ISSN 2347 - 3983
Volume 8. No. 7, July 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter42872020.pdf
https://doi.org/10.30534/ijeter/2020/42872020

Detecting APT attacks based on Network Flow

Lai Van Duong[1], Tisenko Victor Nikolaevich[2],Do Hoang Long[3], Nguyen Quang Dam[4], Nguyen Quoc Hoang[5]
[1,3,4,5]Information Assurance dept. FPT University, Hanoi, Vietnam, duonglvse05009@fpt.edu.vn, longdhse05220@fpt.edu.vn, damnqse05820@fpt.edu.vn, hoangnqse06012@fpt.edu.vn
[2]Department Quality Systems, Peter the Great St. Petersburg Polytechnic University, Russia, St.Petersburg, Polytechnicheskaya, 29, v_tisenko@mail.ru

ABSTRACT

APT attack technique is an advanced and targeted attack technique. This attack technique has been recognized as one of the cyber attacks trend in the world for many years. Therefore, the task of the early detection of APT attacks in information security systems is very necessary today. In this paper, we propose a method of detecting APT attacks based on network flow analysis. Accordingly, network traffic data will be processed into flow networks. Then we use machine learning algorithms to analyze network flow in order to find signs of APT attacks in the system.

Key words: APT, APT detection, abnormal behavior, machine learning, network flow, network traffic.

1. INTRODUCTION

The studies [1, 2] presented an overview of the APT attack. Accordingly, the APT attack is a form of concentrated and targeted attack. It is designed specifically for each specific target and object in order to seek valuable information and send it out [1, 2, 3].

- The term "Advanced" is expressed in the ability to hide and constantly change, making detection of APT attacks very difficult and impossible for conventional detection systems. The term "Persistent" is expressed in identifying the specific target to make the attacks, hiding, and exploitation in each stage. Besides, "Persistent" is also shown in the use of many different techniques and methods to attack on the target until success. The term "Threat" means that APT attacks conducted by coordinated, targeted operation and highly skilled, organized, and well-funded attackers. The studies [1, 2] presented the characteristics and life cycle of the APT attack. Accordingly, APT attacks include the characteristics and processes as follows:
- Targeted: Attackers accurately identify a specific target to attack and exploit. Their target is usually specific organizations, individuals, countries, states, etc. Attackers also determine the specific purpose of an attack. It can be stealing data, doing the pedal to attack the network, breaking havoc, etc. This is in contrast with malware because malware randomly destroys in any infected system.
- Persistent: In order to steal data, attackers must identify vulnerability and evaluate the target's existing security system. After knowing all that, they invade the target, then access the target's privileged server. From there, they must find the target data and finally extract that data from the network. The whole process can take months or even years. In order to achieve the goals, attackers must perform many different steps and attack on multiple components in the system.
- Evasive: the APT attack is designed to evade most traditional security solutions such as Firewall, IPS, Antivirus, etc. In order to gain access to the target server and avoid being detected by network firewalls, attackers often use valid ports and services. Meanwhile, the security device will not be able to detect threats to the system, so it can't prevent the attacks. Attackers often design unique malicious code for each target and have never been detected by anti-virus software because those malicious codes aren’t included in the sample of anti-virus software. When sending data out, in order to avoid being detected by IDS, attackers perform the encryption of data. At that time, the victim's data is stolen without the victim knowing.
- Complex: The big difference between APT and other attacks is that APT combines different techniques scientifically and methodically to attack on targets and security vulnerabilities in organizations. APT can use a wide range of malware tools from simple to complex. Malwares was created to exploit zero-day vulnerabilities. Some techniques commonly used in APT attack techniques include Malware, Social Engineering, Exploiting Zero-day vulnerabilities, and some other ways.

The above characteristics and process of APT attack demonstrate that APT attack technique is very difficult to detect. The studies [1, 4, 5, 6, 7] presented and evaluated three current main approaches for detecting APT

attacks: sign-based, based on behavior analysis, and based on graph analysis. Accordingly, nowadays APT attack detection technique based on behavior analysis is being widely applied because of its effectiveness in detecting APT attacks.

In this paper, we propose a method to detect APT attacks based on analyzing the behavior of network traffic. The difference between our approach with the studies [1], [7] is that instead of directly extracting features that represent the behavior of APT attacks in network traffic, we will analyze network traffic into flow networks and then use Random Forest supervised learning algorithm to classify flow into APT attacks flow and normal flow. With this approach, we will not take too much time to research and extract abnormal features of APT on network traffic while ensuring the effectiveness of the detection method

2. RELATED WORKS

The APT attack detection technique based on abnormal behavior analysis requires two factors [1]: 1) The data must be large enough because the more data the higher the accuracy of the detection process. Currently, solutions often apply big data technology to store and extract information. 2) Processing algorithm. These are algorithms to cluster data to detect signs and behavior of an attack. In order to meet the demands of APT attack detection, the algorithms must often ensure clustering and processing in big data. Some theoretical studies on behavior analysis techniques for detecting APT attacks are as follows:

In the paper [8], the authors detected APT attacks based on two factors: DNS log and Network traffic. For the APT attack detection technique based on the DNS log, the authors used 5 feature groups: Domain-based features, Time-based features, Whois-based features, DNS answer-based features; and Active Probing features. These five feature groups have a total of 14 features for detecting malicious DNS. The clustering algorithm used in the paper is the J48 Decision Tree algorithm. For the APT attack detection technique based on network traffic, the authors presented 6 main features. After detecting an APT attack on both DNS log and Network traffic, the authors used a correlation analysis technique to detect which computer addresses in the system were infected with APT malware. However, in the paper [8], the authors didn't present details of this correlation calculation method.

In the study [9], the authors combined the J48 Decision Tree algorithm with 4 main feature groups: DNS request and answer-based features; Domain-based features; Time-based features; and Who is-based features to detect APT malware command and control domains (C&C Domain). The Global Abnormal Forest and KNN machine learning algorithms are used in this study. The statistical correlation analysis technique is used by the authors to find out some of the new abnormal features of APT attacks. However, the authors didn't present data sources from which these abnormal features would be extracted.

In the article [10], the author used the correlation analysis technique between DNS log and Network traffic, and some machine learning algorithms such as KNN, SVM to detect APT attacks. In the study [11], the authors used domain behavior analysis techniques to detect connections to the control servers of the APT attack campaign. In this study, the authors used the Random Forest algorithm and 4 main feature groups. In the publication [12], the authors proposed grouping correlated features of network traffic components to detect abnormal connections of APT attacks. In addition, several papers [13, 14, 15, 16] presented various methods for detecting APT attacks based on flow.

In this paper, we propose using Network Traffic behavior analysis techniques to detect APT attacks. With analyzing Network Traffic into Flow networks and then using machine learning to detect abnormal connections, we expect to improve the ability to accurately and timely detect APT attack campaigns. Our paper is presented as follows: In section 3, we present a method of detecting APT attacks based on flow behavior analysis. In section 3.1, we discuss methods of analyzing and extracting the behavior of network flow. Section 3.2 presents the flow classification algorithm based on the features and behaviors of flow that are extracted in section 3.1. In section 4, we present the experiments of detecting APT attacks based on network flow using the Random Forest machine algorithm.

3. APT DETECTION BASED ON NETWORK FLOW

3.1 Selecting and extracting abnormal behavior of the flow

In this paper, to analyze and extract features and behaviors of flow from Network traffic, we use the CICFlowMeter tool [17]. The CICFlowMeter tool has the function of analyzing network traffic into network flows and features of network flows. Table 1 below lists the main features of network flow.:

Table 1: List of features

No.	Feature	Description
1	fl_dur	Flow duration
2	tot_fw_pk	Total packets in the forward direction
3	tot_bw_pk	Total packets in the backward direction
4	tot_l_fw_pkt	Total size of packet in forward direction
5	fw_pkt_l_max	Maximum size of packet in forward direction
6	fw_pkt_l_min	Minimum size of packet in forward direction
7	fw_pkt_l_avg	Average size of packet in forward direction

8	fw_pkt_l_std	Standard deviation size of packet in forward direction
9	Bw_pkt_l_max	Maximum size of packet in backward direction
10	Bw_pkt_l_min	Minimum size of packet in backward direction
11	Bw_pkt_l_avg	Mean size of packet in backward direction
12	Bw_pkt_l_std	Standard deviation size of packet in backward direction
13	fl_byt_s	flow byte rate that is number of packets transferred per second
14	fl_pkt_s	flow packets rate that is number of packets transferred per second
15	fl_iat_avg	Average time between two flows
16	fl_iat_std	Standard deviation time two flows
17	fl_iat_max	Maximum time between two flows
18	fl_iat_min	Minimum time between two flows
19	fw_iat_tot	Total time between two packets sent in the forward direction
20	fw_iat_avg	Mean time between two packets sent in the forward direction
21	fw_iat_std	Standard deviation time between two packets sent in the forward direction
22	fw_iat_max	Maximum time between two packets sent in the forward direction
23	fw_iat_min	Minimum time between two packets sent in the forward direction
24	bw_iat_tot	Total time between two packets sent in the backward direction
25	bw_iat_avg	Mean time between two packets sent in the backward direction
26	bw_iat_std	Standard deviation time between two packets sent in the backward direction
27	bw_iat_max	Maximum time between two packets sent in the backward direction
28	bw_iat_min	Minimum time between two packets sent in the backward direction
29	fw_psh_flag	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
30	bw_psh_flag	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
31	fw_urg_flag	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)
32	bw_urg_flag	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
33	fw_hdr_len	Total bytes used for headers in the forward direction
34	bw_hdr_len	Total bytes used for headers in the forward direction
35	fw_pkt_s	Number of forward packets per second
36	bw_pkt_s	Number of backward packets per second
37	pkt_len_min	Minimum length of a flow
38	pkt_len_max	Maximum length of a flow
39	pkt_len_avg	Mean length of a flow
40	pkt_len_std	Standard deviation length of a flow
41	pkt_len_va	Minimum inter-arrival time of packet
42	fin_cnt	Number of packets with FIN
43	syn_cnt	Number of packets with SYN
44	rst_cnt	Number of packets with RST
45	pst_cnt	Number of packets with PUSH
46	ack_cnt	Number of packets with ACK
47	urg_cnt	Number of packets with URG
48	cwe_cnt	Number of packets with CWE
49	ece_cnt	Number of packets with ECE
50	down_up_ratio	Download and upload ratio
51	pkt_size_avg	Average size of packet
52	fw_seg_avg	Average size observed in the forward direction
53	bw_seg_avg	Average size observed in the backward direction
55	fw_byt_blk_avg	Average number of bytes bulk rate in the forward direction
56	fw_pkt_blk_avg	Average number of packets bulk rate in the forward direction
57	fw_blk_rate_avg	Average number of bulk rate in the forward direction
58	bw_byt_blk_avg	Average number of bytes bulk rate in the backward direction
59	bw_pkt_blk_avg	Average number of packets bulk rate in the backward direction
60	bw_blk_rate_avg	Average number of bulk rate in the backward direction
61	subfl_fw_pk	The average number of packets in a sub flow in the forward direction

62	subfl_fw_byt	The average number of bytes in a sub flow in the forward direction
63	subfl_bw_pkt	The average number of packets in a sub flow in the backward direction
64	subfl_bw_byt	The average number of bytes in a sub flow in the backward direction
65	fw_win_byt	Number of bytes sent in initial window in the forward direction
66	bw_win_byt	# of bytes sent in initial window in the backward direction
67	Fw_act_pkt	# of packets with at least 1 byte of TCP data payload in the forward direction
68	fw_seg_min	Minimum segment size observed in the forward direction
69	atv_avg	Mean time a flow was active before becoming idle
70	atv_std	Standard deviation time a flow was active before becoming idle
71	atv_max	Maximum time a flow was active before becoming idle
72	atv_min	Minimum time a flow was active before becoming idle
73	idl_avg	Mean time a flow was idle before becoming active
74	idl_std	Standard deviation time a flow was idle before becoming active
75	idl_max	Maximum time a flow was idle before becoming active
76	idl_min	Minimum time a flow was idle before becoming active

3.2 APT detection method using the Random Forest algorithm

In this paper, to classify flow into normal and APT attack flows as a basis for the conclusion about signs of APT in the system, we use the Random Forest supervised learning algorithm [18]. In the Random Forest algorithm, to select which feature is best suited to be the root node and which features are suitable to be the next internal node, the Random Forest algorithm can use Information Gain or Gini Index. The principle of operation and the classification process of the Random Forest algorithm are detailed in the study [18]. In this paper, we will apply the Random Forest algorithm with default parameters as in the algorithm's structure.

4. EXPRIMENTS AND EVALUATE

4.1. Technical requirements

To implement the algorithm, the programming language used is Python 3.6. With the number of flows greater than one million flows, the project uses the Apache Spark tool to build machine learning models. Apache Spark is an open source tool that is originally developed in 2009 by AMPLab at the University of California. The experimental environment is installed on a computer with the configuration: Intel Xeon 4 Core CPU, 8GB RAM, and 200GB HDD

4.2. Experimental scenarios

Experimental data was collected and analyzed from 29 PCAP files in the Malware Capture CTU-13 dataset. This dataset includes 6 types of malware from APT attacks consisting of Andromeda, Colbalt, Cridex, Dridex, Emotet, and Gh0stRAT [19]. After extracting features and assigning labels, there are 1,722,352 network flows including 791,284 normal network flows and 931,068 network flows of APT attack malware.

In order to create a balance between the data set of the APT attack malware and the normal data set, we collected clean data from servers (including servers of Mocha and recommend system) and PCs (including the normal flows that access to social networking sites such as Facebook, Youtube or normal web pages such as Google, Stack Overflow, etc.). This data is collected within 2 weeks. After the collection process, we obtained 199,885 normal network flows. The experimental scenario is implemented in 3 scenarios as follows:

Scenario 1: Aggregating APT attack malware data from the Malware Capture data set except the Gh0stRAT data set. From this data set, randomly dividing data of each malware at the rate of 70% for training and 30% for testing, and then building model.

Scenario 2: To create objectivity, using the model in scenario 1 to predict the result of identifying the Gh0stRAT data set.

Scenario 3: Aggregating APT attack malware data from the Malware Capture data except the Gh0stRAT and the network flow data collected from PCs accessing social networking sites, newspapers, etc. From this data set, randomly dividing data of each malware at the rate of 70% for training and 30% for testing, and then building model. The network flow data collected from PCs is also divided equally in the 70-30 ratio. Then, two data sets are synthesized together to build the model. From the obtained model, use the testing data set to test.

4.3. Experimental results

4.3.1. Metrics

To evaluate machine learning model, the paper uses the following measurements:

- TP (True Positive) is the number of APT attack network flows that are correctly classified.
- FP (False Positive) is the number of normal network flows that are incorrectly classified as ATP attack flow.
- TN (True Negative) is the number of normal network flows that are correctly classified.
- FN (False Negative) is the number of APT attack network flows that are incorrectly classified as normal flow.

- TPR (True Positive Rate) is the ratio of APT attack network flows that are correctly classified.

$$TPR = \frac{TP}{TP + FN}$$

- FPR (False Postive Rate) is the ratio of normal network flows that are incorrectly classified as ATP attack flow.

$$FPR = \frac{FP}{FP + TN}$$

- TNR (True Negative Rate) is the ratio of normal network flows that are correctly classified.

$$FNR = \frac{FN}{TP + FN}$$

- FNR (False Negative Rate) is the ratio of APT attack network flows that are incorrectly classified as normal flow.

$$TNR = \frac{TN}{FP + TN}$$

- Precision is the ratio of APT attack network flows that are correctly classified among those classified as APT attacks.

$$Precision = \frac{TP}{TP + FP}$$

- Recall is the ratio of APT attack network flows that are correctly classified among those that are actually APT attacks.

$$Recall = \frac{TP}{TP + FN}$$

4.3.2. Experimental results

a) Experimental results for scenario 1

Table 2: Confusion matrix for scenario 1

		Prediction	
		APT	**Normal**
Reality	**APT**	278727	555
	Normal	22447	214992

Table 3: Precision, Recall and Accurancy for scenario 1

Precision	Recall	Accurancy
0.925468334	0.998012761	0.95548468

Table 4: TPR, FPR, FNR, and TNR for scenario 1

TPR	FPR	FNR	TNR
0.9980	0.0945	0.00199	0.905

b) Experimental results for scenario 2

Table 5: Confusion matrix for scenario 2

		Prediction	
		APT	**Normal**
Reality	**APT**	8283	3
	Normal	0	8285

Table 6: Precision, Recall and Accurancy for scenario 2

Precision	Recall	Accurancy
0.99963794	1	0.99981896

Table 7: TPR, FPR, FNR, and TNR for scenario 2

TPR	FPR	TNR	FNR
0.99963794	0.00036206	1	0

c) Experimental results for scenario 3

Table 8: Confusion matrix for scenario 3

		Prediction	
		APT	**Prediction**
Reality	**APT**	131594	1982
	Normal	303164	147688

Table 9: Precision, Recall and Accurancy for scenario 3

Precision	Recall	Accurancy
0.98516201	0.47118683	0.74390344

Table 10: TPR, FPR, FNR, and TNR for scenario 3

TPR	FPR	TNR	FNR
0.47118683	0.00649525	0.99350475	0.52881317

Experimental results in tables 2, 3, 4 show that the APT detection method based on CTU 13 data set has given high efficiency. Especially, the rate of false APT detection is low. This proves that the approach of detecting APT attacks based on network flow has worked well.

Experimental results in tables 5, 6, 7 for scenario 2 show that although Gh0stRAT malware is not used in the training process, the system immediately detected correctly about the existence of the Gh0stRAT malware in the system when applying the training model to detect Gh0stRAT malware.

These experimental results prove that the system was well trained to detect anomalies of APT based on flow features.
For the experimental results of scenario 3 (in tables 8, 9, 10), we see that although the accuracy of APT detection is high, the system still detected relatively many errors due to an imbalance in the ratio between clean data and attack data. This implies that the system still can't learn from other data when the data set isn't standardized.

5. CONCLUSION

In this paper, we presented a method of detecting APT attacks based on network traffic analysis techniques using machine learning. With our proposal, we succeeded in extracting the behavior features of APT malware based on network flow. The experimental results in our paper show that our proposed method is correct and reasonable. This proves that we can find a way to represent information about APT through their statistical features, instead of searching and extracting the characteristics features of APT in network traffic because of these features are difficult to seek and build. In the future, we will analyze and evaluate the statistical features of network traffic in order to optimize the process of selecting and extracting this feature using deep learning techniques.

REFERENCES

[1] E. Code. **Advanced Persistent Threat. Understanding the Danger and How to Protect Your Organization**, *1rd ed.; Elsevier, Amsterdam*, 2012

[2] A. Alshamrani, A. Chowdhary, S, Myneni, D. Huang. A **Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities**. *IEEE Communications Surveys & Tutorials*, vol. 1, pp. 1–29, 2019, https://doi.org/10.3390/s20030731

[3] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, S. Robinson. **Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams**. *In proceedings of the AAAI-17 Workshop on Artificial Intelligence for Cyber Security WS-17-04*. San Francisco, CA USA, February 4–5, 2017.

[4] G. Yan, Q. Li , D. Guo, X. Meng. **Discovering Suspicious APT Behaviors by Analyzing DNS Activities**. *Sensors*, vol. 20, pp. 1-17, 2020.

[5] H.N. Eke, A. Petrovski, H. Ahriz. **The use of machine learning algorithms for detecting advanced persistent threats.** *In proceedings of the 12th International on security of information and networks conference 2019 (SINCONF 2019)*, Sochi, Russia. 12-15 September 2019, pp. 1–8.

[6] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, F.J. Aparicio-Navarro. **Detection of advanced persistent threat using machine-learning correlation analysis**. *Future Generation Computer Systems*. vol. 89, pp. 349–359, 2018. https://doi.org/10.1016/j.future.2018.06.055

[7] B. Stojanovi´c, K. Hofer-Schmitz, U. Kleb. **APT Datasets and Attack Modeling for Automated Detection Methods: A Review**. *Computers &Security*, vol. 92, pp. 1-66, 2020.

[8] WeinaNiu; Xiaosong Zhang; GuoWu Yang; Jianan Zhu; Zhongwei Ren. **Identifying APT Malware Domain Based on Mobile DNS Logging**. *Mathematical Problems in Engineering*. vol. 2017, pp. 1–10, 2017. https://doi.org/10.1155/2017/4916953

[9] Zhao, G.; Xu, K.; Xu, L.; Wu, B. **Detecting APT malware infections based on malicious DNS and traffic analysis**. *IEEE Access*, vol. 3, pp. 1132–1142, 2015.

[10] I. Friedberg, F. Skopik, G. Settanni, R. Fiedler. **Combating advanced persistent threats: from network event correlation to incident detection**. *Computers and Security*, vol. 48, pp. 35–57, 2015.

[11] Do Xuan Cho; Ha Hai Nam. **A Method of Monitoring and Detecting APT Attacks Based on Unknown Domains**. *Procedia Computer Science*, vol. 150, pp. 316-323, 2019. https://doi.org/10.1016/j.procs.2019.02.058

[12] Cho Do Xuan, Lai Van Duong, Tisenko Victor Nikolaevich. **Detecting C&C Server in the APT Attack based on Network Traffic using Machine Learning**. *International Journal of Advanced Computer Science and Applications*, vol. 11, pp. 22-27, 2020.

[13] Abdurrahman Pektas; TankutAcarman. **Botnet detection based on network flow summary and deep learning**. *International Journal of Network Managemen*t,vol. 28, pp.1- 15, 2018.

[14] ŞerifBahtiyar. **A Flow Based Approach to Detect Advanced Persistent Threats in Communication Systems.** *Journal of Natural & Applied Sciences*, vol. 22, pp. 519-528, 2018. https://doi.org/10.19113/sdufbed.98173

[15] J. Lu, K. Chen, Z. Zhuo, X. Zhang. **A temporal correlation and traffic analysis approach for APT attacks detection**. *Cluster Computing*, vol. 20, pp 1–12, 2017.

[16] Do Xuan, Cho, Nguyen, HoaDinh, and Dao, Mai Hoang. **APT Attack Detection Based on Flow Network Analysis Techniques Using Deep Learning**. pp. 1 – 17, 1 Jan. 2020.

[17] CICFlowMeter. Available online: http://www.netflowmeter.ca/netflowmeter.html (accessed on 1 December 2019)

[18] Leo Breiman. **Random Forests**, **Machine Learning**, vol. 45, no. 1, pp. 5- 32, 2001 https://doi.org/10.1023/A:1010933404324

[19] The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic. https://www.stratosphereips.org/datasets-ctu13. [access date 3/1/2020].