Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 2933–2945
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.26981.3468

A novel framework for APT attack detection based on network traffic

Bui Van Cong[a,*], Nguyen Quoc Thanh[b], Nguyen Duy Phuong[c]

[a]Department of Information Technology, University of Economics and Technical Industries, Ha Noi, Vietnam

[b]Information Technology Department, LienVietPostBank, Ha Noi, Vietnam

[c]Department of Information Technology, Posts and Telecommunications Institute of Technology, Ha Noi, Vietnam

(Communicated by Mouquan Shen)

Abstract

APT (Advanced Persistent Threat) attack is a dangerous, targeted attack form with clear targets. APT attack campaigns have huge consequences. Therefore, the problem of researching and developing the APT attack detection solution is very urgent and necessary nowadays. On the other hand, no matter how advanced the APT attack, it has clear processes and lifecycles. Taking advantage of this point, security experts recommend that could develop APT attack detection solutions for each of their life cycles and processes. In APT attacks, hackers often use phishing techniques to perform attacks and steal data. If this attack and phishing phase is detected, the entire APT attack campaign will crash. Therefore, it is necessary to research and deploy technology and solutions that could detect early the APT attack when it is in the stages of attacking and stealing data. This paper proposes an APT attack detection framework based on the Network traffic analysis technique using open-source tools and deep learning models. This research focuses on analyzing Network traffic into different components, then finds ways to extract abnormal behaviors on those components, and finally uses deep learning algorithms to classify Network traffic based on the extracted abnormal behaviors. The abnormal behavior analysis process is presented in detail in section 3.1 of the paper. The APT attack detection method based on Network traffic is presented in section 3.2 of this paper. Finally, the experimental process of the proposal is performed in section 4 of the paper.

Keywords: APT, APT detection, network traffic, LSTM, abnormal behavior analysis
2020 MSC: 68M25

1 Introduction

1.1 The problem

Publications [3, 14] presented the characteristics, process, and life cycle of the APT attack. These characteristics show that the APT attack has specific and clear goals and targets. Any organization, individual, business, or government agency could become a victim of this attack.

*Corresponding author
Email addresses: bvcong@uneti.edu.vn (Bui Van Cong), thanhnq1@lienvietpostbank.com.vn (Nguyen Quoc Thanh), phuongnd@ptit.edu.vn (Nguyen Duy Phuong)

Received: April 2022 *Accepted:* July 2022

In the paper [3], the authors presented some characteristics of the APT attack scenario that make detecting this attack much more difficult than any other threat. One of the difficulties in detecting APT attacks is the lack of public data about this attack. Most victims of APT attacks rarely disclose their data or admit to being victims. However, although the APT attack is advanced and sophisticated with completely new attack methods, it could all be divided into main stages [3, 14, 4]: reconnaissance (information gathering); attack and privilege escalation; stealing information; remove traces.

Studies [3, 28, 26] presented some main approaches for APT attack detection. Accordingly, the approaches based on machine learning and deep learning are being studied and used increasingly in the task of classifying abnormal behaviors of APT. However, studies [28, 26] listed and reviewed some disadvantages of approaches based on abnormal behavior analysis. Besides, in studies [28, 26, 27, 25, 29, 30, 31, 33**?**], some approaches were proposed to address the disadvantages of approaches based on behavior analysis using machine learning. This paper proposes a novel approach based on an APT attack behavior analysis technique using Network traffic and deep learning. Specific characteristics of our approach are as follows:

• Step 1: Analyze network traffic using the Suricata tool. At this step, network traffic data is analyzed and evaluated by the Suricata tool to perform two tasks: i) detect APT attacks based on the ruleset; ii) analyze network traffic into different fields, layers, and components.

• Step 2: Extract abnormal behaviors of APT attacks based on statistical features of different components in network traffic analyzed in step 1. Specifically, this paper uses components: Domain Name System (DNS), HyperText Transfer Protocol (HTTP), Transport Layer Security (TLS), Event, Alert, etc.

• Step 3: Detect APT attacks based on network traffic components using the Long Short Term Memory (LSTM) deep learning model. Accordingly, based on the abnormal behaviors of Network traffic collected in step 2, in this step, the LSTM deep learning model is used to detect which behaviors are APT attack behaviors and which behaviors are normal.

1.2 Contributions of Paper

The practical and scientific significance of our paper includes:

• Proposing an APT attack detection model based on the Suricata open-source tool and the deep learning algorithm. The idea of this combination is based on the method of combining open-source network monitoring tools and deep learning algorithms. In particular, the open-source tool Suricata is one of the free tools but can detect and early warn unusual behavior of network attacks based on rule sets. This tool uses a very rich and diverse set of rules. In addition, in the process of analyzing data to look for signs and behavior of APT attacks, this tool also provides details of the components of the data after it has been analyzed. Therefore, the use of the Suricata tool in this paper will help the system to quickly and accurately detect the behavior of APT attacks, and also provide detailed information for the detection process of APT abnormal behavior using machine learning. Besides, for the deep learning algorithm, we choose to use the LSTM algorithm because this algorithm has been proven to be highly efficient in time datasets. Therefore, in this paper, we use a combination of the Suricata tool and LSTM algorithm.

• Proposing methods to analyze and extract features of network traffic based on the components collected in the Suricata log.

• Proposing to use the LSTM deep learning model for APT attack detection based on network traffic.

2 Related works

The publication [4] used three main feature groups (Domain name lexical features, Ranking features, DNS query features) and the Random Forest (RF) algorithm to detect APT domains. Besides, Yan et al. [32] proposed using the Convolutional Neural Network (CNN) deep learning algorithm to detect APT attacks based on DNS Activities. Accordingly, the authors extracted the main feature groups: Domain Name-based Features; Feature of the Relationship between DNS Request Behavior and Response Behavior from 4,907,147,146 piece dataset from DNS request records of Jilin University Education Network within 47 days. The authors [32] used these features with the CNN algorithm to detect APT attack behaviors. Xiang et al. [24] proposed a method to detect APT attacks on mobile devices based on analyzing DNS logs using machine learning algorithms. The authors argued that there is a big difference between the DNS of APT malware on mobile devices and computers. Therefore, the authors proposed an APT DNS detection process including i) checking the difference between mobile DNS and computer DNS; ii) selecting and extracting features: Total Number of Visits, Number of Accessing Hosts; Domain Length; Solitariness of Access; Repeated

Request; Time of Connection; Domain Structure; Access Regularity; Independent Access. In addition, there are some other approaches for malicious domain detection for supporting APT attack detection, including Vinayakumara et al. [20] used deep learning algorithms, and Nguyen [19] proposed using neutrosophic sets. In the study [23], Wen-Lin Chu et al. proposed an APT attack detection method based on the NSL-KDD dataset using the Support Vector Machine (SVM) algorithm. At the same time, in their research, the authors also used the principal component analysis algorithm to optimize the experimental dataset. In the study [3], Nkiruka Eke et al. proposed an APT attack detection method based on the KDD 99 dataset and deep learning algorithms such as LSTM, Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU). Experimental results showed that the deep learning algorithm yielded higher results than other machine learning algorithms such as SVM, K-nearest Neighbors (KNN), RF Classifier, and Logistic Regression (LR). Some other approaches for detecting anomalies in network traffic, which are used for cyber-attack detection in general and APT attack in particular, include Peng [15] et al. first proposed a network anomaly detection algorithm using Mahout Classifier; Huang [9] proposed to use a clustering algorithm to optimize the network anomaly detection process; Wang et al. [21] proposed using the CNN deep learning algorithm for detecting anomalies based on the NSL-KDD dataset. Ibrahim et al. [7] proposed an APT attack detection method based on a multi-layer analysis technique using Hidden Markov Models. Accordingly, in that study, the authors used Hidden Markov Models to analyze and evaluate the correlation between alerts, and take it as a basis to conclude about APT attacks. The experimental results in [7] showed that the accuracy of the detection model was at least 91.80%. Besides, the accuracy of predicting the next step of the APT campaign based on 2, 3, and 4 correlated alerts were 66.50%, 92.70%, and 100%, respectively. Zimbra [1] proposed a model for detecting the APT attack at multi-stages based on semi-supervised learning. This research used data from an enterprise network with 17,684 hosts from the Los Alamos security lab to rank suspicious hosts involved in APT attack campaigns. The average detection precision of the three APT stages was 90.5%. Lajevardi et al. [11] proposed an approach using low-level interception and correlation between operating system events and network events based on the semantic relationships defined between the entities in system ontology. In the publication [6], Ghafir et al. proposed the MAPT model for APT detection using machine learning algorithms. This model has three main stages: Threat detection, Alert correlation, and Attack prediction. In the experimental process, based on algorithms such as Decision Tree (DT), KNN, SVM, Ensemble, and the network traffic dataset collected in the university, the MAPT system had an accuracy of 84.8%. Another solution proposed by Alshamrani [2] for APT attack detection is based on the combination of multi-source data to learn abnormal behaviors of suspicious users as well as choosing optimally the appropriate countermeasures. In addition, studies [12, 22, 10] proposed models for detecting and tracing APT attacks based on the process of tracking and monitoring different components in the access log.

3 The proposed model architecture

3.1 The proposed model for detect signs of APT attacks

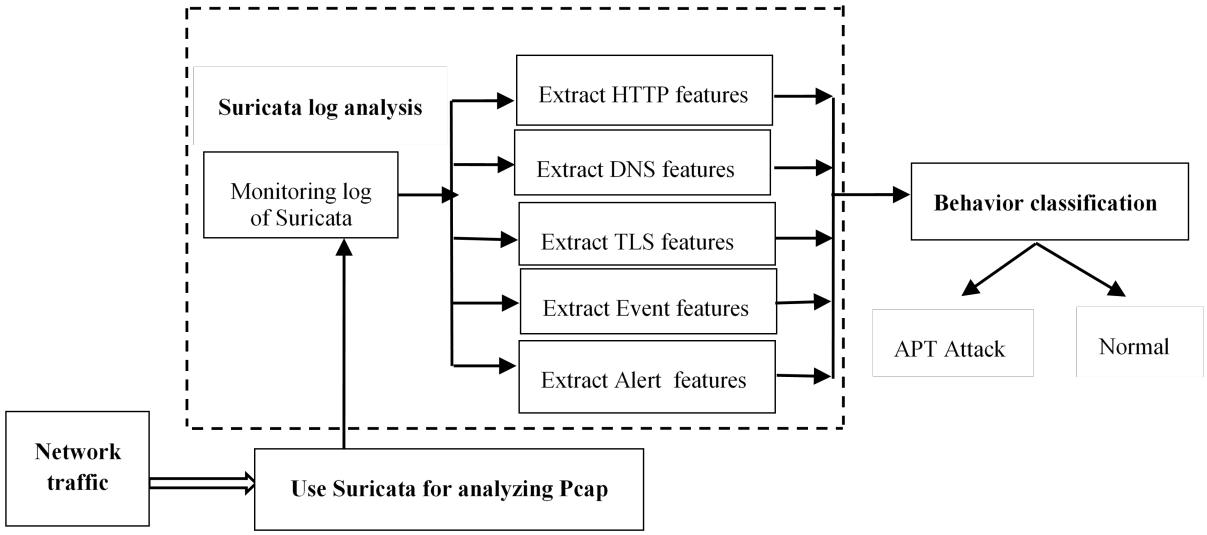

Figure 1: Main process flows

From Figure 1, it can be seen that the process of detecting signs of APT attacks in the APT attack sign detection module is as follows:

• The "Suricata log monitoring and analysis" block: This block has 2 basic functions: monitoring data and analyzing logs. In particular, the data monitoring function is responsible for detecting APT attacks based on the ruleset contributed by the Suricata community. In addition, the log analysis function is responsible for aggregating behaviors in data to build a behavior set of data for detecting APT attacks. The behavior groups of data collected by Suricata include HTTP, DNS, TLS, Event, Alert, etc.

• The "Suricata log analysis" block: Based on components in Network Traffic such as HTTP, DNS, TLS, Event, and Alert, this paper conducts research and extracts behaviors of Network Traffic based on these components. Previous research methods on APT attack detection based on Network Traffic all tried to find and extract typical behaviors of APT attack malware. However, these approaches often require the collected data to be large and to be gathered over a long period. This leads to difficulties in data storage and management. Therefore, this paper improves the old approaches by analyzing Network Traffic into components and then processing and extracting behaviors based on those components. With this approach, this study will use a combination of all behaviors of different events to conclude about the APT attack behaviors.

• APT attack detection: After fully collecting the behaviors of each event based on Network Traffic at the "Suricata log analysis" block, the APT attack detection system proceeds to classify each of these behavior profiles. The results of this classification process point out which behavior profiles are similar to the behavior profiles of known APT attack campaigns and which behavior profiles are not in APT attack campaigns.

Thus, it can be seen that our APT attack detection model is a combination of two detection methods: based on the ruleset using the Suricata tool, and using deep learning. Combining the two detection methods and dividing them into different phases makes our APT attack detection proposal the ability to detect and monitor in real-time.

3.2 Designing main flows in the APT attack sign detection module

3.2.1 The flow of detecting and predicting signs of APT attacks based on the Suricata tool

Figure 2 below depicts the APT attack detection process based on the Suricata tool.

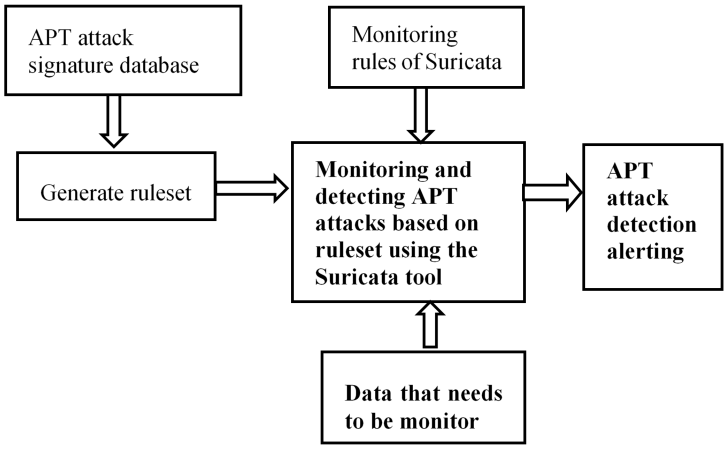

Figure 2: The architecture of APT attack detection model based on Suricata tool and rule set

Suricata tool is one of the powerful tools for supporting the process of detecting and monitoring cyber-attacks in general and APT attacks in particular [23]. In this paper, the research team combines the Suricata tool with the previously provided APT attack signature database as the basis for detecting APT attacks. At this stage, if an attack sign is detected, the alarm will be sent directly to the warning system. This can be considered a 100% accurate sign of the existence of the APT attack campaign in the system. The Suricata tool analyzed the network traffic to compare it with the given APT attack signs and store the network traffic information in its log. Suricata's log contains a lot of important information for the process of gathering and monitoring APT attacks. Therefore, this paper continues to apply the Suricata log analysis technique to look for abnormal behavior signs of APT attacks and takes them as a basis for concluding the existence of APT attacks.

3.2.2 The flow of detecting and predicting APT attacks based on the deep learning

Based on the analysis in Figure 1, it can be seen that the deep learning model for APT attack detection based on network traffic is as follows:

a) Analyzing the Suricata log: As described above, all network traffic is put into the Suricata tool to detect signs of APT attacks. At the same time, all this initial Pcap data is processed by the Suricata tool and saved in the Suricata log.

Figure 3: Some basic information in the Suricata log

As shown in Figure 3, it can be seen that the Suricata log includes some components: DNS log obtained according to Suricata standard; HTTP log; TLS (transport layer security) log; Event log. These can be considered as the basic components of the Pcap obtained by the Suricata tool. Therefore, to accurately detect APT attacks, the APT attack sign detection module has to detect signs of APT attacks on each of those components. To accomplish this task, the APT attack sign detection module needs to extract features of all the above components. After obtaining a list of features of all the above components, these features are built into behavior profiles, and then analyzed these behavior profiles to conclude signs of APT attacks in the system. Next, the paper will present the behavior features of some components of the Suricata log.

• The list of Alert features: The alert log in the Suricata log presents alerts detected by the Suricata tool. The Alert features represent anomalies in the contents of the Pcap file. Table 1 below shows the list of extracted Alert features.

Table 1: List of Alert features in the Suricata log

Feature	Description	Data type
total_alert	The number of alert logs in conversation	Long
alert_gpcd	The number of alerts belonging to the Generic Protocol Command Decode group	Long
alert_mics	The number of alerts belonging to the Misc Activity group	Long
alert_Network _Trojan	The number of alerts belonging to group A Network Trojan was detected	Long
alert_Potentially _Bad_Traffic	The number of alerts belonging to the Potentially Bad Traffic group	Long
alert_Potential _Corporate_Privacy _Violation	The number of alerts belonging to the Potential Corporate Privacy Violation group	Long
alert_others	The number of alerts belonging to other groups	Long

• List of DNS features from the Suricata log. The DNS log obtained from the Suricata log is the DNS query information that Suricata obtained in Pcap. Figure 4 below shows the contents of the DNS log obtained by Suricata.

From the information of the DNS log in the Suricata log, the research team proceeds to extract important features of DNS. Table 2 below shows the list of DNS features that the research team extracted from the DNS log in the Suricata log. These are abnormal features of DNS queries and some statistical features.

• List of features from Event. Table 3 below describes the Event features extracted from the Suricata log.

• List of HTTP features from the Suricata log. HTTP log from the Suricata log is information about access over HTTP protocol in the network. Figure 5 below shows the contents of the HTTP connection in the Suricata log.

Through the information from the HTTP log (see Figure 5), the administrator could determine the information

dns-json.json

1 {"timestamp":"2013-02-04T02:51:03.940614+0000","flow_id":78735912426054,"pcap_cnt":24,"event_type":"dns",
"src_ip":"172.16.253.130","src_port":53,"dest_ip":"8.8.8.8","dest_port":53,"proto":"UDP","dns":
{"type":"query","id":34738,"rrname":"godson355.vicp.cc","rrtype":"A","tx_id":0}}
2 {"timestamp":"2013-02-04T02:51:03.940765+0000","flow_id":957396026874589,"pcap_cnt":25,"event_type":"dns",
"src_ip":"172.16.253.130","src_port":53,"dest_ip":"4.2.2.2","dest_port":53,"proto":"UDP","dns":
{"type":"query","id":34738,"rrname":"godson355.vicp.cc","rrtype":"A","tx_id":0}}
3 {"timestamp":"2013-02-04T02:51:04.317046+0000","flow_id":78735912426054,"pcap_cnt":26,"event_type":"dns",
"src_ip":"8.8.8.8","src_port":53,"dest_ip":"172.16.253.130","dest_port":53,"proto":"UDP","dns":
{"type":"answer","id":34738,"flags":"8180","qr":true,"rd":true,"ra":true,"rcode":"NOERROR",
"rrname":"godson355.vicp.cc","rrtype":"A","ttl":3600,"rdata":"202.85.136.181"}}
4 {"timestamp":"2013-02-04T02:51:04.504475+0000","flow_id":957396026874589,"pcap_cnt":28,"event_type":"dns",
"src_ip":"4.2.2.2","src_port":53,"dest_ip":"172.16.253.130","dest_port":53,"proto":"UDP","dns":
{"type":"answer","id":34738,"flags":"8180","qr":true,"rd":true,"ra":true,"rcode":"NOERROR",
"rrname":"godson355.vicp.cc","rrtype":"A","ttl":3600,"rdata":"202.85.136.181"}}
5 {"timestamp":"2012-10-06T06:12:43.100578+0000","flow_id":957396026874589,"pcap_cnt":66,"event_type":"dns",
"src_ip":"172.16.253.130","src_port":53,"dest_ip":"4.2.2.2","dest_port":53,"proto":"UDP","dns":
{"type":"query","id":4149,"rrname":"www.microsoft.com","rrtype":"A","tx_id":1}}
6 {"timestamp":"2012-10-06T06:12:43.122241+0000","flow_id":957396026874589,"pcap_cnt":67,"event_type":"dns",
"src_ip":"4.2.2.2","src_port":53,"dest_ip":"172.16.253.130","dest_port":53,"proto":"UDP","dns":
{"type":"answer","id":4149,"flags":"8180","qr":true,"rd":true,"ra":true,"rcode":"NOERROR",
"rrname":"www.microsoft.com","rrtype":"CNAME","ttl":3562,"rdata":"toggle.www.ms.akadns.net"}}
7 {"timestamp":"2012-10-06T06:12:43.122241+0000","flow_id":957396026874589,"pcap_cnt":67,"event_type":"dns",
"src_ip":"4.2.2.2","src_port":53,"dest_ip":"172.16.253.130","dest_port":53,"proto":"UDP","dns":
{"type":"answer","id":4149,"flags":"8180","qr":true,"rd":true,"ra":true,"rcode":"NOERROR",
"rrname":"toggle.www.ms.akadns.net","rrtype":"CNAME","ttl":70,"rdata":"g.www.ms.akadns.net"}}
8 {"timestamp":"2012-10-06T06:12:43.122241+0000","flow_id":957396026874589,"pcap_cnt":67,"event_type":"dns",
"src_ip":"4.2.2.2","src_port":53,"dest_ip":"172.16.253.130","dest_port":53,"proto":"UDP","dns":
{"type":"answer","id":4149,"flags":"8180","qr":true,"rd":true,"ra":true,"rcode":"NOERROR",
"rrname":"g.www.ms.akadns.net","rrtype":"CNAME","ttl":70,"rdata":"lb1.www.ms.akadns.net"}}
9 {"timestamp":"2012-10-06T06:12:43.122241+0000","flow_id":957396026874589,"pcap_cnt":67,"event_type":"dns",
"src_ip":"4.2.2.2","src_port":53,"dest_ip":"172.16.253.130","dest_port":53,"proto":"UDP","dns":
{"type":"answer","id":4149,"flags":"8180","qr":true,"rd":true,"ra":true,"rcode":"NOERROR",
"rrname":"lb1.www.ms.akadns.net","rrtype":"A","ttl":194,"rdata":"65.55.57.27"}}
10 {"timestamp":"2012-10-06T06:13:44.143452+0000","flow_id":78735912426054,"pcap_cnt":90,"event_type":"dns",
"src_ip":"172.16.253.130","src_port":53,"dest_ip":"8.8.8.8","dest_port":53,"proto":"UDP","dns":
{"type":"query","id":4951,"rrname":"www.vipyandex.com","rrtype":"A","tx_id":1}}
11 {"timestamp":"1970-01-01T00:00:13.683451+0000","flow_id":739616991571387,"pcap_cnt":214,"event_type":"dns",

Figure 4: Some content stored in DNS in the Suricata log

http-json.json

1 {"timestamp":"2012-10-06T06:13:45.556943+0000","flow_id":1024709606600382,"pcap_cnt":132,
"event_type":"http","src_ip":"172.16.253.130","src_port":1053,"dest_ip":"110.34.193.13","dest_port":80,
"proto":"TCP","tx_id":0,"http":{"hostname":"www.vipyandex.com","url":"\/nt2011\/zy\/nettraveler.asp?
hostid=E81B9088&hostname=DellXT&hostip=172.16.253.130&filename=FileList-1006-233757.ini&filestart=0&
filetext=begin::OgA1AC2QzebTgdToZTkXQaCicYTaZR6RDKbDYWCpKKBhM88YjIajKXLfKOEmQ0nIxm86m46D0YVg::end",
"http_user_agent":"Mozilla\/4.0 (compatible; MSIE 6.0; Windows NT 5.0)","http_content_type":"text\/html"
}

Figure 5: The content of HTTP log in the Suricata log

related to communicative behaviors with the C&C server or file download behavior, etc. From the information that the HTTP log provides, the research team extracts its important behaviors as shown in Table 4. These are abnormal features of the APT attack exposed over the HTTP protocol.

- List of TLS features from the Suricata log. TLS log stores TLS and SSL exchange information of HTTPS connections. Figure 6 below shows the information that Suricata obtains from Pcap about TLS and SSL certificates [8].

From the content stored in the Suricata log, this study evaluates and extracts the features of TLS. These features help find abnormal HTTPS connections that are not reliable. Table 5 below lists and describes some features of TLS that the research team has built and extracted.

Table 2: List of DNS features in the Suricata log

Feature	Description	Data type
Domain name length	Domain length	Integer
Domain name token count	Number of tokens separated from the domain name by the character "."	Integer
Average domain token length	The average length of tokens	Double
Standard divination	Standard divination	Double
Number of special characters	Number of special characters in the domain name	Integer
Number of digits	Number of numeric characters in the domain name	Integer
Number of continuous digits	Number of continuous numeric characters in the domain name	Integer
Longest continuous letters length	Maximum length of continuous letters in the domain name	Integer
Average rank number	The average rank	Integer
Resolved IP count	Number of IP addresses returned in the DNS query	Integer
Distinct country IP count	Number of countries from IP addresses	Integer
Number of private IP	Private IP number	Integer
HTTP Response Status	Response status code	Integer
Name server count	Number of name servers returned in the DNS query	Integer
Mail server count	Number of mail exchange servers returned in the DNS query	Integer
Average TTL	Average TTL (Time to live) of cache records for the domain name at the name server	Integer

Table 3: List of Event features in the Suricata log

Feature	Description	Data type
src_ip	Source IP address	String
dest_ip	Destination IP address	String
start_time	Timestamp of the first log in conversation	Long
end_time	Timestamp of the last log in conversation	Long
duration	The time gap between the first log and the last log in conversation (= end_time - start_time)	Long
domain_label	Suspicious domain name (1: malicious, 2: normal)	Long
label	Label	

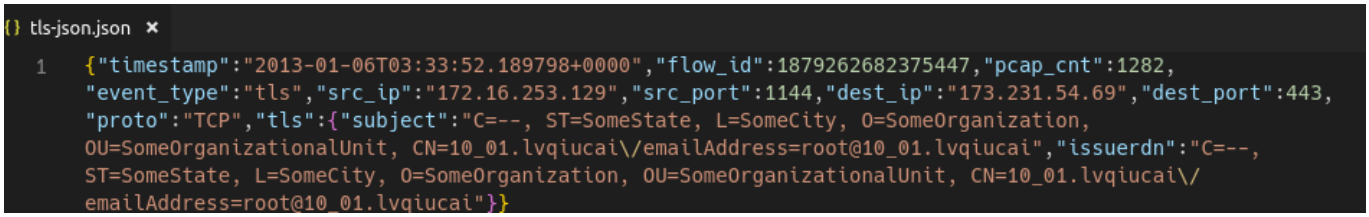

Figure 6: The content of TLS in the Suricata log

b) APT attack detection based on deep learning technique: Based on the behaviors collected from the Suricata log analysis process, the model uses the deep learning algorithm to accurately conclude the existence of APT attack signs in the system. To accomplish this task, this study proposes to use the LSTM deep learning model.

Table 4: List of HTTP features in the Suricata log

Feature	Description	Data type
http_request_count	Number of HTTP requests in conversation	Long
http_protocol_ mismatch_ratio	The ratio of requests using the old version 1.0 to the total number of requests	Double
http_port_mismatch _ratio	The ratio of requests that do not use standard ports (80 for HTTP and 443 for HTTPS)	Double
http_failed_request _count	Number of HTTP logs with response status 4xx or 5xx	Long
http_length_ {min, max, avg, std}	Min, max, mean and standard deviation of HTTP length	Long, Long, Double, Double
http_uri_distince_count	Number of distinguished URI in conversation	Long
http_request_frequency	Frequency of requests in a second	Double

Table 5: List of TLS features in the Suricata log

Feature	Description	Data type
Self-signed TLS	Self-signed TLS certificate	Boolean
Number of TLS heartbleed malformed record	Number of TLS heartbleed malformed alerts	Integer
Number TLS handshake a day (min, max, avg)	Number of TLS handshakes per day	Integer, Integer, Float
Number fail TLS handshake a day (min, max, avg)	Number of fail TLS handshakes per day	Integer, Integer, Float

In the study [17], Hochreiter and Schmidhuber introduced the architecture and mathematical foundations of the LSTM network. The LSTM network is a neural network developed on the structure of RNN [13] to overcome some problems related to Gradient Exploding and Gradient Vanishing when the network is too long. The LSTM network can remember information from the previous state of the network so that it could process series data. Figure 7 illustrates the structure of a basic memory cell in the LSTM network with 4 gates having different tasks.

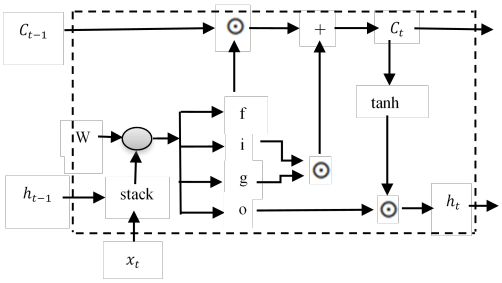

Figure 7: The architecture of a hidden cell of LSTM deep learning network

The gates are used to control how much information from the previous cell could be add or erase. At each time t, we have a hidden state h_t and a cell state c_t with the basic mathematical formulas shown below:

The input gate to control how much data to write:

$$i_t = \sigma(W^{(i)}_{h_{t-1}} + U^{(i)}_{x_t} + b^{(i)}). \tag{3.1}$$

The forget gate to control how much data will be erased:

$$f_t = \sigma(W_{h_{t-1}}^{(f)} + U_{x_t}^{(f)} + b^{(f)}). \tag{3.2}$$

The output gate to control how much data will go through:

$$o_t = \sigma(W_{h_{t-1}}^{(o)} + U_{x_t}^{(o)} + b^{(o)}). \tag{3.3}$$

And the new memory cell to control what will be write:

$$\hat{c}_t = tanh(W_{h_{t-1}}^{(c)} + U_{x_t}^{(c)} + b^{(c)}). \tag{3.4}$$

And two cell:

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t \tag{3.5}$$

$$h_t = o_t \odot c_t \tag{3.6}$$

where: W is the weight matrix of each gate corresponding to the hidden state of the previous cell; U is the weight matrix of each gate corresponding to the input at time t; $\odot$ is the element-wise product operator.

Thus, with the input of the list of features collected and calculated above, the output of this process is the correlation ratio between the behavior profiles of computers in the system with the behavior profiles of computers in APT attack campaigns.

4 Experiments and evaluation

4.1 Experimental dataset

4.1.1 APT attack data

Experimental data were collected and analyzed from 29 network traffic files in the Malware Capture CTU-13 dataset. It consists of 6 malware types from APT attacks including Andromeda, Colbalt, Cridex, Dridex, Emotet, and Gh0stRAT [16].

4.1.2 Normal data

To create a balance between the APT malware dataset and the normal dataset, this study collects clean data from servers (the servers of the Mocha system and recommend) and personal computers (normal access to social networking sites such as Facebook, Youtube, or other normal websites such as Google, StackOverflow, etc.). These data are collected within 2 weeks.

4.1.3 Data synthesis

Total conversation: 193,212. In which: the number of normal conversations is 34,003; the number of malicious conversations is 159,209. The conversation is a collection of Suricata logs (including HTTP, Alert, Event, DNS, etc.) that share the same source and destination IP address. At the same time, it has a "timeout" so that if in a period N there is no request or response, the conversation will be ended.

4.2 Experimental scenarios

The training dataset accounts for 80% of the experimental dataset. The test dataset accounts for 20% of the experimental dataset.

To evaluate the effectiveness of the proposed model, this study conducts the following scenarios:

- Scenario 1. Detect APT malware using the LSTM model proposed by us.
- Scenario 2. Compare with some other approaches. For this scenario, this paper compares the proposed method with other approaches such as RF [8], SVM [23], and Multi-layer Perceptron (MLP) [23] algorithms.

4.3 Classification Measures

• Accuracy: the ratio between the number of correctly predicted points and the total number of points in the test dataset

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.1}$$

• Precision: The ratio between the true positive value and total number of samples classified as positive. The higher value of precision, the more accurate in APT malware detection.

$$Precision = \frac{TP}{TP + FP}. \tag{4.2}$$

• Recall: The ratio between the true positive value and the total real APT malware. The higher value of recall, the lower rate of missing positive samples.

$$Recall = \frac{TP}{TP + FN}. \tag{4.3}$$

• F1-score: The harmonic mean of precision and recall. The higher F1 score, the better the model is.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{4.4}$$

• Where: TP - True positive: The number of APT malware classified correctly; FN - False negative: The number of APT malware classified as normal; TN - True negative: The number of normal conversations classified correctly; FP - False positive: The number of normal conversations classified as APT malware.

4.4 Experimental results

4.4.1 Experimental results of scenario 1

Table 6: Experimental results of detecting APT attacks using LSTM Model

LSTM Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	TPR (%)	FPR (%)
1 layer	94.38	81.58	73.81	77.50	73.81	2.53
2 layer	**96.87**	**86.36**	**90.48**	**88.37**	**90.48**	**2.17**
3 layer	95.21	84.34	86.93	85.84	86.93	2.42

Table 6 shows the experimental results of the LSTM model for the task of detecting APT attacks based on network traffic. The experimental results show that when changing the number of layers of the LSTM model, the classification results were also different. The model gave the best results when the number of hidden layers of the LSTM is 2. Specifically, in the LSTM model using only 1 layer, the classification results were relatively low, in which Accuracy only reached 94.38%, the result of correctly classifying APT malware was only 73.81%, and the result of correctly classifying normal files was only 81.58%. When increasing the number of layers of LSTM to 2 layers, Accuracy was 96.87% (increased by 2.5% compared to the 1-layer LSTM model and 1.8% compared to the 3-layers LSTM model). Similarly, with the Recall measure, the 2-layers LSTM model reached 90.48%, about 16.67%, and 3.55% higher than the other models. In addition, for other measures, the 2-layers LSTM model also gave completely higher results. From the experimental results in Table 6, seeing that the LSTM model has worked effectively and has highlighted the important features of the data to make the classification system highly efficient. However, increasing the number of hidden layers in the LSTM model does not always increase efficiency. Specifically for the 3-layers LSTM model, the experimental results show that this model was not as effective as the 2-layers LSTM model.

4.4.2 Experimental results of scenario 2

The experimental results in Table 7 show that other algorithms such as RF, SVM, or MLP had relatively low efficiency in classifying APT attacks. The reason is that the extracted features in the dataset are all statistical features, so the difference between the malicious data and the clean data is not clearly shown. Therefore, the classification

Table 7: Experimental results of detecting APT attacks using some other algorithms

Algorithm (best param)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	TPR (%)	FPR (%)
RF (50 trees) [24]	**95.23**	**90.91**	**71.43**	**80.00**	**71.43**	**01.08**
SVM [11]	91.85	75.00	57.14	64.85	57.14	02.89
MLP [11]	93.21	79.34	61.28	70.08	61.28	02.32

model faces many difficulties and is prone to mispredictions. Comparing the results in Tables 6 and 7, it can be seen that the LSTM model that is proposed to use in this study brought much higher efficiency than other approaches [11, 24]. This shows that the proposal of using the LSTM model in this paper is not only scientific significance but also practical significance.

5 Conclusion and feature development direction

This study has succeeded in building an APT attack detection system based on the open-source tool and the deep learning algorithm. Our proposed model has not only the ability to quickly and accurately detect signs of APT attacks in network traffic based on the Suricata tool but also the ability to detect abnormal behaviors of this attack type based on the LSTM deep learning model. By proposing the method of calculating and extracting new features from network traffic, this study has succeeded in synthesizing and re-presenting the information of network traffic as a basis for conclusions about APT attacks in the system. The experimental results in the paper have proved the superiority of the proposed model compared with other approaches. This result has not only proved this proposal to be correct and reasonable, but also opened a new approach for detecting other attack methods. In the future, to improve the efficiency of the detection system, the team will continue to find ways to calculate the correlation between features to extract the important features to improve the ability to classify the APT attack.

References

[1] Z. Aaron, C.H. Song, W. Zhaoshun and C. Mumbi, *Modeling and detection of the multi-stages of Advanced Persistent Threats attacks based on semi-supervised learning and complex networks characteristics*, Future Gen. Comput. Syst. **106** (2020), 501–517.

[2] A. Alshamrani, A. Chowdhary, O. Mjihil, S. Myneni and D. Huang, *Combining dynamic and static attack information for attack tracing and event correlation*, 2018 IEEE Glob. Commun. Conf. (GLOBECOM), 2018, pp. 1–7.

[3] A. Alshamrani, A. Chowdhary, S. Myneni and D. Huang, *A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities*, IEEE Commun. Surv. Tutor. **1** (2019), 1–29.

[4] D.X. Cho and H.H. Nam, *Method of monitoring and detecting APT attacks based on unknown domains*, Procedia Comput. Sci. **150** (2019), 316–323.

[5] X.C. Do, D. Duc and D.H. Xuan, *A multi-layer approach for advanced persistent threat detection using machine learning based on network traffic*, J. Intell.Fuzzy Syst. **40** (2021), no. 6, 11311–11329.

[6] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie and F.J. Aparicio-Navarro, *Detection of advanced persistent threat using machine-learning correlation analysis*, Future Gen. Comput. Syst. **89** (2018), 349–359.

[7] I. Ghafir, K.G. Kyriakopoulos, S. Lambotharan, F.J. Aparicio-Navarro, B. AsSadhan, H. Binsalleeh, D.M. Diab, *Hidden Markov models and alert correlations for the prediction of advanced persistent threats*, IEEE Access **7** (2019), 99508–99520.

[8] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Comput. **9** (1997), no. 8, 1735—1780.

[9] H. Huang, H. Deng, Y. Sheng and X. Ye, *Accelerating convolutional neural network-based malware traffic detection through ant-colony clustering*, J. Intell. Fuzzy Syst. **37** (2019), 409–423.

[10] Y. Ji, S. Lee, E. Downing, W. Wang, M. Fazzini, T. Kim, A. Orso and W. Lee, *Rain: Refinable attack investigation with on-demand inter-process information flow tracking*, ACM SIGSAC Conf. Comput. Commun. Security, 2017, pp.377–390.

[11] A. Lajevardi and M. Amini, *A semantic-based correlation approach for detecting hybrid and low-level APTs*, Future Gen. Comput. Syst. **96** (2019), 64–88.

[12] S. Ma, J. Zhai, F. Wang, K.H. Lee, X. Zhang and D. Xu, *MPI: Multiple perspective attack investigation with semantic aware execution partitioning*, 26th USENIX Conf. Security Symp., 2017, pp. 1111-–1128.

[13] *Malware Capture Facility Project*, Available online: https://www.stratosphereips.org/datasets-malware. (Accessed on 8 June 2021).

[14] M. Marchetti, F. Pierazzi, M. Colajanni and A. Guido, *Analysis of high volumes of network traffic for Advanced Persistent Threat detection*, Comput. Networks **109** (2016), 127–141.

[15] H. Peng, L. Liu, J. Liu and J.R. Lewis, *Network traffic anomaly detection algorithm using mahout classifier*, J. Intell. Fuzzy Syst. **37** (2019), 137–144.

[16] M. Shen, P. Ju and F. Shumin, *Event-triggered nonfragile $H_\infty H_\infty$ filtering of Markov jump systems with imperfect transmisions*, Signal Process. **149** (2018).

[17] A. Sherstinsky, *Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network*, Phys. D: Nonlinear Phenomena **404** (2020).

[18] *Suricata*, Available online: https://suricata-ids.org/. (Accessed Feb 14, 2020).

[19] N. Van Can, D.N. Tu, T.A. Tuan, H.V. Long, L.H. Son and N.T.K. Son, *A new method to classify malicious domain name using neutrosophic sets in DGA botnet detection*, J. Intell. Fuzzy Syst. **36** (2020), 4223–4236.

[20] R. Vinayakumara, K.P. Somana and P. Poornachandranb, *Detecting malicious domain names using deep learning approaches at scale*, J. Intell. Fuzzy Syst. **34** (2018), 1355–1367.

[21] H. Wang, Z. Cao and B Hong, *A network intrusion detection system based on convolutional neural network*, J. Intell. Fuzzy Syst. **38** (2020), 7623–7637.

[22] F. Wang, Y. Kwon, S. Ma and X. Zhang, *Lprov: Practical library-aware provenance tracing*, 34th Ann. Comput. Security Appl. Conf., 2018, pp.605–617.

[23] L.C. Wen, J.L. Chih and N.C. Ke, *Detection and classification of advanced persistent threats and attacks using the support vector machine*, Appl. Sci. **9** (2019), 45–79.

[24] Z. Xiang, D. Guo and Q. Li, *Detecting mobile advanced persistent threats based on large-scale DNS logs*, Comput. Secur. **96** (2020).

[25] W. Xianming, Q. Wen, P. Ju and Mo. Shen, *Event-triggered data-driven control of discrete-time nonlinear systems with unknown disturbance*, ISA Transactions (2021) doi:10.1016/j.isatra.2021.11.026.

[26] C.D. Xuan, *Detecting APT attacks based on network traffic using machine learning*, J. Web Engin. **20** (2021), no. 1, 171–190.

[27] C.D. Xuan and H.M. Dao, *A novel approach for APT attack detection based on combined deep learning model*, Neural Comput. Appl. 33 (2021), no. 20, 13251–13264.

[28] C.D. Xuan, H.D. Nguyen and H.M. Dao, *APT attack detection based on flow network analysis techniques using deep learning*, J. Intell. Fuzzy Syst. **290** (2020), no. 3, 4785–4801.

[29] S. Yan, Z. Gu, J. H. Park, X. Xie and C. Dou, *Probability-density-dependent load frequency control of power systems with random delays and cyber-attacks via circuital implementation*, IEEE Trans. Smart Grid doi:10.1109/TSG.2022.3178976.

[30] S. Yan, Z. Gu and J. H. Park, *Memory-event-triggered H_∞ load frequency control of multi-area power systems with cyber-attacks and communication delays*, IEEE Trans. Network Sci. Engin. **8** (2021), no. 2, 1571–1583.

[31] S. Yan, Z. Gu, S.K. Nguang, F. Yang and L. Zhang, *Co-design of event-triggered scheme and H_∞ output control for Markov jump systems against deception attacks*, IEEE Access **8** (2020), 106554–106563.

[32] G. Yan, Q. Li, D. Guo and X. Meng, *Discovering suspicious APT behaviors by analyzing DNS activities*, Sensors **20** (2020), 1–17.

[33] S. Yan, S.K. Nguang and L. Zhang, *Nonfragile integral-based event-triggered control of uncertain cyber-physical systems under cyber-attacks*, Complexity **2019** (2019).