Multidimensional and Hierarchical Anomaly Detection System of Web Attacks

Jiawei Li, Jianfeng Guan*, and Zhongbai Jiang
Beijing University of Posts Telecommunications, Beijing, 100876, China
ljwemls@gmail.com, {jfguan, zbjiang}@bupt.edu.cn

Abstract

In recent years, the large-scale dynamic HTTP requests have raised great challenges for traditional detection system in web applications. In general, intrusion detection is classified into misuse detection and anomaly detection. Misuse detection system has its own disadvantages in poor adaptability and high-cost of renewal and maintenance. Therefore, anomaly detection system has emerged as the improvement of misuse detection which can identify previously unknown attacks. However, learning-based anomaly detection system is prone to result in high false positives. This paper presents a hierarchical anomaly detection system that combines multidimensional feature generating system and classification system. The whole system is divided into three steps: firstly, construct a separate statistical model based on large quantities of HTTP access records; secondly, adopt unsupervised learning algorithms to build a variety of detecting subsystems; finally, merge the results of every subsystem by classification algorithm. We have evaluated this system by one month real web logs of *QIHU*360. The results demonstrate that the proposed model has a good detection performance and time complexity.

Keywords: Web Attacks, Anomaly Detection, Statistical Model, Classification

1 Introduction

With the rapid development of the Internet, web applications are becoming richer and richer, but at the same time, web-based attacks also become more and more complex. So anomaly detection has long become the central study subject of web security.

Misuse detection system constructs models to the already known attacks and has a good detection performance on all kinds of known attacks. But the main limitations lie in the following:

(1) Poor adaptability: the judgment criteria of similar regular expressions are easily evaded, and the system cannot effectively detect unknown types of attack.

(2) High threshold: signature-based detecting method depends largely on the experience and judgment of security researchers. It has a high demand for the expertise level of developers, therefore, causing a high study threshold.

(3) High cost: rulebase needs to be regular update and maintenance by experts, while every modification of online system costs a lot.

Intrusion detection system relies on normal behavior of users and applications to identify anomalous activity. When the present behavior is detected different, the system is considered to suffer a certain attack. If the anomaly detection model has good adaptation, it can detect several kinds of malicious attacks and has the ability of detecting some unknown intrusion patterns. However, learning-based intrusion systems are prone to producing a large number of false positives that it isn't as precise as misuse detection system. In [5], Bayesian Hidden Markov Model is proposed to compose models and express inter-model

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 2, Article No. 9 (September 15, 2016)

*Corresponding authors: Institute of Network Technology, Beijing University of Posts Telecommunications, 10 Xitucheng Road, Haidian, Beijing, 100876, China, Tel: +86-186-1121-1658

dependencies in order to further enhance performance and availability. By improving $k-means$ algorithm, a system for clustering anomalies and classifying clusters is proposed in [12] to detect intrusion behaviors including system or users' non-normal behavior and unauthorized use of computer resources. These kinds of systems can only report that there is an attacks without any supporting description of the anomaly that has been detected. Juvonen and Hamalaine[10] considers a log anomaly detection framework which is based on dimensionality reduction by using random projection and uses Mahalanobis distance to find outliers. Ippoliti[9] introduces a support vector machine based on adaptive anomaly detection and correlation mechanism for flow analysis. Application of growing hierarchical self-organizing map to analyze the network traffic data and visualize the distribution of attack patterns is investigated in [8]. Study[14][7] try to use the Shannon entropy analysis to detect these attacks, which assumes that attacks should have more complexity than common access requests. Unfortunately, the system detects the overall anomaly which results in a relatively sophisticated detection algorithm and therefore, is hard for online detection. Baddar [1] presents several categorizations of anomaly detection solutions which have adopted the machine learning approach. SVM, simulated annealing and decision trees were used to select features and obtain decision rules to help identify new attacks in [13]. Barani [2] proposed GAAIS algorithm based on genetic algorithm (GA) and artificial immune system (AIS) for dynamic intrusion detection in ad hoc networks. In [6], Discriminative Restricted Botlzman Machine (DRBM) is used to develop a self-learning anomaly detection approach which combines strong generative modeling with classification accuracy in the network's traffic.

This paper proposes a comprehensive detection system against Web anomaly. It combines multi-dimensional feature generating system and hierarchical classification system, which makes a comprehensive assessment to the web-based access behavior. Whole system is divided into three steps: firstly, construct a separate statistical model which is based on large quantities of HTTP access records under each specific domain; secondly, construct a number of detecting subsystems based on statistical characteristics and unsupervised learning algorithms; finally, remap the results of every detecting subsystem to the new feature space and use classification algorithm for model merging.

This paper is structured as follows. Section 2 presents the architecture for whole system and briefly describes its main subsystems with related algorithms. Section 3 provides an experimental evaluation of every subsystem and integrated detection model with respect to real-world corporate dataset. Finally, Section 4 draws conclusions and outlines future work.

2 Whole System

The whole anomaly detection system is mainly consist of four modules: data preprocessing, construction of statistical characteristics, multidimensional subsystems and detection module. The system flow chart is shown in Figure 1:

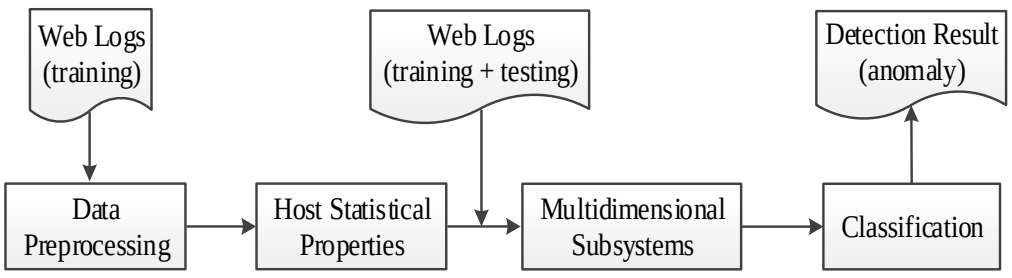

Figure 1: System Flow Chart

2.1 Data Preprocessing

The input of data preprocessing module is Web log (parsed HTTP packets). A complete access record at least includes the field of time stamp, source IP, source port, destination IP, destination port, URI and host domain, etc. The module mainly processes error, absent or repetitive data, and classifies log records according to different host domains.

2.2 Construction of Statistical Characteristics

Calculate all the related statistical parameters according to the log records under a certain domain output by data preprocessing module, and construct a statistical model under specific domains. The following four categories are included:

- **HPFL** (Hierarchical Path Frequency List): count the frequency of every hierarchical path and the two neighboring hierarchical path according to the path field of each URI ;
- **BPS** (Bag of Parameter Subset): First, as for every record under the same path, extract all the parameters $(p_1, p_2, ..., p_k)$ that appear in the parameter field and form a subset of parameter $S_i = \{\{p_1, p_2, ..., p_k\}\}$. Then gather the different parameter subsets and form a BPS under a certain path, that is $S = S_1 \cup S_2 \cup ... \cup S_n$ (n is the record number under a specific path);
- **PRS** (Parameter Rule Set): Firstly, extract parameters and compose a directed graph according to each ordered sequence from every URI under the same path. Secondly, traverse all the parameters (nodes) and calculate the links between the nodes to make sure whether they are connected. Finally, construct PRS (S). Take parameter x and y for example:

  (1) If node x can reach node y, but node y cannot reach node x, it means node x is surely before node y, then add side (y, x) to S.

  (2) If node y can reach node x, but node x cannot reach node y, it means node y is surely before node x, then add side (x, y) to S.

  (3) All the other conditions cannot conclude the fixed sequence, nor can they change the rule set S.
- **EPS** (Enumeration Parameter Set): enumeration parameter refers to that parameter value under certain variables all come from a fixed finite EPS[11], such as content identification or index. We firstly introduce two auxiliary functions $f(x)$ and $g(x)$ as shown in Equation 1 and 2.

$$f(k) = k \qquad k = 1, 2, 3...n \tag{1}$$

$$g(k) = \begin{cases} 0 & , k = 0 \\ g(k-1)+1 & , q_k \text{ does not occur in the first k} - 1 \text{ parameter values} \\ g(k-1)-1 & , q_k \text{ has occured in the first k} - 1 \text{ parameter values} \end{cases} \tag{2}$$

Suppose a certain parameter q under a path has n values $(q_1, q_2, ..., q_n)$. Then $f(x)$ and $g(x)$ are concluded from the actual data, then calculate the correlation coefficient ρ of the two functions. We can judge whether a certain variable name belongs to enumeration parameter set by using the following rules:

(1) If $\rho < 0$ (negative correlation), variable name q is enumeration;

(2) If $\rho > 0$ (positive correlation), variable name q is stochastic;

(3) If $\rho \approx 0$, it is hard to judge the type of variable name q. To avoid false positive, it can be dealt as stochastic type.

EPS contains all parameters that are judged as enumeration.

2.3 Multidimensional Subsystems

Detect the behavioral characteristics from different dimensions and form new feature vectors for final classification. Every subsystem will detect the certain dimension of a single record, generating a probability that the record can be regarded as a normal behavior. Whole multidimensional module is consist of three parts and the construction is shown in Figure 2.

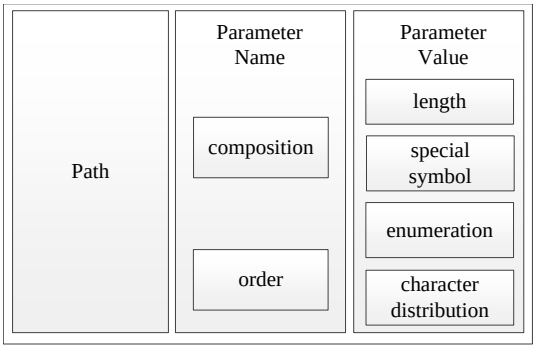

Figure 2: Construction of Subsystems

2.3.1 Path

Calculate the normal probability of the path in URI by using Ngrams[3] (we choose Bigram here). Consider a specific path: $a/b/c?x = 1\&y = 2$. As for $P(b|a)$, according to the conditional probability and maximum likelihood estimation:

$$P(b|a) = \frac{P(a,b)}{P(a)} = \frac{Count(a,b)}{Count(a)} \tag{3}$$

Among it, $Count(a,b)$ represents the frequency that path a appear before path b. $Count(a)$ stands for the frequency that Path a occur. The specific number can be acquired from HPFL in previous module. As for the imbalance caused by different lengths of paths, we can adopt the related regularization term to revise it. The example is shown in Table 1.

Table 1: The process of calculating path probability

URI	$a/b/c?x = 1\&y = 2$
path	$/a/b/c$
path depth	3
add location identifier	$HEAD/a/b/c/END$
normal probability	$\sqrt[3]{\log[p(a\|HEAD) * P(b\|a) * P(c\|b) * P(END\|c)]}$

2.3.2 Variable Correlation

In most cases, users don't manually input all parameters of URI in the browser but click the page links or buttons to open a web page. The URI generated from those links or buttons is pre-defined by client programs, scripts or HTML forms. This mechanism make the parameters have a strong regularity in composition, number and order, etc. Therefore, the conditions of the excess, lack, disorder of the relevant parameters should be regarded as abnormal.

- **Variable Composition**

  If the variable composition extracted from a record's URI is not in its corresponding BPS, it is judged to be anomaly, output 0. Otherwise, output 1;

- **Variable Order**

  If the corresponding PRS contains the order of the record's variable, indicating that the variable sequence is abnormal, output 0. Otherwise, output 1.

2.3.3 Parameter Value

Normal query parameter values are automatically generated by web scripts or users' input through HTML forms. The string length of these parameters is either fixed or changed in a small range. However, many malicious attacks contain parameter values that are often far beyond the normal range. For example, buffer overflow attack often contains malicious binary code with hundreds of bytes, while XSS attack will be embedded with a large number of invasive scripts. Meanwhile, under normal conditions, the character distribution of the specific parameter values has a regular structure. As for anomaly records, there will be a completely different character distribution, for example some exploratory behaviors before the attacks will contain a large number of "." symbols so as to achieve traverse routing space.

This module is consist of four parts: length distribution, special symbols, enumeration and character distribution.

- **Length Distribution**

  The revised Chebyshev inequality is used to calculate the normal probability of the length distribution. Assuming the variable x_i whose means of the value length distribution is μ_i, and standard deviation is σ_i. If a record whose corresponding parameter value length is l_i, if $l_i \leq \mu_i$, then output 1. Otherwise, as shown in Equation 4, calculate the normal probability $P(x_i)$ with distribution of variable x_i as output:

$$P(x_i) = \frac{1}{1+\frac{1}{\frac{\sigma^2}{2\varepsilon_i{}^2}}} = \frac{1}{1+2\frac{\varepsilon_i{}^2}{\sigma^2}} \qquad ,\varepsilon_i = l_i - \mu \tag{4}$$

- **Special Symbols**

  The special symbol sets: $S = \{$"!", "@", "#", "%", "$*$", "(", ")", "{", "}"$\ldots\}$. The probability of all special symbols contained in parameter value of the whole model space is used as its normal probability. For a specific record, get each normal probability of the special symbol contained in the parameter value and take its minimum as variable's normal probability. Then take the minimum normal probability of every different variable in the record as the output of the module.

- **Enumeration**

  For a specific record, if a variable is judged as enumeration parameter and not belong to the relevant EPS, this behavior is then asserted as anomaly and output 0. Otherwise, the subsystem output 1.

- **Character Distribution**

  The character distribution contains direct character distribution test[11] and aggregated character distribution test. The former is based on the single ASCII code for interval division as shown in Table 2, and the latter is based on the character categories (uppercase letters, lowercase letters, control character, digit, unprintable characters, out-of-range characters) as shown in Table 3. Combined with the probability expectation among specific ranges of characters, the Chi-square test is used to respectively calculate the normal probability of character distribution under those two conditions.

Table 2: Direct character distribution test

Interval number (position of ordered probability sequence)	1 (0)	2 (1-3)	3 (4-6)	4 (7-11)	5 (12-15)	6 (16-255)
x	x_1	x_2	x_3	x_4	x_5	x_6
y	y_1	y_2	y_3	y_4	y_5	y_6
...	...	...	...	...	...	...

Table 3: Aggregated character distribution test

Character type	uppercase letter	lowercase letter	control character	digit	unprintable character	out-of-range character
x	x_1	x_2	x_3	x_4	x_5	x_6
y	y_1	y_2	y_3	y_4	y_5	y_6
...	...	...	...	...	...	...

x_i and y_i respectively represent the probability expectations that the frequency of characters in a given interval $(\sum_i x_i = 1, \sum_i y_i = 1)$.

Take the direct character distribution test to illustrate the calculation process of a specific record: assuming the URI is "$a/b?x = 123@mm\&y = nn\#1\#2$". The parameter value of variable x is $123@mm$, the frequency of the ASCII code is shown in Table 4:

Table 4: ASCII Code frequency of variable x

ASCII Code	0	1	...	49	50	51	...	64	...	109	...	255
Frequency	0	0	...	1	1	1	...	1	...	2	...	...

Without taking the character value into account, the frequency is rearranged by the order from big to small, then the ordered frequency $(2,1,1,1,1,0,0,...,0)$ is acquired. The sequence includes 256 elements, and the corresponding index is $(0,1,2,...,255)$. Then according to the partition of Table 2, sum up the frequency of the sequence in the corresponding interval to get the following interval distribution as shown in Table 5:

Table 5: Character distribution of variable x

Interval number (position of ordered probability sequence)	1 (0)	2 (1-3)	3 (4-6)	4 (7-11)	5 (12-15)	6 (16-255)
Interval Frequency	2	3	1	0	0	0

The interval frequency distribution are tested with Chi-square (DOF is 5). $P({\chi_x}^2|5)$ is the normal probability for the character distribution corresponding to the variable x. The calculation process

of variable y is similar to the above, eventually the normal probability of the record is the minimum of the two, that is $\min(P(\chi_x^2|5), P(\chi_y^2|5))$.

- **Generation of Feature Vector**

  According to the detection result of each subsystem, each record is mapped into an 8 dimensional feature vectors as shown in Table 6, which is used as the input features of detection model.

Table 6: An instance of feature vector

Feature	path	variable composition	variable order	value length	special symbols	enum-eration	value ditribution	value distribution2
$x_{8\times1}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(6)}$	$x^{(7)}$	$x^{(8)}$

$x^{(1)}, x^{(4)}, x^{(5)}, x^{(7)}, x^{(8)}$ are the floats from zero to one, $x^{(2)}, x^{(3)}, x^{(6)}$ are boolean parameter which is zero or one.

2.4 Detection model

Based on the multidimensional feature vector above, classification algorithm (Decision Tree, SGD, SVM, etc.) was used to detect anomaly on web logs. The detection flow chart is shown in Figure 3:

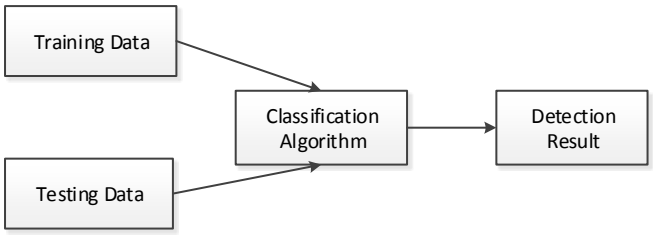

Figure 3: Detection Flow Chart

The input matrix of training data is shown as Equation 5:

$$\begin{bmatrix} x_1^{(1)}, x_1^{(2)}, x_1^{(3)}, x_1^{(4)}, x_1^{(5)}, x_1^{(6)}, x_1^{(7)}, x_1^{(8)}, y_1 \\ x_2^{(1)}, x_2^{(2)}, x_2^{(3)}, x_2^{(4)}, x_2^{(5)}, x_2^{(6)}, x_2^{(7)}, x_2^{(8)}, y_2 \\ \cdots \\ x_m^{(1)}, x_m^{(2)}, x_m^{(3)}, x_m^{(4)}, x_m^{(5)}, x_m^{(6)}, x_m^{(7)}, x_m^{(8)}, y_m \end{bmatrix}_{m\times9} \tag{5}$$

m represents the total record number of training dataset. $x_a^{(b)}$ represents the b^{th} $(1 \leq b \leq 8)$ dimension of a^{th} record in the training dataset. y_k represents the real label of k^{th} record (-1 represents anomaly, +1 represents normal record).

The input matrix of testing data is shown as Equation 6:

$$\begin{bmatrix} \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)} \\ \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)} \\ \cdots \\ \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)} \end{bmatrix}_{n\times8} \tag{6}$$

n represents the total record number of testing dataset. $\hat{x}_a^{(b)}$ represents the b^{th} $(1 \leq b \leq 8)$ dimension of a^{th} record in the testing dataset.

The output matrix of detection model is shown as Equation 7:

$$\begin{bmatrix} \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{x}_1^{(1)}, \hat{y}_1 \\ \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{x}_2^{(1)}, \hat{y}_2 \\ \ldots \\ \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)}, \hat{x}_n^{(1)} \end{bmatrix}_{n \times 9} \tag{7}$$

n and $\hat{x}_a^{(b)}$ represent the same meaning in Equation 6. Alternatively, y_k represents the predicted label of k^{th} record (-1 represents anomaly, +1 represents normal record).

3 EVALUATION

3.1 Dataset

The dataset adopted in this paper is a one-month web logs (parsed HTTP packets) under a certain domain (*xiaoshuo*.360.*cn*) in company *QIHU*360. There are 883355 access records, with the proportion of normal records to anomaly being 8:1. Each access record is labeled as either normal, or as an attack, with exactly one specific attack type. Attacks fall into six main categories: XSS(Cross Site Scripting), SQL Injection, XML Injection, CRLF Injection, LFI(Local File Include) and Directory Traversal. Due to the intercompany scanners used for emulating attacks, the input dataset contains various typical attack types and the total anomaly proportion is larger than conventional access logs.

The first 20% of normal records (according to the time order) are adopted to construct statistic models, while the rest of the whole records are split medially into training and testing data for the detection model. The detail is shown in Table 7.

Table 7: The record number of input dataset

	Statistic model	Training data	Testing data
Normal record (768187)	157237	314475	314475
Abnormal record	0	48584	48584

3.2 Single Model

Based on Decision Tree, we use the output of each subsystem as a separate classifier to detect anomaly records. The corresponding detection performance of the single model is shown in Table 8.

Table 8: The detection performance of each single model

	Path	Variable composition	Variable order	Value length
Precision	0.87065	0.79964	0.74570	0.78180
Recall	0.88890	0.75662	0.83323	0.79723
	Special Symbols	Enumeration	Value distribution	Value distribution2
Precision	0.84426	0.74407	0.87101	0.84947
Recall	0.89971	0.80023	0.89908	0.89986

It can be observed from the above table that *path*, *special symbols* and two kinds of *parameter value distribution* features achieve an good performance in the single model. Obviously one important reason is that general injection attacks need to ship the shell code (HTML, scripts, Linux commands, etc) and additional padding into the fields of HTML form which may well contain variety of special symbols. Xss scripting attempts require a substantial amount of characters, thereby increasing the abnormal parameter value noticeably. Directory traversal attempts stand out because of the unusual structure of the repetitions of slashes and dots, which can be easily detected by the feature - *path* or *value distribution*. Above all, these three features have strong separate classification ability in our verification environment.

However, with the supreme precision and recall being 0.871 and 0.899, it is hard to meet the practical requirements of an corporate-level detection system.

3.3 Integrated Model

With the multidimensional feature vector as shown in Equation 7, different machine learning algorithms are selected as classification models. The corresponding detection performance is shown in the Table 9:

Table 9: The detection performance of different algorithms

	Decision Tree	SGD	SVM	Random Forest	Xgboost
Precision	0.92358	0.96750	0.98524	0.98750	0.99988
Recall	0.94156	0.95730	0.95114	0.99455	0.99640
Cost time(s)	128	460	9434	1497	697

It can be observed from the above table that the detection performance of any integrated model is much higher than that of the single model. The precision and recall of Decision Tree, the simplest classification model is higher than 0.92. Compared with five kinds of prediction algorithms, Xgboost[4] possesses the optimal detection performance, with merely about 0.0001 false positive (FP) rate. In order to avoid overfitting, Xgboost does a second order Tailor expansion on object function and adds an extra regularization term to balance the decrease of loss function and model complexity. Besides robust, its time complexity is far below the SVM and it is better than another tree model, Random Forest.

4 Conclusion

To deal with the limitation of misuse detection system and traditional intrusion system, this paper has represented a comprehensive detection system, combining feature generating with prediction algorithms to achieve a integrated assessment on behavioral features of the web access records. Especially, using Xgboost as a merging model, the detection precision and recall will reach 0.9998 and 0.9964 that are really remarkable. Additionally, the whole module has an ideal time complexity, which is sufficient for applying this mechanism in corporate-level detection system.

In the future, we plan to continue the use of hierarchical approaches to detect web-based intrusions. We are going to concentrate on the improvement of the detection performance and adding self-adaption threshold to automatic update domain's statistical characteristics.

Acknowledgments

This work was partially supported by the National Basic Research Program of China (973 Program) under Grant No. 2013CB329102, in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61232017, 61372112 and 61003283.

We would like to thank *QIHU*360 corporation which made it possible to test our detection system on abundant labeled web logs from various web sites.

References

[1] S. A.-H. Baddar, A. Merlo, and M. Migliardi. Anomaly detection in computer networks: A state-of-the-art review. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 5(4):29–64, December 2014.

[2] F. Barani. A hybrid approach for dynamic intrusion detection in ad hoc networks using genetic algorithm and artifical immune system. In *Proc. of the 2014 Iranian Conference on Intelligent Systems (ICIS'14), Bam, Iran*, pages 1–6. IEEE, February 2014.

[3] T. Brants and A. C. Popat. Large language models in machine translation. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07), Prague, Czech Republic*, pages 858–867. Association for Computational Linguistics, June 2007.

[4] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. `http://arxiv.org/abs/1603.02754`, Mar 2016.

[5] E. Dorj and C. Chen. A bayesian hidden markov model-based approach for anomaly detection in electronic systems. In *Proc. of the 2013 IEEE Aerospace Conference (IAC'13), Big Sky, Montana, USA*, pages 1–10. IEEE, March 2013.

[6] U. Fiore and F. Palmieri. Network anomaly detection with the restricted boltzmann machine. *Neurocomputing*, 122:13–23, December 2013.

[7] S. K. Gautam and H. Om. Anomaly detection system using entropy based technique. In *Proc. of the 1st International Conference on Next Generation Computing Technologies (NGCT'15), Dehradun, India*, pages 738–743. IEEE, September 2015.

[8] S.-Y. Huang and Y.-N. Huang. Network traffic anomaly detection based on growing hierarchical som. In *Proc. of the 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'13), Budapest, Hungary*, pages 1–2. IEEE, June 2013.

[9] D. Ippoliti and X. Zhou. Online adaptive anomaly detection for augmented network flows. In *Proc. of the 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS'14), Paris, France*, pages 9–11. IEEE, September 2014.

[10] A. Juvonen and T. Hamalainen. An efficient network log anomaly detection system using random projection dimensionality reduction. In *Proc. of the 6th International Conference on New Technologies, Mobility and Security (NTMS'14), Dubai, UAE*, pages 1–5. IEEE, March 2014.

[11] C. Kruegel and G. Vigna. Anomaly detection of web-based attacks. In *Proc. of the 10th ACM conference on Computer and communications security (CCS'03), Washington, DC, USA*, pages 251–261. ACM Press, October 2003.

[12] H. Li and Q. Wu. Research of clustering algorithm based on information entropy and frequency sensitive discrepancy metric in anomaly detection. In *Proc. of the International Conference on Information Science and Cloud Computing Companion (ISCC-C'13), Guangzhou, China*, pages 799–805. IEEE, December 2013.

[13] S.-W. Lin and K.-C. Ying. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. *Applied Soft Computing*, 12(10):3285–3290, October 2012.

[14] L. Zhao and F. Wang. An efficient entropy-based network anomaly detection method using mib. In *Proc. of the 2014 International Conference on Progress in Informatics and Computing (PIC'14), Shanghai, China*, pages 428–432. IEEE, May 2014.

Author Biography

Jiawei Li received his B.S. degree in Telecommunications Engineering from the Beijing University of Posts Telecommunications(BUPT) of China in June 2014. He is currently working toward the M.S. degree in the Institute of Network Technology at BUPT. His research interests include data mining, machine earning and network security.

Jiafeng Guan received his B.S. degree from Northeastern University of China in July 2004, and received the Ph.D. degree in communications and information system from the Beijing Jiaotong University in Jan. 2010. He is an associate professor in the Institute of Network Technology at Beijing University of Posts and Telecommunications (BUPT). His main research interests focus around Mobile Internet, network security and future network.

Zhongbai Jiang received her B.S Degree in Zhengzhou University, China, in 2014. He is currently working toward a Ph.D. degree at the Institute of Network Technology, Beijing University of Posts and Telecommunications, China. His current research interests are in the areas of high-speed railway networks, wireless communications and wireless networking.