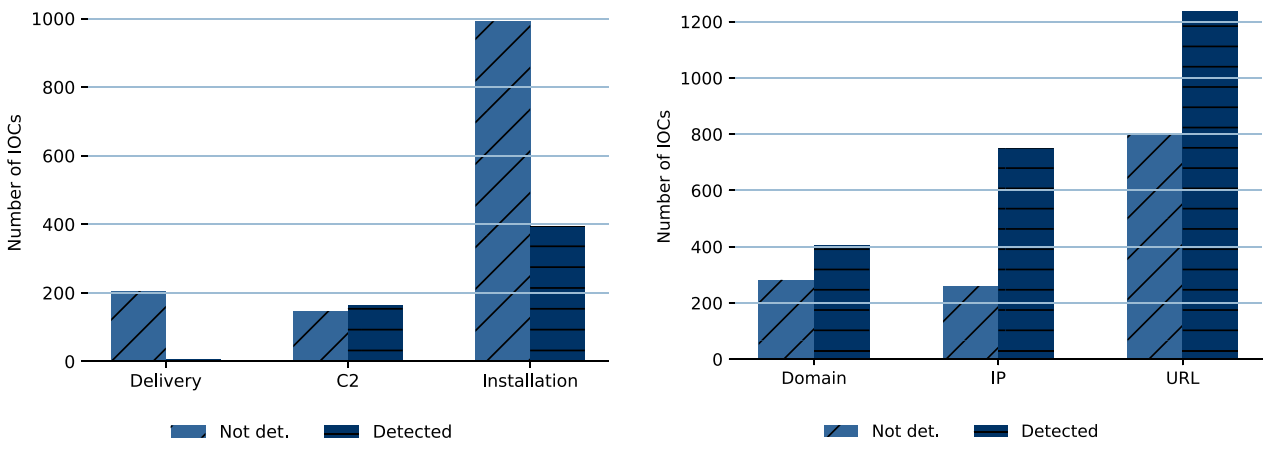


Fig. 8 The verification (using the VirusTotal analysis) for *Malicious* IOCs discovered from TriCTI. Note: *Not found* is a fraction of IOCs that are not found in VT, *Not det.* indicates that the IOCs are found in VT but none of the antivirus scanners mark it as malicious, and *Detected* represents the number of the IOCs that are found in VT and are labeled malicious by at least one antivirus scanners

As seen in Fig. 8, the VT results verified that most of the hashes and URLs labeled by TriCTI to be *Malicious* are also malicious in nature. This proves the ability of our model to discover malicious IOCs in the wild. However, a small part of the hashes and URLs are not found in VT, this probably indicates that these open-source intelligence vendors focus on threats that are not as relevant to most VT users' interest (Li et al. 2019). As for domain and IP, a significant portion are marked as no malicious behavior. Not like hash and URL, domain and IP are time-sensitive to the blocklist, that is, the IPs and domains in the blocklist after a period of time will change ownership (Lever et al. 2016) and may become benign.

Figure 9a, b show the verification about a part of campaign stages of our model. It is worth noting that some IOCs with their campaign stages marked by TriCTI may not be able to verify through the *Relationships* query interface (such as the *Not det.* hashes in the *Installation* stage in Fig. 9a). Intuitively, the *Relationships* lacks some



(a) The campaign stages of hashes. (b) The *Command and Control* campaign stages.

Fig. 9 The verification (using the VirusTotal analysis) for *Delivery*, *Installation*, and *Command and Control* campaign stages of the hash, as well as the *Command and Control* campaign stages of the domain, IP, and URL discovered from TriCTI. Note: *Not det.* indicates that the IOCs are found in VT but there are not existing any kind of malicious *Relationships* to indicate their campaign stages, and *Detected* represents the number of the IOCs that are found in VT and have at least one malicious *Relationships*

description of *Installation* behavior, so only part of the *Installation* campaign stages can be verified. For these IOCs that are not able to be verified in the campaign stages through *Relationships* (VT provides *Relationships* with limitations), we again use VT's query interface to verify whether they are malicious. If these fraction of IOCs are malicious, then it means that these IOCs have a high probability of being in the campaign stages (it's just that VT lacks the description of the *Relationships*). As shown in Fig. 10, among them, about 56.8% of the IOCs that are not able to be verified in the campaign stages through *Relationships* are malicious. This confirms that VT has limitations in identifying the *ALL* campaign stages of IOCs, and our model TriCTI can discover the whole campaign stages based on the description of malicious behavior about IOCs.

Constraints for discovery of cybersecurity campaign stages

While our basic goal is to discover actionable CTI, which reveals the campaign stage of IOC, we cannot use the previously shared corpus of security reports directly as we are obliged to introduce campaign triggers to enhance the classification performance of our model. Concretely, the sentence input to our model not only contains candidate IOCs but more importantly, the location of the trigger phrase needs to be pointed out. Table 6 summarizes how the different goals and additional specifications of previous works do not fully satisfy our requirements.

Overall, several works focus on the IOCs extraction (Liao et al. 2016; Zhou et al. 2018; Long et al. 2019; Kim et al. 2020, 2019), but either the type of IOCs extracted varied from our work (e.g., lack of Email) (Zhu and Dumitras 2018; Kim et al. 2020) or the contextual description of the IOCs was lacking (Liao et al. 2016; Zhou et al. 2018; Long et al. 2019; Kim et al. 2020, 2019). Moreover, there are papers that describe only part of our campaign phases, such as Baiting, Exploitation, Installation, Command & Control, and the corpus are publicly unavailable (Zhu and Dumitras 2018). On the whole, the quantity of papers that corpus is publicly shared is relatively small, and even if the dataset is available, it's not sufficient to satisfy our demands, so we constructed a cybersecurity corpus, which focuses

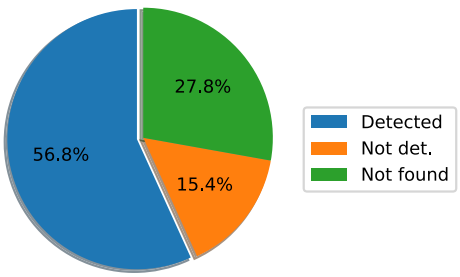


Fig. 10 The re-verification (using the VirusTotal analysis) for IOCs that are not able to be verified the campaign stages through *Relationships*. Note: *Not found* is a fraction of IOCs that are not found in VT, *Not det.* indicates that the IOCs are found in VT but none of the antivirus scanners mark it as malicious, and *Detected* represents the number of the IOCs that are found in VT and are labeled malicious by at least one antivirus scanners

Table 6 Overview of goals and additional specifications of CTI discovery

Goal	Public available	Campaign stages	IOC types	Related work
Interdependent relationships among heterogeneous IOCs	✓	✗	✓	Zhao et al. (2020)
IOC extraction	✗	✗	✓	Liao et al. (2016), Zhou et al. (2018), Long et al. (2019)
IOC extraction	✓	✗	Partially-compliant	Kim et al. (2020)
IOC extraction	✓	✗	✓	Kim et al. (2019)
IOC extraction and part of the campaign stages classification	✗	Partially-compliant	Partially-compliant	Zhu and Dumitras (2018)
Extraction of Threat Actions	✗	✓	✓	Husari et al. (2017)
Actionable CTI	✓	✓	✓	Our work

Table 7 Concept drift analysis of trigger-enhanced campaign stages classification model

Dataset	Precision	Recall	F1 (macro)
DS-1	87.16	87.10	87.02
DS-2	81.93	80.79	80.76

on trigger phrases, and the data of our study is available at: https://github.com/lingren0/TriCTI.

Model generality over time

As analyzed in Section 5.4, given that the uniqueness of our approach is the introduction of a triggered-enhanced neural network to discover actionable CTI, we cannot test the generality of our model with another publicly available dataset. Instead, we are more concerned about whether performance can remain generalizable across datasets of varying time spans. Actually, exploring whether the model can cope with concept drift (De Silva et al. 2021; Le Pochat et al. 2020) is quite important in practice. To measure the impact of concept drift on our trained trigger word vector (2013-2020), i.e., how much it will affect the performance of future practical work, we use DS-2 (2021) to evaluate the generalization performance of the DS-1 trained model.

As shown in Table 7, concept drift occurs that the performance of TriCTI drops on the DS-2 dataset. This suggests that new attack patterns are emerging over time and our model is unable to capture these changes. To keep high accuracy, the model should be retrained accordingly over time.

Limitations and future work

To analyze the limitations of TriCTI, We randomly validate the large-scale actionable intelligence discovered by the TriCTI and carefully select the misclassified examples to analysis. After careful manual inspection of bad cases, we discover that there are two primary reasons for the classification error. The details are shown in Table 8. One of the important reasons is the complexity of sentences describing cybersecurity attacks, that is, one sentence may contain multiple campaign stages, which leads to contradictions in the model. For example, sentence (a) describes two behaviors, one is installation and the other is remote communication. Another reason is

Table 8 Error analysis

Reasons	Examples	Analysis
Multi-label	a. b14d8faf7f0cbcfad051cefe5f39645f - dispci.exe installs the bootlocker, communicates with the driver	Two kinds of label conflict: *Installation* and *Command and Control*
	b. Via an associated C2 IP address 108.61.214.194, we found an equivalent page on the phishing domain www.battllestategames.com	Two kinds of label conflict: *Command and control* and *Delivery*
Incorrect association of IOCs with the campaign stages	c. We observed the sample in the sandbox launched a DDoS attack against 185.63.190.95 around 2017-04-23 21:45:00	185.63.190.95 is the IP address of the victim
	d. A typical representative of this malware family is an obfuscated Java script using ADODB.Stream technology to download and run DLL, EXE and PDF files	ADODB.Stream is not a domain address

that IOCs are incorrectly associated with the campaign stages. Our model can correctly determine the campaign stage according to the attack behaviors described by sentences (c) and (d), however, the campaign stage cannot be accurately associated with candidate IOCs identified by regular expressions. For example, sentence (c) describes the DDoS attack, but "85.63.190.95" is the IP address of the victim. "ADODB.Stream" in the sentence (d) is recognized as domain by regular expressions, but it is the name of an obfuscation technique.

In this paper, we use the trigger-enhanced model to classify the campaign stages of the sentences. After error analysis, it can be found that most fraction of the misclassified results are due to the fact that the campaign stage described by the sentence is not correlated with the IOC identified by the regular expression. Consequently, a useful future direction is to introduce dependency parse trees. We can construct a dependency parse tree by associating candidate IOCs with the contexts they depend on. In this way, the classification model can successfully determine the campaign stages related to the IOCs based on the well-designed dependency parse tree that expresses the contextual meaning of the IOCs. More interestingly, we also would like to specify a more fine-grained description of cyber threat intelligence campaign stages.

Conclusions

In this paper, we design and develop a trigger-enhanced system named TriCTI to discover actionable threat intelligence, that is, conveying a richer context of IOCs by revealing their campaign stages. So that we can obtain complete visibility across campaign stages. Specifically, we apply BERT to pre-train the trigger vectors that can explain the campaign stage and we also considered the features of the IOCs contained in the sentence to jointly improve the performance of the model. To cope with the challenge of the scarcity of annotation for cybersecurity corpus, we devise a data augmentation without breaking the label compatibility, modifying trigger words, and altering IOCs. The experimental results prove that the performance on the augmented data set is better, and the TriCTI we proposed has higher accuracy (86.99%) than other state-of-the-art models, such as BERT. Thanks to the highly explanatory trigger vectors improve the attention weight of the keywords that can best represent the sentence campaign stages. Finally, our system discovered 113,543 actionable CTI from more than 29k cybersecurity reports from 2000 to 2021. Significantly, we prove the actionability of discovered CTI by using large-scale field data from VirusTotal (VT). We find that VT has limitations in describing attack behaviors using *Relationships* they offered. And our TriCTI system is able to perfectly cover all campaign stages regarding IOCs. Once a threat intrusion is detected, security professionals can quickly adjust the defense strategy according to the actionable CTI.

Acknowledgements
The authors would like to thank all the anonymous reviewers for their useful comments and suggestions.

Authors' contributions
JY designed the technical route, performed the experiments, and drafted the manuscript. YH assisted in collecting the data. JJ, ZJ, and JL made contributions to revise the manuscript. XW, PY, and NL participated in problem discussions and revised the article. All authors read and approved the final manuscript.

Funding
Our research was supported by the National Key Research and Development Program of China (Nos. 2019QY1301, 2018YFB0805005, 2018YFC0824801). This research was also partially supported by the Key Laboratory of Network Assessment Technology, the Chinese Academy of Sciences, and the Beijing Key Laboratory of Network Security and Protection Technology.

Availability of data and materials
Campaign stages trigger corpus can be found at https://github.com/lingren0/TriCTI.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China. [2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100029, China. [3]College of Information Engineering, Capital Normal University, Beijing 100048, China.

Received: 29 June 2021 Accepted: 11 January 2022
Published online: 02 April 2022

References

AlienVault: Open Threat Intelligence (2021) https://otx.alienvault.com/. Accessed 16 June 2021
Amazon: Alexa (2021) https://www.alexa.com/topsites/. Accessed 25 May 2021
Bouwman X, Griffioen H, Egbers J, Doerr C, Klievink B, van Eeten M (2020) A different cup of TI? the added value of commercial threat intelligence. In: 29th USENIX security symposium (USENIX security 20), pp 433–450
CleanMX (2021) CleanMX. https://support.clean-mx.com/clean-mx/index.php. Accessed 25 May 2021
De Silva R, Nabeel M, Elvitigala C, Khalil I, Yu T, Keppitiyagama C (2021) Compromised or attacker-owned: a large scale classification and study of hosting domains of malicious urls. In: 30th USENIX security symposium (USENIX security 21)
Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
Dionísio N, Alves F, Ferreira PM, Bessani A (2019) Cyberthreat detection from twitter using deep neural networks. In: 2019 international joint conference on neural networks (IJCNN), pp 1–8. IEEE
Dong Y, Guo W, Chen Y, Xing X, Zhang Y, Wang G (2019) Towards the detection of inconsistencies in public security vulnerability reports. In: 28th USENIX security symposium (USENIX Security 19), pp 869–885
Hadsell R, Chopra S, LeCun Y (2006) Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), vol 2, pp 1735–1742. IEEE

Husari G, Al-Shaer E, Ahmed M, Chu B, Niu X (2017) 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
Husari G, Niu X, Chu B, Al-Shaer E (2018) Using entropy and mutual information to extract threat actions from cyber threat intelligence. In: 2018 IEEE international conference on intelligence and security informatics (ISI), pp 1–6. IEEE
Hutchins EM, Cloppert MJ, Amin RM et al (2011) Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Leading Issues Inf Warfare Secur Res 1(1):80
Jeff M (2021) The security intelligence handbook. https://cyber-edge.com/resources/the-security-intelligence-handbook-third-edition/. Accessed 16 June 2021
Kim G, Lee C, Jo J, Lim H (2020) Automatic extraction of named entities of cyber threats using a deep BI-LSYM-CRF network. Int J Mach Learn Cybern 11(10):2341–2355
Kim D, Kim HK (2019) Automated dataset generation system for collaborative research of cyber threat analysis. Secur Commun Netw
Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
Le Pochat V, Maroofi S, Van Goethem T, Preuveneers D, Duda A, Joosen W, Korczyński M, et al (2020) A practical approach for taking down avalanche botnets under real-world constraints. In: Proceedings of the 27th annual network and distributed system security symposium. Internet Society
Lever C, Walls R, Nadji Y, Dagon D, McDaniel P, Antonakakis M (2016) Domain-z: 28 registrations later measuring the exploitation of residual trust in domains. In: 2016 IEEE symposium on security and privacy (SP), pp 691–706. IEEE
Li VG, Dunn M, Pearce P, McCoy D, Voelker GM, Savage S (2019) Reading the tea leaves: a comparative analysis of threat intelligence. In: 28th USENIX security symposium (USENIX Security 19), pp 851–867
Liao X, Yuan K, Wang X, Li Z, Xing L, Beyah R (2016) Acing the IOC game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 755–766
Lin BY, Lee D-H, Shen M, Moreno R, Huang X, Shiralkar P, Ren X (2020) Triggerner: Learning with entity triggers as explanations for named entity recognition. arXiv preprint arXiv:2004.07493
Lin Z, Feng M, Santos CNd, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130
Long Z, Tan L, Zhou S, He C, Liu X (2019) Collecting indicators of compromise from unstructured text of cybersecurity articles using neural-based sequence labelling. In: 2019 international joint conference on neural networks (IJCNN), pp 1–8. IEEE
MITRE: Common Attack Pattern Enumeration and Classification (CAPEC) (2021) https://capec.mitre.org/index.html. Accessed 25 May 2021
MITRE: Malware Attribute Enumeration and Characterization (MAEC) (2021) https://maecproject.github.io/. Accessed 25 May 2021
MITRE: MITRE ATT&CK (2021) https://attack.mitre.org/. Accessed 25 May 2021
OASIS: STIX (2021) https://oasis-open.github.io/cti-documentation/stix/intro.html. Accessed 25 May 2021
Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
Samtani S, Abate M, Benjamin V, Li W (2020) Cybersecurity as an industry: a cyber threat intelligence perspective. Palgrave Handb Int Cybercrime Cyberdev 135–154
Satyapanich T, Ferraro F, Finin T (2020) CASIE: extracting cybersecurity event information from text. UMBC Faculty Collection
Singh S, Sharma PK, Moon SY, Moon D, Park JH (2019) A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions. J Supercomput 75(8):4543–4574
Spacy V3.0 https://spacy.io/. Accessed 25 May 2021
Tang D, Qin B, Feng X, Liu T (2015) Effective lstms for target-dependent sentiment classification. arXiv preprint arXiv:1512.01100
Van Der Maaten L (2014) Accelerating T-SNE using tree-based algorithms. J Mach Learn Res 15(1):3221–3245
Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:5998–6008
VirusTotal (2021) https://developers.virustotal.com/v3.0. Accessed 25 May 2021
Wu X, Lv S, Zang L, Han J, Hu S (2019) Conditional bert contextual augmentation. In: International conference on computational science, pp 84–95. Springer
Yadav T, Rao AM (2015) Technical aspects of cyber kill chain. In: International symposium on security in computing and communication, pp 438–452. Springer
Zane P (2021) The threat intelligence handbook. https://cyber-edge.com/resources/the-threat-intelligence-handbook-second-edition/. Accessed 16 June 2021
Zhao J, Yan Q, Liu X, Li B, Zuo G (2020) Cyber threat intelligence modeling based on heterogeneous graph convolutional network. In: 23rd international symposium on research in attacks, intrusions and defenses (RAID 2020), pp 241–256
Zhou S, Long Z, Tan L, Guo H (2018) Automatic identification of indicators of compromise using neural-based sequence labelling. arXiv preprint arXiv:1810.10156
Zhu Z, Dumitras T (2018) Chainsmith: automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: 2018 IEEE European symposium on security and privacy (EuroS&P), pp 458–472. IEEE

Publisher's Note