Intrusion Detection Using a More General feature extraction method for Payload-based Anomaly One-Class Classifier

Xuan Nam Nguyen, Dai Tho Nguyen*
University of Engineering and Technology
Vietnam National University, Hanoi
* UMI 209 UMMISCO IRD/UPMC
nguyendaitho@vnu.edu.vn

Abstract—In this paper, we proposed a method to extract more general features of data for payload-based anomaly IDS. However, because of the significant rise in the number of features, there are numerous redundancies, leading to the rise in the complexity and the decrease in the accuracy of the classification. To that end, we apply Chi square [9] feature selection method to pick up the best features in the feature set. We have done many experiments on real world dataset of HTTP-based attacks to evaluate the performance of our classifier using our feature extraction method. The results show that our classifier can quickly detect the attack packets with very high true positive rate while keeping the false positive rate at a very low level. Besides, the results also indicate that our classifier outperforms other classifiers such as McPAD [10], and PAY [12, 13].

I. INTRODUCTION

A. *Intrusion Detection Systems*

In term of Intrusion Detection System (IDS), two major approaches are signature-based and anomaly-based. Signature-based IDSs detect the malicious packets by specific patterns such as byte sequences in network traffic, or instruction sequences of malware. Signature-based detectors are usually used in Anti-virus softwares, which can easily detect known types of virus or malware. However, they are unable to detect new kinds of attack, which have had no patterns yet. On the other hand, anomaly-based IDSs use the differences between the normal and attack packets to detect the malicious packets. More precisely, this kind of detector builds a model of normal packets. Any packets which are not like normal, are considered as the attack. The advantage of anomaly-based IDSs is that they can detect many kinds of attack, including zero-day attacks (unknown attacks). However, it has to suffer from high false positive, and the aim of researching about anomaly-based IDSs is to reduce the false positive rate as much as possible.

There are many proposals about anomaly detectors, and they all meet one problem: "How to extract features of data to build the best model". To solve this problem, in PAYL [12, 13], the authors proposed a payload-based extraction method, which focuses on the byte sequences of the payload in the packets. Using this method, we don't have to care about the variety of protocols or syntax of packets on the application layer. To extract the features from the byte sequences, they count the occurrence frequency of 256^n types of n-gram (n bytes consecutively). The occurrence of each type are stored in a dimension of the feature vector, so the vector has 256^n dimensions. The author did experiments with n = 2, and their detector was quite accurate, but suffering from a relatively high false positive rate. The problem is, if n is too small ($n \leq 2$) the feature vectors have insufficient information to represent the payloads, but the bigger n ($n > 2$) may lead the exponential rise in the number of dimensions, which causes the curse of dimensionality [4].

To handle this, in McPAD [10], they proposed an improvement version of n-gram feature extraction, in which they replaced n-gram by 2_v-gram (2 bytes are apart n positions from each other, v goes from 0 ... N-1). After obtaining N feature vectors, they used them to train N One-Class classifier, each one corresponded to one feature vector. The output of N One-Class classifiers were combined by a set of rules (Max, Min Average) to yield the final output, which was used to determine whether the coming packets is the attack or not. By using 2_v-gram, the authors tried to gather more information from the occurrence frequency of two-bytes values. They took into account pairs of bytes which are apart from instead of being next to each other only. However, each 2_v-gram vector is assigned to a One-Class classifier, so each classifier doesn't have enough structure information to represent the data, therefore their outputs are less accurate.

B. *Our contribution*

As the content of packets (normal messages or malicious code) are in form of languages, we can consider bytes in the payload as the characters in the text. Therefore, we conducted

a small test to determine that the occurrence frequency of two characters which their distances are various in the range contains any valued information.

The most frequent pairs of bytes of each text and pairs are v positions apart from each other. In this test, text 1, 2, 3 are pieces of text. With each piece, the most frequent pairs in all three texts are er, th, he.

TABLE I: TOP 5 MOST FREQUENT PAIRS OF BYTES

v| Text	Text 1	Text 2	Text 3
0	er, th, in, he, re	he, th, in, er, an	er, th, re, on, he
1	et, te, ae, es, ai	te, ad, et, ie, ig	te, to, et, ei, ie
2	ee, tn, rt, ei, eo	ea, eo, ee, ei, tn	ei, ee, et, tn, er
…	….	…	…
7	ee, te, et, en, ei	ee, et, te, he, eo	ee, en, et, ei, to

When v = 1, the pair et, te, ie are the most frequent. At the very long distance v = 7, there are still some pairs of bytes appearing at the top of frequency of three texts such as ee, te, et, ei. That is to say, even at a long distance (v = 7), the frequencies of two-characters still have information, and we can use them as the feature to represent for English documents. Therefore, the occurrence frequency of two bytes which are v (v is various from 0 to N-1) positions apart from each other can be used as features in feature vectors.

Therefore, it is reasonable to take into account a range of distance of two-bytes values, instead of a fix distance. For that reason, we use all vectors 2_0-gram, 2_1-gram,…, 2_{N-1}-gram in one classifier only. As the matter of fact, the feature vector obtained by the above method is $256^2 * N$ – dimensional. The numerous number of features may lead to the curse of dimensionality [4]. As the matter of fact, general feature vector needs to be reduced its number of dimensions.

In McPAD, the authors used a feature clustering algorithm proposed by Dhillon et al [3] to reduce the dimensions of feature vectors in each classifier. They distributed the features into clusters so that the number of clusters was minimum while the value of information loss was acceptable. Then, with each cluster, the average value of all features in the cluster was calculated and used as the feature of the new feature vector.

We tried to apply this algorithm, but the result was not good. When the false positive rate is 10^{-2}, the true positive is nearly 0.5. The reason for this pool result is that our feature vector has N time more features ($256^2 * N$ features) than each feature vector in classifiers of McPAD (256^2 features), thereby having more both value features and redundant features. Dividing features into clusters and using the average value of the cluster as the new feature reduces the amount of information contained in valued features, so we need to eliminate the junk features and keep the valued features only. To that end, we apply a feature selection method based on Chi-Square test, which picks up a set of best features, and we put them into the final feature vector. Our new detector using this feature selection method achieves a very high accuracy but running in a very short time. We talk about our evaluated and comparative experiments in Experiment section of this paper.

II. BACKGROUND

A. *One-Class Classification*

The anomaly detection is a two-class classification problem, with one of the classes is very under-sampled. This stems from the fact that it is very expensive and difficult to collect the malicious network activities, since most of the activities are normal [11]. In this case, one-class classification techniques, which only need the normal traffic data to train the model, can fit the anomaly detection context. In [3, 12], the one-class SVM has been shown to be highly effective in text classification problems. Moreover, the payload-based anomaly detection using n-gram frequency and the text classification problems both use the frequency vectors as the feature [8]. For that reason, we can apply one-class SVM to our problems.

B. *Chi-Square Test for feature selection*

Feature selection is a process that chooses a subset from the original feature set according to some criterions. The selected feature retains original physical meaning and provides a better understanding for the data and learning process. Depending on if the class label information is required, feature selection can be either unsupervised or supervised. For supervised methods, the correlation of each feature with the class label is computed by distance, information dependence, or consistency measures [2]. Further theoretical study based on information theory can be found on [7] and complete reviews can be found on [14, 1].

As we mentioned above, the payload anomaly detection problem using *n*-gram frequency as features is analogous to text classification. For that reason, the feature selection methods which are good at text classification can be used in our case.

There are several feature selection methods used in text classification like Information Gain (IG) [14], Chi-Square (CHI) [14, 5], Document Frequency (DF) [14], and Mutual Information (MI) [6]. In [14], the experiments indicated that IG and CHI are the most effective method. Since CHI is easier to implement, we apply this method to reduce the feature number of feature vectors.

In statistics, the Chi-Square test is applied to test the independence of two events, where two events A and B are defined to be *independent* if $P(AB) = P(A)P(B)$ or, equivalently, $P(A|B) = P(A)$ and $P(B|A) = P(B)$. In feature selection, the two events are occurrence of the term

and occurrence of the class. We then rank terms with respect to the following quantity:

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

Where $e_t = 1$ when the document D contains term t, otherwise $e_t = 0$ and $e_c = 1$ when the document D belongs to class c, otherwise $e_c = 0$. N is the *observed* frequency in D and E the *expected* frequency. For example, E_{11} is the expected frequency of t and c occurring together in a document assuming that term and class are independent.

III. OUR MORE GENERAL DETECTOR

A. *Feature extraction*

To extract the features of data, we focus on the payload of packets. In detail, the occurrence frequency of two-bytes values are calculated, and the distance of these pairs of bytes are various from 0 to N-1, instead of the fix distance as McPAD did. For example, with pair AB (the first byte is A, and the second one is B), we take into account AB, A*B, A**B,…, A(*)$^{N-1}$B. In order to do that, N vectors 2_v-gram are extracted (v from 0 – N-1), then we join them together.

For example, if one element X have 3 possible values A, B, and C, and we have a payload $AABCCABAA$, then the appearance frequency vector of 2_0-grams is as below:

AA	BB	CC	AB	BA	BC	CB	AC	CA
2	0	1	2	1	1	0	0	1

(The occurrence frequency of pair AA is 2)

The occurrence frequency vector of 2_1-grams:

A*A	B*B	C*C	A*B	B*A	B*C	C*B	A*C	C*A
1	0	0	1	1	1	1	1	1

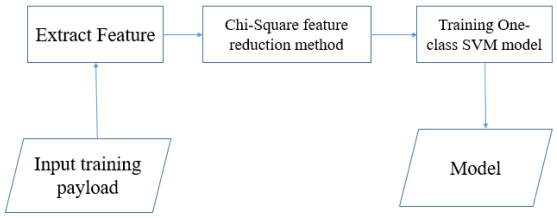
The joint of two occurrence frequency vectors:

AA	BB	CC	AB	…	B*C	C*B	A*C	C*A
2	0	1	2	…	1	1	1	1

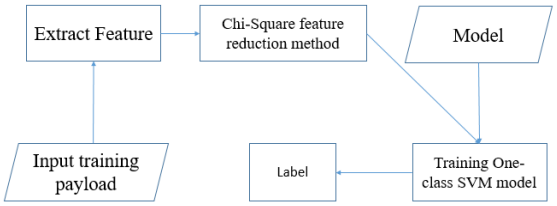
As a result, this more general method takes advantage of the amount of information in the occurrence frequency of two-bytes values. On the other hand, due to rise in the number of features, the number of redundancies also increases. To that end, in feature selection process, we calculate the χ^2 value of each feature, then K features having the largest χ^2 values are chosen to put into the final feature vector.

B. *The working flow of our detector*

Our detector‘s working process consists of two phases: Training phase and detecting phase. Firstly, in the training phase, the input is the payload of normal packets, and the detector extracts feature vectors of them as the process above. Then, feature vectors are used to train One-Class SVM model. The detector uses the model below to classify future payloads.



Secondly, in the detecting phase, the input is also the payload of packets. After the feature extraction process, the detector uses the model obtained in the training phase to classify feature vectors of payloads and determine which payloads are malicious.



C. *Experiments*

1) *Implementation*

We extend the McPAD open-source program[1] to implement our new detector. The following modules are our contributions. The first module is to extract data by the way we described in section III.A. The last one is Chi-square feature selection module, which is used to choose the best features and eliminate the redundancies. In the experiment, the value of N (v goes from 0 to N-1) is 10, because the number of classifiers used in the experiment of McPAD is 10. In addition, in McPAD, the number of features decreased from 256[2] to 160, so we let K (number of features after feature selection process) is also 160.

2) *Validation Metrics*

We use two metrics in performance evaluation, including the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). The ROC curve is a graph indicating the tradeoff between false and true positive rate [8]. For a classifier, the area under this curve shows how accurate this classifier can detect the malicious payload. As a result, we use AUC to record this value. However, it is unrealistic to use a classifier with the false positive rate is over 10%. Therefore, we only take into account the AUC values when the false positive rate is in the range of [0, 0.1].

[1] http://roberto.perdisci.googlepages.com/mcpad

3) Datasets

In our experiences, we use two datasets. The normal traffic dataset contains the network activities of the first week of DARPA'99 dataset[2]. This dataset is used to train the one-class SVM model and evaluate the false positive rate. On the other hand, the malicious traffic dataset is taken from McPAD website, divided into 3 following subsets:

- Generic Attacks consists of 66 HTTP attacks[3]. Among these, 11 are categorized as shell-code attacks that carry executable code in the payload. The remaining attack categories include Failure to handle exceptional conditions, File disclosure, Information leak, Input validation error, Poor memory management, Poor resource management, Signed interpretation of unsigned value, URL decoding error.
- Shell-code Attacks contains the 11 shell-code attacks from the subset above.
- CLET Attacks contains 96 polymorphic attacks generated using the polymorphic engine CLET [11].

4) Experiment result

a) Validation of our detector

Table II shows that the AUC value of our detector is very close to 1, that means it is very good and can be used in practice.

TABLE II: AUC VALUE OF DETECTORS

Type of Attack	AUC
Generic Attacks	0.97227
Shell-code	0.98656
CLET	0.96737

b) Comparing with PAYL and McPAD

In this section, we compare the performance of our detector with those of PAYL [13] and McPAD [10]. Specifically, we use 4 types of attacks: Generic, Shell-code, CLET

.

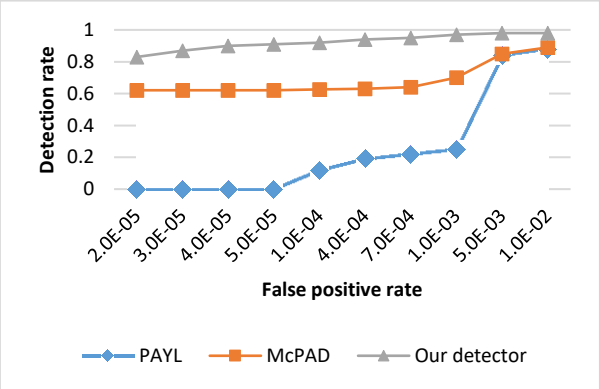


Fig. 1. Generic Attacks

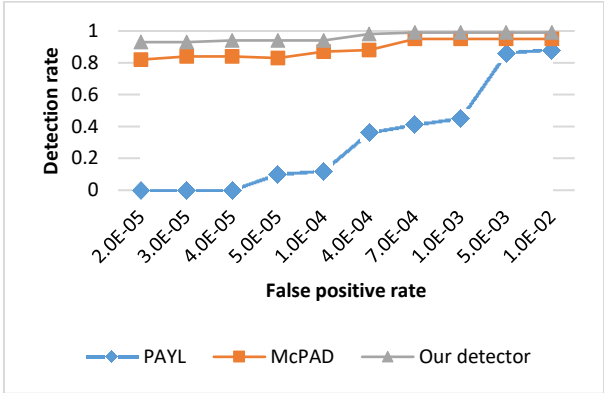


Fig. 2. Shell-code Attacks

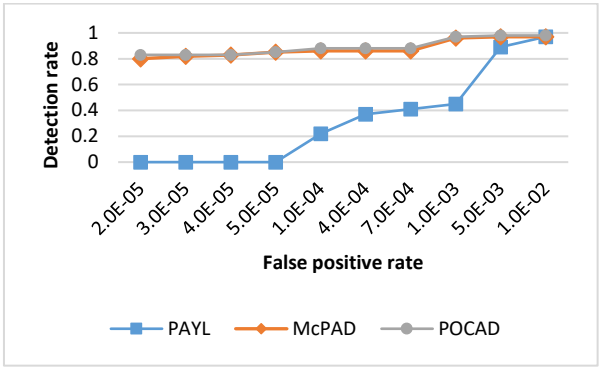


Fig. 3. CLET Attacks

Figure 1, 2, 3 indicate that with Generic, Shellcode and CLET attacks, PAYL has a very low true positive rate (or detection rate) when its false positive rate stays at very low level (less than $5^* 10^{-3}$). With Generic attacks, McPAD has a quite high true positive rate at that level of false positive rate, and our detector outperforms it in the whole range of false positive rate. For example, when the false positive rate is at $2 * 10^{-5}$, the true positive of McPAD is 0.6, substantially less than our detector, 0.8. Next, when the false positive rate is 10^{-2} , while the true positive of ours is nearly 1, this figure of McPAD is just above 0.9. Besides, with Shellcode and CLET attacks, our detector's performance is slightly better than McPAD, since both detectors are very good at detecting these kinds of attacks.

TABLE II: AVERAGE TIME PER PAYLOAD

Detector	AVG processing time (ms)
PAYL	0.039
McPAD	13.39
Our detector	2.06

[2] https://www.ll.mit.edu/ideval/data/1999/training/week1/index.html

[3] http://roberto.perdisci.com/publications/publication-files/HTTP_generic-attacks.zip?attredirects=0

Table II compares the average time consuming when detectors process a payload. Although PAYL runs very fast, it suffers from the relatively high false positive rate as shown in charts above. Interestingly, our detector works much 6 times faster than McPAD, but significantly more accurate, because McPAD has to run many classifiers at the same time, and our detector has only one classifier.

IV. CONCLUSION

We propose a novel, more general method to extract features of data for payload-based anomaly IDS, consisting of an improved 2_v-gram feature extraction method and a feature selection method based on Chi-Square test. Our method takes into account the various values of distances between two bytes, so that more information may be obtained from the occurrence frequency of two-bytes values. Then, we apply this method to build a new detector using One-Class SVM classifier to effectively detect network intrusion attacks. Our extensive performance evaluation of this detector shows that it can quickly detect different types of HTTP-based attacks and consistently achieves a high detection rate as well as a very low false positive rate. Our detector also outperforms state of the art payload-based detection schemes such as McPAD [10] and PAYL [13]. We thus believe this detector can be useful for payload-based intrusion detection in practice.

REFERENCE

[1] Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial Intelligence, 97, 245-271.

[2] Dash and Liu. Feature selection for classification. Intelligent Data Analysis Volume 1, Issues 1–4, 1997, Pages 131-156

[3] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research, 3:1265–1287, 2003.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification. Wiley, 2000

[5] Ogura H, Amano H, Kondo M. Feature selection with a measure of deviations from Poisson in text categorization. Expert Systems with Applications. 2009;36(3):6826–6832.

[6] Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005;27(8):1226–1238.

[7] D. Koller and M. Sahami (1996). "Toward Optimal Feature Selection." Proceedings of the Thirteenth International Conference on Machine Learning (ICML) (pp. 284-292).

[8] E. Leopold and J. Kindermann. Text categorization with support vector machines. How to represent texts in input space? Machine Learning, 46:423–444, 2002.

[9] H. Liu and H. Motoda, editors. Computational Methods of Feature Selection. Chapman and Hall/CRC Press, 2007.

[10] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, W. Lee. "McPAD : A Multiple Classifier System for Accurate Payload-based Anomaly Detection." Computer Networks, Special Issue on Traffic Classification and Its Applications to Modern Networks, 5(6), 2009, pp. 864-881.

[11] D. M. J. Tax. One-Class Classification, Concept Learning in the Absence of Counter Examples. PhD thesis, Delft University of Technology, Delft, Netherland, 2001.

[12] K. Wang and S. Stolfo. Anomalous payload-based worm detection and signature generation. In Recent Advances in Intrusion Detection (RAID), 2005.

[13] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In Recent Advances in Intrusion Detection (RAID), 2004.

[14] Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. Proceedings of the 14th International Conference on Machine Learning (ICML '97); 1997; Nashville, Tenn, USA. Morgan Kaufmann; pp. 412–420.