Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 154 (2019) 543–548

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Congress of Information and Communication Technology, ICICT 2019

A Method of Detecting Storage Based Network Steganography Using Machine Learning

Cho D.X[a,b],*, Thuong D.T.H[a] , Dung N.K[c]

[a]*Information Security dept. Posts and Telecommunications Institute of Technology Hanoi, Vietnam*
[b]*Information Security dept. FPT University, Hanoi, Vietnam*
[c]*Information Technology dept. Thanh Dong University, Hanoi, Vietnam*

Abstract

Today, the techniques of network steganography are widely applied. In addition to the outstanding advantages about the ability to hide and transmit secret information, it has a huge disadvantage that is being exploited by hackers to transmit information or communicate with the control host. Network steganography storage method is one of the network steganography techniques that is being much applied. Due to the characteristics of the storage based network steganography are different from other network steganography techniques, the detection of this techniques is difficult. The traditional tools and methods used to detect steganography are difficult to detect the signs of steganographic packets that use this technique. Therefore, in this paper, the authors propose using machine learning to detect abnormal behavior of steganographic packets.

Keywords: *Information Security; network steganography; detection storage based network steganography; machine learning.*

1. Overview of Detection Network Steganography

Network steganography is the technique of hiding secret data in legitimate transmissions in communication networks without destroying the used hidden data carrier.

* Corresponding author. Tel.: +84965068868;
E-mail address: chodx@ptit.edu.vn; chodx@fpt.edu.vn

10.1016/j.procs.2019.06.086

In the document [1], Wojciech Mazurczyk et al. presented a number of network steganography techniques and classified them based on how the secret data are hidden into the carrier. Network steganography is classified into storage, timing and hybrid methods. Storage methods hide secret data in user data or by modifying protocol fields. Timing methods hide secret data in the protocol messages timing or the packets timing. Hybrid methods combine two methods. Storage methods are the most popular methods. In this paper, the authors propose the detection method of storage based network steganography.

Each network steganography method can be characterized by three features:

Steganography bandwidth: how much secret data one is able to send per time unit.

Undetectability: an inability of an adversary to detect a stegano-gram inside a carrier.

Robustness: the amount of alteration a stegano-gram can withstand without destroying the secret data.

For each network steganography method, there is always a trade-off between maximizing steganography bandwidth and still remaining undetected (and retaining an acceptable level of robustness).

Through this section, can notice that, in recent years, with the development of various hiding methods, storage based network steganography has become a new kind of threat for network security. Therefore, the detection of storage based network steganography is very necessary.

Related to detection of storage based network steganography has three main approaches below:

Signature-based detection: defines patterns or set of rules that are stored in the database to decide whether the pattern is a steganography or not. It is used to detect only the known attacks. Unknown attacks cannot be detected using signature based detection.

Statistics-based detection: use a variety of statistical calculations on the data to detect the embedded message.

Detection based on machine learning: This is an advanced approach. This approach use classification algorithms and extracted features to detect steganography.

In this paper, the authors propose use machine learning to detect storage based network steganography.

2. Related Works

Signature-based detection: Mike Sieffert proposed use IDS to detect network steganography in data sections of packets [2].

Statistics-based detection:

Dittmann J Statistical proposed steganalysis for LSB (Least Significant Bits) based VoIP steganography [4]

The MAP based detection method was proposed in [3]. This method use Markov model to detect network steganography in TCP ISN and RST fields of TCP packets.

Detection based on machine learning:

Use SVM algorithm method to detect network steganography in TCP/IP was proposed by Taeshik Sohn [5].

Use Naive Bayes algorithm method to detect secret information hidden in TCP/IP header [6].

Signature-based and statistics-based detection methods only effectively detect when information is hidden in user data. Using SVM and Naive Bayes algorithms is quite effective. In this paper, the authors propose using Random Forest algorithms to detect storage based network steganography.

3. A method of Detecting Storage Based Network Steganography Using Machine Learning

3.1. Proposed detection model based on machine learning

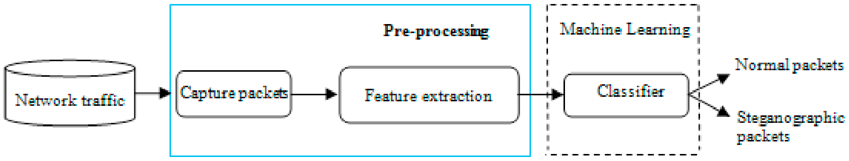

Fig. 1. Detection model based on machine learning

In order to detect storage based network steganography, it is necessary first to collect packets in network traffic in real time. There are some available tools to assist in capturing and analyzing network traffic components such as Wireshark, tcpdump, tshark, Network Miner. Each tool has its own advantages and disadvantages. In this paper, the authors using Wireshark to capture network packets.

Then the features will be extracted from each packet. Those features are the most basic abnormal behavior of packets. These behaviors will be the best basis to determine which packets are normal or steganographic. In this paper, besides reusing the features described in the paper [5], the authors will propose some new features to enhance the ability to detect secret information hidden in the packages. Details of these features will be presented in next section of this paper.

Finally, after obtaining a data set of feature records which are characteristic of packets, these records will be classified as either normal or steganographic by the machine learning algorithm. Machine learning approach to detecting abnormal behavior is not a new method, but in the field of detecting network steganography has not been applied much. In this paper, the authors propose using Random Forest algorithms into this model. In the field of detecting network steganography, there are no research presenting the application of the Random Forest algorithm to the detection of abnormal behavior.

Thus, from Figure 1, can see that the process of detecting storage based network steganography using learning machine consists of two main parts: i) select the features list and extract those features; ii) apply Random Forest algorithm to classify packets. Next, the paper will detail to clarify these two parts.

3.2. The features description

The paper [5] presented some features and characteristics of these attributes to detect abnormal behavior of steganographic packets using the storage based network steganography technique.

Table 1 below shows a list of seven important features that have been applied to detect storage based network steganography in TCP/IP.

Table 1. 7 features are used to detect storage based network steganography in TCP/IP

No.	Category	Features	Type	Describe
1	IP header	IP Identification	Integer	This field is used for uniquely identifying the group of fragments of a single IP datagram
2		IP Flags	Integer	This field is used to control or identify fragments
3		IP Fragment Offset	Integer	This field specifies the offset of a particular fragment relative to the beginning of the original un-fragmented IP datagram
4		IP header Checksum	Integer	This field is used for error-checking of the header
5	TCP header	TCP Control Flag	Integer	This field indicate a particular connection state or provide additional information
6		TCP Checksum	Integer	This field is used for error-checking of the header, the Payload and a Pseudo-Header
7		TCP Sequence Number	Integer	This field counts bytes in the data stream

From table 1, can see that the features are proposed by the authors is only focused on detecting secret data hidden in TCP header and IP header. Therefore, if hackers use new network steganography techniques, these attributes will not bring high accuracy detection results. In this paper, the authors propose a number of new features that are capable of detecting a new storage based network steganography technique (hide information in ICMP payload). Table 2 lists the new features that the authors recommend using.

Table 2. 5 features are proposed to detect storage based network steganography in ICMP

No.	Category	Features	Type	Describe
1	ICMP payload	* ICMP Checksum Status	Integer	This field describes status of ICMP Checksum
2		* ICMP Identifier (BE)	Integer	This field is used to identify a session
3		* ICMP Sequence number (BE)	Integer	This sequence number field is displayed in big endian (BE) formats. This field value is incrementing from one ICMP echo request/reply to the next.
4		* ICMP Sequence number (LE)	Integer	This sequence number field is displayed in little endian (LE) formats This field value is incrementing from one ICMP echo request/reply to the next.
5		* ICMP Data	Integer	This field contains data section

3.3. Random Forest algorithms

Regarding using machine learning to detect abnormalities signs, there are several main algorithms such as SVM, Naive Bayes [9]. Depending on the research subjects, different algorithms can be used. In the paper [5] proposed using SVM algorithm to detect network steganography in TCP/IP. Moreover, in the paper [6] proposed using Naive Bayes algorithm to detect secret data in TCP/IP header. In this paper, the authors propose using Random Forest algorithms to detect storage based network steganography.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks. This algorithm operate by constructing a multitude of decision trees. Those decision trees are not the same. Each tree will run and give independent result [9].

According to Leo Breiman [7] and Ibrahim A Ibrahim [8], random forest algorithm includes training and testing phases. Figure 2 shows the structure of Random Forest algorithm.

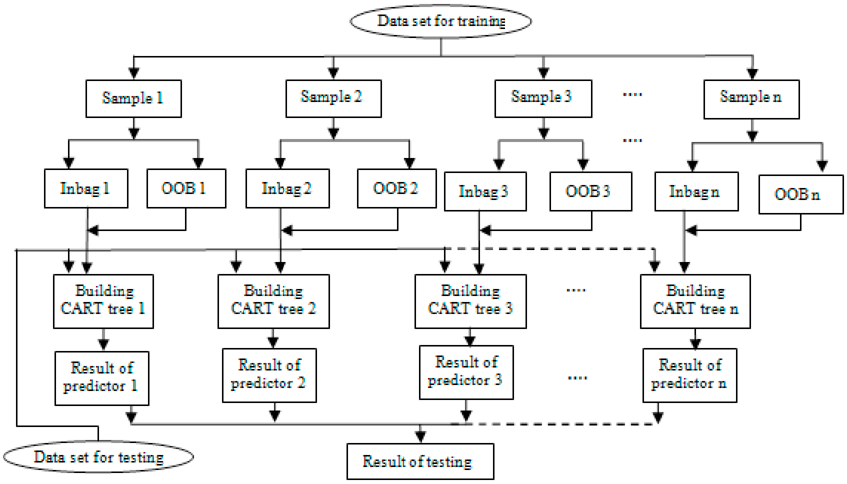

Fig. 2. Structure of the RFs algorithm

In the training phase, this algorithm starts by drawing multiple bootstrap samples (N) from the original data and then creates a number of unpruned Classification and Regression Tree (CART) for each N. Not all data of samples are involved in the tree-creating process. About one-third data of each sample are left out in the OOB (out of bag) data. The OOB data is used to estimate the prediction error as trees are added to the forest in the construction phase. After creating the final splits, the data are predicted at each bootstrap by using the tree growth technique.

In the testing phase, the testing data are distributed in the forest to start the prediction procedure. The final nodes are predicted by determining the average aggregation of the predictors through all trees.

4. Installation and Experiments

4.1. Installation and data set

Installation requirements:

Software Installations: Python version 3.6.5

Hardware requirements: RAM 8GB; CPU Intel Core i3 1.70GHz.

Training and testing data sets: The data used for training and testing in algorithm for detecting storage based network steganography has been collected as follows:

Data set of normal packets: 25.000 normal packets have been collected on Internet.

Data set of steganographic packets: 30.000 steganographic packets using three storage based network steganography techniques (hidden data in TCP header and IP header with cover_tcp tool; hidden data in ICMP payload with Pingtransfer tool).

All packets are extracted 12 features which describe in table 1 and table 2. The data is divided into 2 data sets, 80% of the data is used for training in the classification model; 20% of the data is used for testing. The normal packets are labeled ‘0’ and the steganographic packets are labeled ‘1’.

4.2. Experiments

Followings are some experiment results of the method detecting storage based network steganography using machine learning with some scenarios:

Using different data sets with different ratios of normal versus steganography packets:

A: 25 000 normal packets + 15 000 steganographic packets (including 5 000 packets hidden data in TCP header, 5 000 packets hidden data in IP header and 5 000 packets hidden data in ICMP payload).

B: 25 000 normal packets + 30 000 steganographic packets (including 12 000 packets hidden data in TCP header, 8 000 packets hidden data in IP header and 10 000 packets hidden data in ICMP payload).

Using different algorithms include Random Forest, SVM and Naïve Bayes: The authors will use all three algorithms into the detection model and evaluate the experiment results to conclude whether Random Forest is indeed more effective than the other algorithms.

The experiment results with data set A are shown in the table 3 below:

Table 3. Experiment results on testing data set A

Algorithm	TP (%)	FP (%)	FN (%)	TN (%)	Accuracy (%)	Training time (s)	Prediction time (s)
Random Forest	99.95	0.10	0.05	99.90	99.93	1.66	0.06
SVM	100.0	1.90	0.00	98.10	98.50	48.27	1.79
Naïve Bayes	85.16	0.24	14.84	99.76	95.95	0.29	0.03

The experiment results with data set B are shown in the table 4 below:

Table 4. Experiment results on testing data set B

Algorithm	TP (%)	FP (%)	FN (%)	TN (%)	Accuracy (%)	Training time (s)	Prediction time (s)
Random Forest	99.90	0.04	0.10	99.96	99.95	2.37	0.09
SVM	98.89	0.96	1.11	99.04	98.95	50.59	1.83
Naïve Bayes	98.50	0.85	1.50	99.15	98.82	0.06	0.01

From Table 3 and Table 4, the authors will evaluate the experiment results in three aspects:

Accuracy: the accuracy of the model using Random Forest algorithm (more than 99.9%) is much higher than using SVM (98.50-98.95%) and Naïve Bayes (95.95-98.82%).

Prediction time: the prediction time of the model using Random Forest algorithm is much lower than using SVM and higher than using Naïve Bayes. However, because the accuracy of Naïve Bayes is quite low, Random Forest is still evaluated to be more effective than Naïve Bayes.

Effectiveness in different data set: With data set A, the accuracy of SVM and Naïve Bayes is quite low. With data set B, their accuracy increases significantly. Meanwhile, the accuracy of Random Forest with two data sets are high and not much difference. Thus, Random Forest can work effectively even if the steganographic packets ratio of data set is low. This is brings much practical significance because in practice, the number of steganographic packets is often much less than the number of normal packets.

Thus, through the experiment results, the authors found that when detecting storage based network steganography, Random Forest algorithm is more effective than SVM and Naïve Bayes.

5. Conclusions

Thus, in this paper, the authors presented the problem of application of machine learning technique into the process of detecting storage based network steganography. Besides, the new features which the authors proposed has been proven that bring high accuracy in practice. This is brings great practical significance in the detection of abnormal behavior of steganographic packets.

6. References

1. Wojciech Mazurczyk, Amir Houmansadr, Krzysztof Szczypiorski, Steffen Wendzel, Sebastian Zander. Information hiding in communication networks. IEEE Press; 2016. p. 44-88.
2. Mike Sieffert, Rodney Forbes, Charles Green, Leonard Popyack, Thomas Blake. Assured Information Security: Stego Intrusion Detection System. The Digital Forensic Research Conference; 2004.
3. J. Zhai, G. Liu,Y. Dai. Detection of TCP covert channel based on Markov Model. Telecommun Syst; 2013. p. 333-343.
4. Dittmann J, Hesse D, Hillert R. Steganography and steganalysis in voice-over IP scenarios: operational aspects and first experiences with a new steganalysis tool set. In: Proc SPIE, Vol 5681, Security, Steganography, and Watermarking of Multimedia Contents VII, San Jose; 2005. p. 607–618.
5. Taeshik Sohn, JungTaek Seo, and Jongsub Moon. A study on the covert channel detection of TCP/IP header using support vector machine. In Proceedings of the 5th international conference of information and community security; 2003. p. 313–324.
6. Ms. Apurva, N. Mahajan, Prof. I. R. Shaikh. Detect Covert Channels in TCP/IP Header using Naive Bayes. International Journal of Computer Science and Mobile Computing. Vol 4; 2015. p. 881-886.
7. Leo Breiman. Random Forests. Machine Learning; 2001: 45. p. 5- 32.
8. Ibrahim A Ibrahim, Tamer Khatib, Azah Mohamed, Wilfried Elmenreich. Modeling of the output current of a photovoltaic grid-connected system using random forests technique. Energy Exploration & Exploitation; 2018.
9. Alex Smola, S.V.N. Vishwanathan. Introduction to Machine Learning, Cambridge University Press; 2008.