2017 Fifth International Symposium on Computing and Networking

Long-term Performance of a Generic Intrusion Detection Method Using Doc2vec

Mamoru Mimura* and Hidema Tanaka*

* National Defense Academy 1-10-20, Hashirimizu, Yokosuka Kanagawa 239-8686 Japan

***Abstract*—Cyber attack techniques in cyberspace are evolving every second, and detecting unknown malicious communication is a challenging task. Pattern-matching-based techniques and using malicious website blacklists are not effective, because the attacker can easily change the traffic pattern or the attack infrastructure. There are many behavior based detection methods which use the characteristic of drive-by-download attacks or C&C traffic. However, many previous methods specialize the attack techniques and the adaptability is limited. Moreover, they have to decide the feature vectors every attack method.**

Accordingly, we propose a generic detection method which is independent of attack methods and does not need devising feature vectors. This method uses Paragraph Vector an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents, and learns the context in proxy server logs.

This paper reveals the long-term performance and the effectiveness to the other dataset. This paper conducts timeline analysis with the dataset which contains captured traffic from Exploit Kit (EK) between 2014 and 2017. This paper also demonstrates cross-dataset validation by showing that an automated feature extraction scheme learned from one dataset can be used successfully for classification on another dataset. The experimental results show the proposed method is effective over three years, and effective on another dataset too. The proposed method achieves an F-measure of 0.95 in the timeline analysis and an F-measure of 0.96 on the other dataset.

I. Introduction

Information and communication technology has been developing rapidly. Many physical devices, vehicles, buildings and other items are embedded with electronics, software, sensors, actuators and network connectivity, which enable these objects to collect and exchange data. These devices are based on the common architecture, and are becoming parts of cyberspace. While these devices enhance convenience and user productivity, it comes with security risks. For instance, many devices have unlisted security vulnerabilities. Thus, cyber threat is proportionately increasing and the public concern is high. Cyber attack techniques in cyberspace are evolving every second, and detecting unknown malicious communication is a challenging task. The "Morris Worm" of November 2, 1988 was one of the first computer worms distributed via the Internet. Internet worms such as the "Code Red" or the "Blaster" used to be a main threat in early first decade of the 2000s. In recent years, Drive-by Download attack (DbD attack) or Spear Phishing attack (SP attack) are main attack techniques in the Internet. After the initial intrusion, the attacker takes the control of the victim's computer over the Internet via a Command and Control (C&C) server. The victim's computer serves as a stepping stone to further deep intrusion. Security analysts in Security Operations Center (SOC) attempt to detect cyber attacks and deal with the incidents. They investigate IDS (Intrusion Detection System) alerts or logs recorded in network devices such as a firewall or a proxy server to detect cyber attacks. In general, intrusion detection techniques on the network are classified roughly into methods using pattern matching and methods using blacklists. The methods using pattern matching are effective, if the malicious traffic contains a unique string pattern. An IDS uses fixed strings or regular expression to describe the signatures. However, recent Exploit Kits (EKs) (e.g. Angler EK, RIG EK) communicate via standard http protocol to imitate normal http communication. An EK is a software kit designed to run on web servers, with the purpose of identifying software vulnerabilities in client machines communicating with it. An EK discovers and exploits vulnerabilities to upload and execute malicious code on the client via standard http protocol. Some queries do not contain the EK specific strings. Therefore, it is difficult to describe the signatures. In this case, an IDS can use the malicious destination server (e.g. Landing site, C&C server) address as the signature. A firewall or a proxy server can also use the malicious destination server address as the blacklist. However, the attacker can change the malicious destination servers easily to evade detection by network devices. Some attackers use compromised hosts as stepping stones. Therefore, using the blacklist does not play critical role. In addition, the malicious server address has to be already-known before the cyber attack. Thus, detecting unknown malicious communication is a challenging task.

There are many behavior-based detection methods to detect unknown malicious communication. These methods capture the characteristics of DbD attacks or C&C traffic, and detect unseen malicious communication. However, many previous methods can detect only DbD attacks or C&C traffic. Because these methods usually use different detection techniques. If attackers change the attack techniques, these previous methods barely detect unseen malicious communication. Besides the security researchers have to devise the feature vectors to capture the characteristics. Furthermore, many previous methods require monitoring all network traffic. Many organizations, however, do not keep all network traffic because the size is too huge. Security incidents often occur in the organizations that did not take the countermeasures adequately. In most cases,

2379-1896/17 $31.00 © 2017 IEEE
DOI 10.1109/CANDAR.2017.109

there are inadequate log files to investigate the incident in the vulnerable organizations. Sometimes we might retrieve only log files on a single proxy server.

Accordingly,we propose a generic detection method which is independent of attack methods and does not need devising feature vectors. This method presumes proxy server logs are written with natural language, and focuses on the characteristic that Neural Network (NN) learns feature vector representation automatically. This method uses Paragraph Vector an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents, and learns the context in proxy server logs. They conducted cross validation and timeline analysis with MWS datasets[1]. However, the malicious traffic in these datasets is not up-to-date, and the long-term performance is unknown. Besides, whether the proposed method is effective on another dataset is not clear.

This paper reveals the long-term performance and the effectiveness to another dataset. This paper conducts timeline analysis with the dataset which contains captured traffic from Exploit Kit (EK) between 2014 and 2017. We used the most up-to-date captured traffic which was downloaded from the website MALWARE-TRAFFIC-ANALYSIS.NET[2]. This paper also demonstrates cross-dataset validation by showing that an automated feature extraction scheme learned from one dataset can be used successfully for classification on another dataset. The experimental results show the proposed method is effective over three years, and effective on another dataset too. The proposed method achieves an F-measure of 0.95 in the timeline analysis and an F-measure of 0.95 on the other dataset.

The main contributions of this paper are three-fold: (1) Verified that the proposed method could detect up-to-date EKs. (2) Verified that the proposed method was effective over three years. (3) Verified that the proposed method was effective on another dataset too.

The rest of the paper is organized as follows. Next section discusses related works and makes clear the difference among this method and previous methods. Section 3 describes Natural Language Processing (NLP) techniques which include Paragraph Vector. Section 4 describes the generic detection method based on the NLP techniques. Section 5 shows experimental results applying the proposed method to the multiple datasets. Section 6 discusses the results, and reveals the long-term performance and the effectiveness to the other dataset.

II. Related Work

This paper aims to detect malicious communication, even if the C&C server address is unknown and the distinctive communication pattern is unknown. The malicious communication includes DbD attacks and C&C traffic. Many previous methods based on the traffic behavior require monitoring all network traffic. However, it is unusual to obtain all network traffic in actual incidents. Therefore, this paper focuses on proxy server logs.

Kruegel et al.[4] categorized URIs by the path, and extracted the parameters from the query strings. Their statistical model learns the features, and detects the statistical outliers as attacks. Their method expects detecting direct attacks to web servers such as buffer over flow, directory traversal, cross-site scripting and so on. The proposed method uses statistical machine learning models for binary classification of malicious communication and benign communication. The proposed method expects indirect attacks such as DbD attacks or SP attacks too.

Choi et al.[5] extracted feature vectors from the domain, the path and so on from URLs, and proposed a method using machine learning models to classify malicious URLs and benign URLs. Their method uses not only proxy server logs but also the URL popularity, the contents, the DNS traffic or any other traffic. The proposed method uses only proxy server logs and does not require any other information obtained from the outside. The proposed method does not even demand devising feature vectors.

Ma et al.[6] extracted feature vectors from the host name, the top level domain name, the path and so on included in URLs, and proposed online learning algorithm to classify malicious URLs and benign URLs. Their method divides URLs into tokens by the delimiter such as “dot” (.), “slash” (/), “question mark” (?), “equal” (=), “and” (&) and so on. The proposed method uses the similar techniques to obtain tokens from URLs. These tokens serve as words to construct a corpus. Their method requires not only proxy server logs but also searching the whois database for IP address and domain name registration information, blacklists, the geographical feature, the bandwidth speed and so on. The proposed method uses only proxy server logs. This method does not even demand devising feature vectors.

Huang et al.[7] extracted feature vectors from the structure, the characteristic strings and the brand name included in URLs to detect malicious URLs with machine learning. Their method aims to detect phishing URLs. The proposed aims to detect DbD attacks and C&C traffic, however, is not limited to these attacks. The proposed method does not even demand devising feature vectors.

Zhao et al.[8] focused on the cost to force users to analyze and label malicious traffic, and proposed an online active learning framework which updates the classifier to detect malicious URLs. Their method uses the whois database for domain name registration information, blacklists and so on. The proposed method uses only proxy server logs and does not require any other information obtained from the outside.

Invernizzi et al.[9] built a network graph from IP addresses, domain names, FQDNs, URLs, paths, file names and so on. Their method focuses on the correlation among nodes to detect malware distribution. Their method uses only the parameters obtained from proxy server logs. However, their method has to cover many range of IP addresses, and performs in large-scale networks such as ISPs. In addition, their method needs the downloaded file types. The proposed method performs at any scale and does not need the downloaded file types.

Nelms et al.[10] focused on DbD attacks, and extracted

the Location field, the Referrer field and so on from http Request messages and Response messages to build a URL transfer graph. They proposed a trace back system which could go back to the source from the URL transfer graph. Their method uses the hop count, the domain age, common features of the domain names to detect malicious URLs. The proposed method uses only proxy server logs and does not require any other information obtained from the outside. In addition, the proposed method can detect not only DbD attacks but also any other types of attacks.

Bartos et al.[11] categorized proxy server logs into flows, and extracted many features from the URLs, the paths, the queries, the file names and so on. They proposed how to learn the feature vectors to classify malicious URLs. Their method can decide the optimum feature vectors automatically. However, their method demands devising basic features for learning. The proposed method does not even demand devising basic features.

Mimura et al.[12] categorized proxy server logs by FQDNs to extract feature vectors, and proposed a RAT (Remote Access Trojan or Remote Administration Tool) detection method using machine learning techniques. Their method uses the characteristic that RATs continues to access the same path regularly. However, their method performs for only C&C traffic. The proposed method can detect not only C&C traffic but also any other types of attacks.

III. Natural Language Processing (NLP) Technique

A. Word2vec

To calculate various measures to characterize a text, we have to transform the text into a vector. Word2vec[3] is a model that produces word embeddings. Word embedding is the collective name for a set of language modeling and feature learning techniques in NLP where words from the vocabulary are mapped to vectors of real numbers. This model is a shallow, two-layer neural network that is trained to reconstruct linguistic contexts of words. This model takes as its input a large corpus of text and produces a vector space, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to each other in the space. Word2vec is based on the distributional hypothesis, which motivates that the meaning of a word can be gauged by its context. Thus, if two words occur in the same position in two sentences, they are very much related either in semantics or syntactics. Word2vec utilizes two algorithms to produce a distributed representation of words. One is Continuous-Bag-of-Words (CBoW), and the other is skip-gram. In the CBoW algorithm, the model predicts the current word from a window of surrounding context words. In the skip-gram algorithm, the model uses the current word to predict the surrounding window of context words. Word2vec enables to calculate semantic similarity between two words and infer similar words semantically. However, Word2vec is a model that merely produces word embeddings. To calculate semantic similarity between two documents, this model has to be extended.

B. Paragraph Vector (Doc2vec)

An extension of Word2vec to construct embeddings from entire documents has been proposed[13]. This extension is called Doc2vec or Paragraph2vec and has been implemented. Doc2vec is based on the same distributional hypothesis, which motivates that the meaning of a sentence can be gauged by its context. Thus, if two sentences occur in the same position in two paragraphs, they are very much related either in semantics or syntactics in the same way. Doc2vec utilizes two algorithms to produce Paragraph Vector a distributed representation of entire documents. One is Distributed-Memory (DM), and the other is Distributed-Bag-of-Words (DBoW). DM is the extension of CBoW, and the only change in this model is adding a document ID as a window of surrounding context words. DBoW is the extension of skip-gram, and the current word was replaced by the current document ID. Doc2vec enables to calculate semantic similarity between two documents and infer similar documents semantically. Some implementations support also inference of document embeddings on unseen documents. This function is important to develop a practical system to detect unseen malicious communication. Because, unseen malicious communication might include an unknown word (e.g. newly-changed FQDN, random strings).

IV. Proposed Method

A. Separating logs with spaces

The key idea of the proposed method is processing proxy server logs as a natural language. In order to put it into practice, proxy server logs have to be separated into words. Most proxy server logs include date and time at which transaction completed, request line from the client (includes the method, the URL and the user agent), HTTP status code returned to the client and size of the object returned to the client. The client means user's computer which connects to servers over the Internet. A proxy server records the contents on a line in chronological order. The line originates the request from an internal client and is coupled with the response from the server.

The proposed method divides proxy server logs into HTTP status code, request line from the client, size of the object returned to the client and user agent. Furthermore, the request line is divided into method, URL and protocol version.

B. Separating URLs with spaces

A URL is the most important element to detect malicious communication. Therefore, the proposed method separates URLs with spaces. Fig. 1 shows an example of a URL leaving a space between words.

First, the proposed method divides a URL into scheme, FQDN (Fully Qualified Domain Name) and path which includes query strings. After that, this method separates the FQDN by "dot" (.). Then we can derive top level domain name, sub domain name and so on, which means the country,

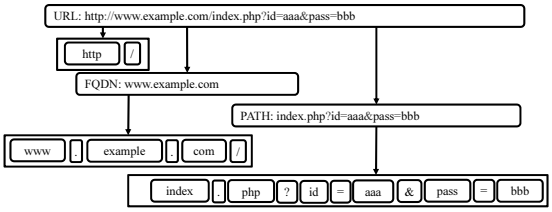

Fig. 1. An example of a URL leaving a space between words.

the organization, the use or the purpose (e.g. www, mail). This method separates the path by "slash" (/) and "dot" (.), and also separates the query string by "question mark" (?), "equal" (=) and "and" (&). Then, we can derive the directory name, the file name, the extension from the path. We can also derive the variable names and the values from the query string, which are used in the running program on the server. This method excludes the values used only once from query strings. This method leaves the delimiters as a word to construct a corpus. Because the delimiters are related to the contiguous word meanings. For instance, some delimiters such as "slash" (/) are closely related to the structure of the URL.

In conclusion, the proposed method divides URLs into words by the delimiters which are "dot" (.), "slash" (/), "question mark" (?), "equal" (=) and "and" (&). These words construct a corpus to describe proxy server logs.

C. Overview

Fig. 2 shows an overview of the proposed method. First, the proposed method constructs a corpus from malicious proxy server logs and benign proxy server logs. Both logs are separated by the previously mentioned method. Then, the Doc2vec constructs a vector space from the corpus, and converts both proxy server logs into vectors with the labels. These labeled vectors are training data for classifiers. The classifiers are Support Vector Machine (SVM), Random Forests (RF) and Multi-Layer Perceptron (MLP).

A SVM model is a representation of the training data as points in space, mapped so that the training data of the separate categories are divided by a clear gap that is as wide as possible. Test data are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. RF are an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. MLP is a class of feedforward artificial neural network, which consists of at least three layers of nodes. Each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training.

These are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training data, each labeled as belonging to one or the other of two categories, these training algorithms build a model that assigns new examples to one category or the other. After that, we convert unknown

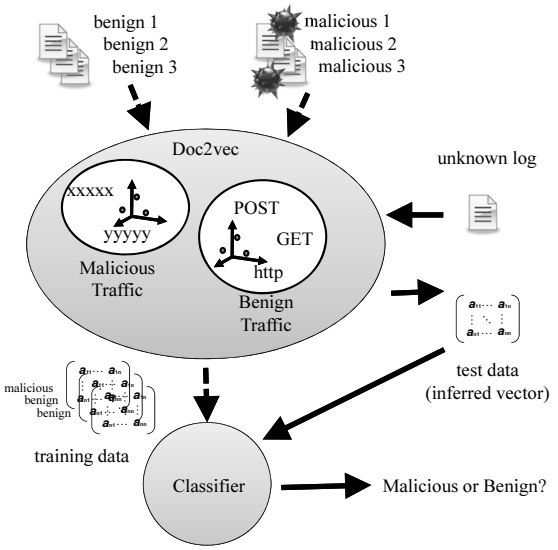

Fig. 2. An overview of the proposed method.

TABLE I
THE PARAMETERS FOR THE DOC2VEC MODEL

Dimensionality of the feature vectors	100
Window	15
Number of epochs	30
Training algorithm	DBoW

proxy server logs into vectors. These unlabeled vectors are test data for the classifiers. Finally, we input these unlabeled vectors to the classifiers, and can obtain a predicted label. The predicted label is either malicious or benign.

D. Implementation

The proposed method was developed by Python-2.7 with open source machine learning libraries, gensim-1.01[14], scikit-learn-0.18.0[15] and chainer-1.23[16].

Gensim is a Python library to realize unsupervised semantic modelling from plain text, and includes a Doc2vec model. TABLE I shows the parameters for the Doc2vec model. We set the dimensionality of the feature vectors 100, and chose DBoW which was the extension of skip-gram. The window is the maximum distance between the predicted word and context words used for prediction within a document.

Scikit-learn is a machine-learning library for Python that provides tools for data mining with a focus on machine learning, and supports SVM and RF. The proposed method uses a SVC function with a liner kernel for SVM. This method also uses a RandomForestClassifier function for RF.

Chainer is a flexible Python framework for neural networks, which supports MLP with CUDA computation. We use CUDA 8.0 and cuDNN-8.0. TABLE II shows the parameters for the MLP model. The number of input layer units is the dimensionality of the test data. Thus, the dimensionality is 100 as we mentioned before. The number of labels is 2,

TABLE II
THE PARAMETERS FOR THE MLP MODEL

Number of input layer units	100
Number of hidden layer units	500
Number of labels	2
Activation function	ReLU
Dropout ratio	0
Minibatch size	100
Optimizer	Adam

namely benign or malicious. ReLU (Rectified Linear Unit) is an activation function defined as follows.

$$f(x) = x^+ = \max(0, x)$$

It is also known as a ramp function and has been used in convolutional networks more effectively than the widely used logistic sigmoid. Adam (Adaptive moment estimation) is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments[17]. This method is well suited for problems that are large in terms of data and parameters, and also appropriate for non-stationary objectives and problems with very noisy and sparse gradients.

The proposed method uses cross entropy as follows to define the loss function in optimization.

$$E = -\sum_{n}^{N}\sum_{i}^{D} t_{ni} \log y_i$$

N is number of the data, and *D* is number of the output layer units. *y* means predicted label, and *t* means true label.

V. EXPERIMENT

A. Dataset

To reveal the effectiveness to up-to-date EKs, we use captured pcap files from EKs between 2014 and 2017, which were downloaded from the website MALWARE-TRAFFIC-ANALYSIS.NET[2]. We chose some EKs which communicate via standard http protocol to imitate normal http communication. We name these pcap files MTA dataset. We also use the D3M (Drive-by Download Data by Marionette) dataset and the NCD (Normal Communication Data in MWSCup 2014) for the cross-dataset validation. These datasets are parts of MWS datasets[1], and include pcap files. The MTA and the D3M contain malicious traffic and the NCD contains benign traffic. TABLE III shows the detail.

The MTA is a set of packet traces downloaded from the website. This dataset includes traffic from many EKs. (e.g. Angler EK, Neutrino EK, Magnitude EK, RIG EK, Nuclear EK) This dataset contains the traffic from the latest version of EKs. The D3M is a set of packet traces collected from the web-client, high-interaction honeypot system, which is based on Internet Explorer on Windows OS with several vulnerable

TABLE III
THE DETAIL OF THE DATASETS.

MTA		D3M	
year	size	year	size
2014	238M	2010	130M
2015	186M	2011	24.8M
2016	373M	2012	33.2M
2017	109M	2013	14.6M
-	-	2014	23.3M
-	-	2015	334M

plugins, such as Adobe Reader, Flash Player, Java and so on. This data focuses on DbD attacks caused by crawling malicious web sites according to threat transitions from exploiting OS services remotely. The datasets contain packet traces for the two periods; infection and after infection. This dataset includes EKs (e.g. Blackhole EK, Elenore, Mpack). However these EKs are not up-to-date.

The proposed method aims to detect malicious communication from proxy server logs. Thus, we have to convert these pcap files into pseudo proxy server logs. First, we extracted http traffic from these pcap files. Then, we coupled the requests and the response, and made the pseudo proxy server logs in the standard log format. After that, we compounded the malicious logs and the benign logs into datasets at the same rate. We split the dataset into training data and test data to conduct 10-fold cross validation and timeline analysis. The proposed method uses only training data to construct a corpus. Because, we presume that the test data is unknown communication. We convert a bag of 10 simply consecutive lines into a vector. In the timeline analysis, we chose an annual traffic as training data, and the subsequent traffic is the test data.

B. Metrics

Three evaluation metrics are used: Precision (P), Recall (R) and F-measure (F). These metrics are used to performance of every class of communication.

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

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

$$F - measure = \frac{2Recall \times Precision}{Recall + Precision}$$

TP (True Positive), *FP* (False Positive), *TN* (True Negative) and *FN* (False Nagative) are defined as shown in TABLE IV

TP is the number of instances correctly classified as benign or malicious, *TN* is the number of instances correctly classified as Not-benign or Not-malicious. *FP* is the number of instances incorrectly classified as benign or malicious, *FN* is the number of instances incorrectly classified as Not-benign or Not-malicious.

TABLE IV
CONFUSION MATRIX FOR TWO POSSIBLE OUTCOMES.

		True label	
		Positive	Negative
Predicted label	Positive	*TP*	*FP*
	Negative	*FN*	*TN*

TABLE V
EXPERIMENT ENVIRONMENT

CPU	Core i7-5820K 3.3GHz
Memory	DDR4 SDRAM 24GB
GPU	GeForce GTX980/4G
OS	Windows 8.1

C. Experimental Environment

TABLE V shows the experiment environment. The proposed method needs only a simple computer, and does not require special equipment.

D. Result

TABLE VI shows the results of the 10-fold cross-validation. Good results have been obtained under each classifier. There was no marked difference among the results.

TABLE VII shows the results of the timeline analysis. In the case that we used the 2014's for training data and the 2015's for test data, the both F-measures have reached 0.99. In the case that we used the 2016's for test data, the both F-measures have reached 0.96. In the case that we used the 2017's for test data, the benign F-measure has reached 0.95 and the malicious one has reached 0.96.

In the case that we used the 2015's for training data and the 2016's for test data, the both F-measures have reached 0.95. In the case that we used the 2017's for test data, the both F-measures have reached 0.93. In the case that we used the 2016's for training data and the 2017's for test data, the both F-measures have reached 0.98. In summary, the proposed method was effective over three years.

TABLE VIII shows the results of the cross-dataset validation. In the case that we used the MTA for training data and the D3M for test data, the both F-measures have reached 0.95. In the case that we used the D3M for training data and the MTA for test data, the both F-measures have reached 0.96. There was no marked difference among the results.

VI. DISCUSSION

A. Accuracy

As the results of the experiments, the proposed method was effective. This is because there are some differences in words and the structure. The proposed method divides URLs into words by the delimiters, and Doc2vec illustrates the ratio of the co-occurrence probabilities of the two words within the window size. Some delimiters are closely related to the structure of the URL. The proposed method learns the structure automatically, even if a human makes no clear indication.

TABLE VI
THE RESULT OF THE 10-FOLD CROSS-VALIDATION.

classifier	NCD (Benign)			MTA (Malicious)		
	P	R	F	P	R	F
SVM	1.00	0.92	0.96	0.93	1.00	0.96
RF	0.96	0.96	0.96	0.97	0.96	0.96
MLP	0.99	0.97	0.98	0.96	0.99	0.97

TABLE VII
THE RESULT OF THE TIMELINE ANALYSIS.

training data	test data	classifier	NCD (Benign)			MTA (Malicious)		
			P	R	F	P	R	F
2014	2015	SVM	1.00	0.91	0.95	0.92	1.00	0.96
		RF	0.85	0.93	0.89	0.93	0.85	0.89
		MLP	1.00	0.98	**0.99**	0.98	1.00	**0.99**
2014	2016	SVM	0,97	0.93	0.95	0.93	0.97	0.95
		RF	0.97	0.95	**0.96**	0.95	0.97	**0.96**
		MLP	0.92	0.96	0.94	0.96	0.91	0.93
2014	2017	SVM	1.00	0.91	**0.95**	0.92	1.00	**0.96**
		RF	1.00	0.90	**0.95**	0.90	1.00	0.95
		MLP	0.90	0.93	0.92	0.92	0.89	0.90
2015	2016	SVM	1.00	0.90	**0.95**	0.91	1.00	**0.95**
		RF	0.85	0.95	0.90	0.94	0.83	0.88
		MLP	0.93	0.94	0.94	0.94	0.94	0.94
2015	2017	SVM	1.00	0.81	0.90	0.82	1.00	0.90
		RF	0.95	0.91	**0.93**	0.92	0.95	**0.93**
		MLP	0.87	0.88	0.87	0.89	0.88	0.89
2016	2017	SVM	1.00	0.93	0.96	0.93	1.00	0.97
		RF	0.93	0.98	0.96	0.98	0.93	0.96
		MLP	0.96	1.00	**0.98**	1.00	0.97	**0.98**

There are some causes of the false positives and false negatives. The primary cause was the sites which provided web APIs (e.g. Authentication, Streaming). A web API is an application programming interface (API) for either a web server or a web browser. A web API produces interactive communication with many parameters to provide a coating high functionality and convenience. This behavior is similar to Exploit Kits or C&C traffic. The attacker needs many parameters to control victim's computer at his will. Thus, Exploit Kits and C&C traffic have to produce interactive communication with many parameters.

The other cause was the sites which provided update programs or pattern files (e.g. Anti Virus Scanner). This behavior is similar to downloading malware and malware infections. We can mitigate these false positives with the whitelist.

Besides, some benign traffic was mixed in the malicious traffic. In a sense, these impurities are not the cause. We can enhance purity of the malicious traffic to improve the accuracy.

TABLE VIII
THE RESULT OF THE CROSS-DATASET VALIDATION.

training data	test data	classifier	NCD (Benign)			MTA (Malicious)		
			P	R	F	P	R	F
MTA	D3M	SVM	0.97	0.91	0.94	0.92	0.97	0.94
		RF	0.80	0.95	0.87	0.94	0.76	0.84
		MLP	0.94	0.96	**0.95**	0.96	0.94	**0.95**
D3M	MTA	SVM	0.96	0.96	**0.96**	0.96	0.96	**0.96**
		RF	0.86	0.96	0.91	0.96	0.85	0.90
		MLP	0.92	0.97	0.94	0.97	0.92	0.94

B. Durability

The proposed method was effective over three years. This result means that these EKs have common universal characteristics. The MTA dataset contains many EKs, Angler-EK, Fiesta-EK, FlashPack-EK, Magnitude-EK, Neutrino-EK, Nuclear-EK, RIG-EK, and so on. Therefore, the proposed method can learn the universal characteristics of EKs.

The proposed method learns the difference of normal communication and malicious communication automatically with neural networks. In neural networks, it is difficult to specify what feature of an input data a specific feature map captures. This means that an attacker cannot recognize the features either. We tried some experiments to specify the feature. We conducted the same experiments without some elements (e.g. FQDN, User Argent). However, there was no notable change in the performance. This means that our method does not rely on a specific element. Therefore, an attacker has no effective countermeasure to evade this method. The only option is imitating normal communication completely. Thus, our method is effective and durable in the long term.

C. Practical use

The proposed method was effective in the other dataset. This means the proposed method is powerful and versatile. Many previous methods require monitoring all network traffic. Many organizations, however, do not keep all network traffic because the size is too huge. Actually, we would have to investigate insufficient information such as proxy server logs. Thus, these methods are not practical. The proposed method does not require monitoring all network traffic, and only uses malicious and benign proxy server logs. In this paper, we obtained pseudo proxy server logs from pcap files. We can obtain these malicious pcap files easily from the websites, which disclose malicious traffic data. We can also obtain benign pcap files easily from everywhere. Thus, the proposed method has few constraints in practical use.

VII. CONCLUSION

In this paper, we described how to construct a corpus from proxy server logs and the generic detection method using supervised learning models to classify normal communication and malicious communication. This paper conducted cross-validation and timeline analysis with the MTA dataset which contains captured traffic from EKs between 2014 and 2017. This paper also demonstrated cross-dataset validation which uses the MTA dataset and the D3M dataset. Consequently, the proposed method can detect up-to-date traffic from EKs in proxy server logs. We verified that the proposed method was effective over three years, and effective on the other dataset too. The proposed method achieved an F-measure of 0.95 in the timeline analysis and an F-measure of 0.96 on the other dataset.

In this time, we presumed proxy server logs are written with natural language. We can presume any other logs such as IDS alerts, firewall logs, SIEM (Security Information and Event Management) events are written with natural language in the same manner. This might enable to classify the detail of the any attacks automatically.

REFERENCES

[1] Hatada, M., Akiyama, M., Matsuki, T., Kasama, T., *Empowering Anti-malware Research in Japan by Sharing the MWS Datasets*, IPSJ Journal, Vol.56, No.9, 2015.

[2] MALWARE-TRAFFIC-ANALYSIS.NET, http://www.malware-traffic-analysis.net/

[3] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and Dean, J., *Distributed Representations of Words and Phrases and Their Compositionality*, In Advances in Neural Information Processing Systems, pp.3111-3119, 2013.

[4] Kruegel, C., and Vigna, G., *Anomaly Detection of Webbased Attacks* Proc. 10th ACM Conference on Computer and Communications Security, pp.251-261, 2003.

[5] Choi, H., Zhu, B.B., and Lee, H., *Detecting Malicious Web Links and Identifying Their Attack Types* Proc. 2nd USENIX Conference on Web Application Development, pp.1-11, 2011.

[6] Ma, J., Saul, L.K., Savage, S., and Voelker, G.M., *Learning to Detect Malicious URLs* ACM Trans. on Intelligent Systems and Technology, Vol.2, No.3, Article 30, 2011.

[7] Huang, H., Qian, L., and Wang, Y., *A SVM-based Technique to Detect Phishing URLs* Information Technology Journal, Vol.11, No.7, pp.921-925, 2012.

[8] Zhao, P., and Hoi, S.C., *Cost-sensitive Online Active Learning with Application to Malicious URL Detection* Proc. 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.919-927, 2013).

[9] Invernizzi, L., Miskovic, S., Torres, R., Saha, S., Lee, S., Mellia, M., Kruegel, C., and Vigna, G., *Nazca: Detecting Malware Distribution in Large-scale Networks* Proc. Network and Distributed System Security Symposium, 2014.

[10] Nelms, T., Perdisci, R., Antonakakis, M., and Ahamad, M., *Webwitness: Investigating, Categorizing, and Mitigating Malware Download Paths* Proc. 24th USENIX Security Symposium, pp.1025-1040, 2015.

[11] Bartos, K., and Sofka. M., *Optimized Invariant Representation of Network Traffic for Detecting Unseen Malware Variants* Proc. 25th USENIX Security Symposium, pp.806-822, 2016.

[12] Mimura, M., Otsubo, Y., Tanaka, H., and Tanaka, H., *A Practical Experiment of the HTTP-Based RAT Detection Method in Proxy Server Logs* Proc. 12th Asia Joint Conference on Information Security, 2017.

[13] Le, Q., and Mikolov, T., *Distributed Representations of Sentences and Documents*, Proc. 31st International Conference on Machine Learning, pp.1188-1196, 2014.

[14] gensim, https://radimrehurek.com/gensim/

[15] scikit-learn. http://scikit-learn.org/

[16] Chainer, https://chainer.org/

[17] Kingma, D.P., and Ba, J., *Adam: A Method for Stochastic Optimization*, Proc. 3rd International Conference for Learning Representations, 2015.