Intrusion/Anomaly Detection and Malware Mitigation

An Effective Intrusion Detection Model Based on Pls-Logistic Regression with Feature Augmentation

Jie Gu[1,2](✉)

[1] Postdoctoral Research Station, Agricultural Bank of China, Beijing 100005, China
gujie@pku.edu.cn
[2] School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

Abstract. Computer network is playing a significantly important role in our society, including commerce, communication, consumption and entertainment. Therefore, network security has become increasingly important. Intrusion detection systems have received considerable attention, which not only can detect known attacks or intrusions, but also can detect unknown attacks. Among the various methods applied to intrusion detection, logistic regression is the most widely used, which can achieve good performances and have good interpretability at the same time. However, intrusion detection systems usually confront with data of large scale and high dimension. How to reduce the dimension and improve the data quality is significant to improve the detection performances. Therefore, in this paper, we propose an effective intrusion detection model based on pls-logistic regression with feature augmentation. More specifically, the feature augmentation technique is implemented on the original features with goal of obtaining high-qualified training data; and then, pls-logistic regression is applied on the newly transformed data to perform dimension reduction and detection model building. The NSL-KDD dataset is used to evaluate the proposed method, and the empirical results show that our proposed method can achieve good performances in terms of accuracy, detection rate and false alarm rate.

Keywords: Feature augmentation · Intrusion detection · Logistic regression · Partial least square · Network security

1 Introduction

With the rapid development of internet, networks are becoming more and more important in our daily life. Organizations rely heavily on networks to do on-line transactions, and also, individuals are dependent on networks to work, study and entertain. In a word, networks are an essentially indispensable part in modern society. However, this over-dependence on networks might have potential risk, because considerable information that relates to organization operation and individual activities is accumulated and stored. It would cause huge losses, when the networks are been invaded or attacked.

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 133–140, 2020.
https://doi.org/10.1007/978-981-33-4922-3_10

Intrusion detection systems are the most widely used tool to protect information from being compromised. Intrusion detection has been long considered as a classification problem [1, 2]. Various statistic-based and machine-learning-based methods have been applied to improve the performances of intrusion detection systems [3, 4]. However, machine learning-based methods for intrusion detection suffer criticisms [5]. Though many machine- learning-based detection methods, such as support vector (SVM) machine and artificial neural network (ANN), could achieve better detection performances, the detailed procedures of the detection process remain unknown. It is called the black-box which is not favorable for practical applications. Moreover, machine-learning-based detection methods are common time-consuming. For example, the training complexity of SVM cannot be tolerable when confront with large-scale and high dimension dataset. However, the statistic-based detection methods could cover these shortages to a large extent in terms of the model interpretation and training speed. Therefore, it can be inferred that when compared to machine-learning-based intrusion detection approaches, statistic-based intrusion detection method have some advantages, that is, good interpretability and fast training speed.

Among these statistic-based detection methods, logistic regression is the most widely used classification approach, which could achieve good detection performances [6–8]. It is worthy to noting that logistic regression could model the correlations among feature and take into account of the joint effects between features to produce a decision boundary to separate different classes effectively. Therefore, logistic regression can be considered as an effective detection method. However, we should also realize that to achieve further improvement in detection performance, it may not be sufficient to use logistic regression alone. Review of related work in intrusion detection indicates that data quality data quality has been considered as a critical determinant [9].

Therefore, in our study, we propose an effective intrusion detection framework based on pls-logistic regression with feature augmentation. Specifically, the feature augmentation technique is used to improve the data quality, and pls-logistic regression is chosen to reduce the dimension and build the intrusion detection model using the transformed data. The reminder of this paper is organized as follows. In Sect. 2, we give a brief overview of feature augmentation and pls-logistic regression. Section 3 describes the details of the proposed intrusion detection model. Section 4 presents the experiment settings, results and discussions. Finally, Sect. 5 comes to conclusion.

2 Methodology

To better illustrate the proposed detection model, firstly, we briefly review the main principles of the feature augmentation [10] in Sect. 2.1, as well as the pls-logistic regression classification model [11] in Sect. 2.2.

2.1 Feature Augmentation

Following Fan et al. (2016), suppose we have a pair of random variables $(\mathbf{X}, Y)$ with n observations, where $\mathbf{X} \in \mathbb{R}^p$ denotes the original features and $Y \in \{0, 1\}$ denotes the corresponding binary response. The logarithm marginal density ratio transformation is used as the feature augmentation technique to transform the original features. Specifically, for $X_j, j = 1, 2, \ldots, p$ in $\mathbf{X}$, denote by f_j, g_j the class conditional densities, respectively, for class 1 and class 0, that is, $(X_j|Y = 1) \sim f_j$ and $(X_j|Y = 0) \sim g_j$. Denote by ${}^1X_j = \{X_{ij}|Y_i = 1, i = 1, 2, \ldots, n\}$ and ${}^0X_j = \{X_{ij}|Y_i = 0, i = 1, 2, \ldots, n\}$. Then, f_j, g_j is obtained by kernel density estimation on ${}^1X_j$ and ${}^0X_j$, and denote the estimates by $\hat{f}_j$ and $\hat{g}_j$, respectively. Thus, the feature augmentation for X_j using logarithm marginal density ratio transformation is shown as follows:

$$X_j^{'} = \log\hat{f}_j(X_j) - \log\hat{g}_j(X_j), \tag{1}$$

where $X_j^{'}$ denotes the transformed feature for the j th feature X_j.

2.2 Pls-Logistic Regression Classification Model

Suppose we have a pair of random variables (X, Y), where $X \in \mathbb{R}^p$ denotes the original features and $Y \in \{0, 1\}$ denotes the corresponding binary response. The procedures of pls-logistic regression is depicted as follows:

Step 1. Perform univariate logistic regression on each feature to obtain p coefficients denoted by $\omega^1 = \left(\omega_1, \omega_2, \cdots, \omega_p\right)$. Denote the normalized ω^1 by $\bar{\omega}^1$.
Step 2. Extract the first pls component t_1 by $t_1 = \mathbf{X} \cdot \bar{\omega}^1$.
Step 3. Perform OLS regression of X against t_1. Denote the residual of X by $\mathbf{X}^*$.
Step 4. Perform logistic regression on each feature of $\mathbf{X}^*$ against t_1 to obtain the p coefficients of features in $\mathbf{X}^*$, denoted by ω^2, and then normalize ω^2 to $\bar{\omega}^2$.
Step 5. Extract the second pls component t_2 by $t_2 = \mathbf{X}^* \cdot \bar{\omega}^2$.
Step 6. Repeat Step 3, Step 4 and Step 5 until the stopping criteria are satisfied.
Step 7. Denote by $t_1, t_2, \cdots, t_h$ the final extracted pls components. Perform the logistic regression on these pls components to build the classification model.

3 Proposed Intrusion Detection Model: Fa-Plslogistic

In this section, we present the main procedures of our proposed intrusion detection model based on pls-logistic with feature augmentation. By embedding the data quality improvement technique into pls-logistic, we can obtain an effective intrusion detection with good performances and less complexity. First, we perform feature transformations on the original features to obtain high-quality training data that can significantly improve the detection performances. Then, the pls-logistic regression is perform on the newly transformed data to conduct dimension reduction and build the intrusion detection model. For clarity, the detailed procedures are summarized as follows:

- **Step 1.** *Data transformation*
  Perform feature transformations on the original data to obtain high-qualified training data.
- **Step 2.** *Detection model building*
  Use the newly obtained data from Step 1 to train pls-logistic-based classifier and build the intrusion detection model.

- **Step 3.** *Intrusion detection*
  For a new testing sample, it is first transformed by the logarithm marginal density ratio transformation illustrated in Sect. 2.1; then, the transformed data is fed into the built intrusion detection model to classify it as either an intrusion or a normal.

4 Experimental Setting

4.1 Dataset Description

In our study, the NSL-KDD dataset is used to evaluate the performance of the proposed intrusion detection model. The NSL-KDD dataset is a modified version of KDD 99 dataset which is considered as the benchmark dataset in intrusion detection domain. However, the KDD 99 dataset suffers from some drawbacks [12, 13]. For example, there are redundant and duplicate records which cause the classifier would be biased towards these more frequent records. The NSL-KDD dataset was proposed by [14] by removing all the redundant samples and reconstituting the dataset, making it more reasonable not only in data size, but also in data structure. The NSL-KDD dataset contains TCP connections that consist of 41 features and one labeling feature.

4.2 Experimental Results and Discussion

In order to prevent the dominance of features with large ranges, we normalize the data into a range of [0, 1] before conducting the experiments. To evaluate our proposed detection model, the 10-fold cross validation has been adopted and the performance is evaluated by the following measurements according to the confusion matrix presented in Table 1.

Table 1. Confusion matrix

		Predicted	
		Attack	Normal
Actual	Attack	TP	FN
	Normal	FP	TN

Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$, Detection rate (DR) $= \frac{TP}{TP+FN}$, False alarm rate (FAR) $= \frac{FP}{TN+FP}$

To verify the effectiveness of our proposed intrusion detection model, we first compare the detection performance of Fa-plslogistic with that of the naïve-plslogistic detection model (pls-logistic regression on original data without feature transformation). The 10-fold cross validation results of these two detection models on NSL-KDD dataset with regard to accuracy, DR, FAR and training time are summarized in Table 2.

As the results shown in Table 2, our proposed intrusion detection model takes clear advantages over the naïve-plslogistic detection model, indicating that the data quality improvement technique can greatly boost the detection performance. More specifically,

Table 2. Performances of proposed methods

Metric	Pls-logistic (with feature augmentation)	Pls-logistic (without feature augmentation)
Accuracy (%)	97.39(0.33)	91.29(6.03)
DR (%)	96.95(0.32)	88.59(12.84)
FAR (%)	2.23(0.45)	6.35(3.05)
Training time (in sec)	98.66	137.51

the accuracy and detection rate of our proposed model both exceed 96%, while naïve-plslogistic only achieves 91.29% and 88. 59%, respectively. Besides, in terms of false alarm rate, our proposed method is below 2.3%, while naïve-plslogistic is over 6%. Moreover, the performances of our proposed is also more robust than that of naïve-plslogistic.

To further demonstrate the advantages of our proposed method, the training time required by Fa-plslogistic and naïve-plslogistic is also compared in Table 2. As shown, the training time of our proposed method is superior to that of naïve-plslogistic. Specifically, naïve-plslogistic demands about 1.39 time as much training time as Fa-plslogistic does. Thus, it can be inferred that our proposed method is much more concise than naïve-plslogistic, which can reduce the training time.

Therefore, according to the comparison results, it can be concluded that our proposed intrusion detection model is more effective than naïve-plslogistic and can achieve better detection performances.

Standard errors are in the parentheses in percentage form.

In addition, we examine which features are influential on the intrusion detection. Here, for simplicity, the feature whose coefficient is greater than 1 after standardization is considered to be important. Thus, the influential features recognized during the 10-fold cross-validation are shown in Table 3.

According to the results in Table 3, the important features for intrusion detection are listed in descending order by frequency: land, su_attempted, num_failed_logins, src_bytes, urgent, hot, num_root, num_compromised, root_shell, is_guest_login and dst_bytes. These features are helpful in practice to efficiently detect network intrusion and attacks.

Furthermore, in order to better interpret the effectiveness of our proposed method in intrusion detection, performance comparisons between our proposed model and other existing methods in intrusion detection using NSL-KDD dataset are conduct. The comparison results are summarized in Table 4.

From the comparison results shown in Table 4, our proposed method outperforms other intrusion detection methods with regard to detection accuracy. However, it should be noted that Table 4 just provides a snapshot of performance comparison between our proposed method and other detection methods. Thus, it can be claimed that our proposed method always performs better when compared to any other methods. Nevertheless, from the results above, we can make a conclusion that our proposed method

Table 3. Influential features for intrusion detection

K-fold	Influential feature
1	src_bytes, land, hot, su_attempted, num_root
2	land, num_failed_logins
3	land, urgent, num_failed_logins
4	src_bytes, land, hot, root_shell, su_attempted, is_guest_login
5	dst_bytes, num_compromised
6	src_bytes, land, num_compromised, root_shell, num_root
7	land, urgent, num_failed_logins, num_compromised, su_attempted, num_root
8	src_bytes, urgent, hot, num_failed_logins, su_attempted
9	su_attempted
10	src_bytes, land, urgent, hot, num_failed_logins, su_attempted, is_guest_login

Table 4. Performance comparisons of proposed method and other detection methods

Method	Accuracy (%)
GHSOM [15]	96.02
A-GHSOM [16]	96.63
Naïve Bayes + N2B [17]	96.50
AdaBoost [17]	90.31
Proposed method	97.39

still possesses advantages in intrusion detection and can provide inspirations for the following researches.

5 Conclusion

Intrusion detection system is critical to network security. In this paper, we proposed an effective intrusion detection model based on pls-logistic with feature augmentation. Though the pls-logistic classifier might achieve a good performance, the detection capacity is much more dependent on the quality of the training data. Therefore, in order to increase the detection capacity, we use the logarithm marginal density ratio transformation on the original data to obtain high-quality training data for pls-logistic before building the intrusion detection model. Empirical results on NSL-KDD dataset show that our proposed intrusion detection model is effective and can achieve good and robust detection performances.

Acknowledgments. This research was financially supported by National Natural Science Foundation of China (Grant No. 72001222, 61832001, 61702016).

References

1. Kumar, G., Thakur, K., Ayyagari, M.R.: MLEsIDSs: machine learning-based ensembles for intrusion detection systems—a review. J. Supercomput. **76**(11), 8938–8971 (2020). https://doi.org/10.1007/s11227-020-03196-z
2. Bamakan, S.M.H., Wang, H., Yingjie, T., Shi, Y.: An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing **199**, 90–102 (2016)
3. Moustafa, N., Hu, J., Slay, J.: A holistic review of network anomaly detection systems: a comprehensive survey. J. Netw. Comput. Appl. **128**, 33–55 (2019)
4. Tsai, C.F., Hsu, Y.F., Lin, C.Y., Lin, W.Y.: Intrusion detection by machine learning: a review. Expert Syst. Appl. **36**(10), 11994–12000 (2009)
5. Sommer, R., Paxson, V.: Outside the closed world: on using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy, pp. 305–316 (2010)
6. Wang, Y.: A multinomial logistic regression modeling approach for anomaly intrusion detection. Comput. Secur. **24**(8), 662–674 (2005)
7. Mok, M.S., Sohn, S.Y., Ju, Y.H.: Random effects logistic regression model for anomaly detection. Expert Syst. Appl. **37**(10), 7162–7166 (2005)
8. Ji, S.Y., Choi, S., Jeong, D.H.: Designing an internet traffic predictive model by applying a signal processing method. J. Netw. Syst. Manag. **23**(4), 998–1015 (2015)
9. Aburomman, A.A., Reaz, M.B.I.: A survey of intrusion detection systems based on ensemble and hybrid classifiers. Comput. Secur. **65**, 135–152 (2017)
10. Fan, J., Feng, Y., Jiang, J., Tong, X.: Feature augmentation via nonparametrics and selection (FANS) in high-dimensional classification. J. Am. Stat. Assoc. **111**(513), 275–287 (2016)
11. Bastien, P., Vinzi, V.E., Tenenhaus, M.: Pls generalised linear regression. Comput. Stat. Data Anal. **48**(1), 17–46 (2005)
12. Mahoney, M.V., Chan, P.K.: An analysis of the 1999 DARPA/lincoln laboratory evaluation data for network anomaly detection. In: Vigna, G., Kruegel, C., Jonsson, E. (eds.) RAID 2003. LNCS, vol. 2820, pp. 220–237. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45248-5_13
13. Bamakan, S.M.H., Wang, H., Yingjie, T., Shi, Y.: An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing **199**, 90–102 (2016)
14. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications, pp. 1–6. IEEE (2009)
15. Yu, Z., Tsai, J.J., Weigert, T.: An adaptive automatically tuning intrusion detection system. ACM Trans. Auton. Adapt. Syst. **3**(3), 10 (2008)
16. Ippoliti, D., Zhou, X.: A-GHSOM: an adaptive growing hierarchical self-organizing map for network anomaly detection. J. Parallel Distrib. Comput. **72**(12), 1576–1590 (2012)
17. Panda, M., Abraham, A., Patra, M.R.: Discriminative multinomial naive bayes for network intrusion detection. In: Proceedings of 2010 Sixth International Conference on Information Assurance and Security, pp. 5–10. IEEE (2010)

DeepHTTP: Anomalous HTTP Traffic Detection and Malicious Pattern Mining Based on Deep Learning

Yuqi Yu[1](✉), Hanbing Yan[1](✉), Yuan Ma[3], Hao Zhou[1], and Hongchao Guan[2]

[1] National Computer Network Emergency Response Technical Team/Coordination Center of China, Chaoyang District, Beijing, China
{yyq,yhb}@cert.org.cn
[2] Beijing University of Posts and Telecommunications, Haidian District, Beijing, China
[3] Chongqing Municipal Public Security Bureau, Yuzhong District Branch, Chongqing, China

Abstract. Hypertext Transfer Protocol (HTTP) accounts for a large portion of Internet application-layer traffic. Since the payload of HTTP traffic can record website status and user request information, many studies use HTTP protocol traffic for web application attack detection. In this work, we propose DeepHTTP, an HTTP traffic detection framework based on deep learning. Unlike previous studies, this framework not only performs malicious traffic detection but also uses the deep learning model to mine malicious fields of the traffic payload. The detection model is called AT-Bi-LSTM, which is based on Bidirectional Long Short-Term Memory (Bi-LSTM) with attention mechanism. The attention mechanism can improve the discriminative ability and make the result interpretable. To enhance the generalization ability of the model, this paper proposes a novel feature extraction method. Experiments show that DeepHTTP has an excellent performance in malicious traffic discrimination and pattern mining.

Keywords: Bi-LSTM · Attention mechanism · Anomalous HTTP traffic detection · Malicious pattern mining

1 Introduction

According to the 2018 Internet Security Report released by China National Computer Network Emergency Response Technical Team/Coordination Center (CNCERT/CC) [1], website attacks and exploits occur frequently. How to improve the ability of web attack detection is one of the urgent problems in the field of network security.

Among various network protocols, Hypertext Transfer Protocol (HTTP) occupies a considerable proportion of the application layer traffic of the Internet. Since HTTP traffic can record website access states and request content, it provides an excellent source of information for web application attack detection [2–4]. We focus on HTTP traffic mainly for three reasons. 1) Although protocol HTTPS is used by 57.4% of all the websites [5], HTTP traffic still accounts for a large proportion of network traffic. Research [6] shows that for smaller B2B websites, the uptake of HTTPS is low. Because they lack awareness

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 141–161, 2020.
https://doi.org/10.1007/978-981-33-4922-3_11

of the streaming importance of SSL. Also, the perceived complexity of switching to HTTPS is high. 2) A large majority of malware uses HTTP to communicate with their C&C server or to steal data. Many web application attacks use HTTP, such as Cross-site scripting attack (XSS), SQL injection, and so on. 3) The HTTP protocol is transmitted in clear text, which makes it easier to analyze network behaviors.

In this paper, we design DeepHTTP, a complete framework for detecting malicious HTTP traffic based on deep learning. The main contributions are as follows.

Firstly, unlike researches that only detect malicious URLs (Uniform Resource Locators) [7, 8], we extract both URL and POST body (if the HTTP method is POST) to detect web application attacks. This is of great help to portray network behavior more comprehensively.

Secondly, we perform an in-depth analysis of the types and encoding forms of HTTP traffic requests, then propose an effective method to extract content and structure features from HTTP payload (in this paper, “payload” refers to URL and POST body). Content and structure features are used for classification.

Thirdly, the detection model AT-Bi-LSTM is Bidirectional Long Short-Term Memory (Bi-LSTM) [9] with attention mechanism [10]. Since each HTTP request follows the protocol specification and grammar standards, we treat elements in traffic payload as vocabulary in natural language processing and use Bi-LSTM to learn the contextual relationship. The attention mechanism can automatically dig out critical parts, which can enhance the detection capabilities of the model. Due to the introduction of attention mechanism, the model is more interpretable than other deep learning models.

Finally, we design a module for malicious pattern mining. The “malicious pattern” is essentially a collection of strings representing web attacks. Specifically, we cluster malicious traffic entries and perform pattern mining for each cluster. Then we can generate new rules based on the mined malicious patterns. New rules will be configured into detection systems to capture specific types of web attacks.

In a word, DeepHTTP is a complete framework that can automatically distinguish malicious traffic and perform pattern mining. We set up a process that can verify and update data efficiently. The model is updated periodically so that it can adapt to new malicious traffic that appears over time.

The rest of this paper is organized as follows. Section 2 gives a summary of the relevant research. Section 3 briefly introduces the system framework and data preprocessing methods. The proposed model is introduced in detail in Sect. 4, including the malicious traffic detection model and pattern mining method. We launched a comprehensive experiment to demonstrate the effectiveness of the model. The experimental results are shown in Sect. 5. Section 6 gives the conclusions and future works.

2 Related Work

2.1 Malicious Traffic Detection

In recent years, quite a few researches are aiming for detecting anomaly traffic and web application attacks. Communication traffic contains lots of information that can be used to mine anomaly behaviors. Lakhina et al. [58] perform a method that fuses information from flow measurements taken throughout a network. Wang et al. [59] propose Anagram, a content anomaly detector that models a mixture of high-order n-grams designed to detect anomalous and "suspicious" network packet payloads. To select the important features from huge feature spaces, Zseby et al. [60] propose a multi-stage feature selection method using filters and stepwise regression wrappers to deal with feature selection problem for anomaly detection. The methods mentioned above care less about the structural features of communication payloads which are important for distinguishing anomaly attacking behaviors and mining anomaly patterns. In this paper, we put forward a structure extraction approach, which can help enhance the ability to detect anomaly traffic. The structure feature also makes an important role in pattern mining.

Existing approaches for anomalous HTTP traffic detection can be roughly divided into two categories according to data type: feature distribution-based methods [11, 12] and content-based methods [13]. Content-based methods can get rid of the dependency of artificial feature extraction and is suitable for different application scenarios. Nelms T et al. [14] use HTTP headers to generate control protocol templates including URL path, user-agent, parameter names, etc. Because Uniform Resource Locator (URL) is rich in information and often used by attackers to pass abnormal information, identifying malicious URLs is a hot studied problem in the security detection [8, 15, 16]. In this paper, we use both URL and POST body (if the HTTP method is POST) to detect web attacks. We do not use other parameters in the HTTP header because these fields (like Date, Host, and User-agent, etc.) have different value types and less valid information.

Various methods have been used for detection. Juvonen and Sipola [18] propose a framework to find abnormal behaviors from HTTP server logs based on dimensionality reduction. Researchers compare random projection, principal component analysis, and diffusion map for anomaly detection. Ringberg et al. [19] propose a nonparametric hidden Markov model with explicit state duration, which is applied to cluster and scout the HTTP-session processes. This approach analyses the HTTP traffic by session scale, not the specific traffic entries. Additionally, there are also many kinds of research based on traditional methods such as IDS (intrusion detection system and other rule-based systems) [3, 20–22]. Since malicious traffic detection is essentially an imbalanced classification problem, many studies propose anomaly-based detection approaches that generate models merely from the benign network data [17]. However, in practical applications, the anomaly-based detection model usually has a high false-positive rate. This problem undoubtedly increases the workload of manual verification.

With the rapid development of artificial intelligence, deep learning has been widely used in various fields and has a remarkable effect on natural language processing. Recently, deep learning has been applied to anomaly detection [8, 23–25]. Erfani et al. [25] present a hybrid model where an unsupervised DBN is trained to extract generic underlying features, and a one-class SVM is trained from the features learned by the DBN. LSTM model is used for anomaly detection and diagnosis from System Logs [24]. In this article, we use deep learning methods to build detection models to enhance detection capabilities.

2.2 Pattern Mining Method

In addition to detecting malicious traffic and attack behaviors, some researches focus on pattern mining of cluster traffic. Most existing methods for traffic pattern recognition and mining are based on clustering algorithms [26, 27]. Le et al. [27] propose a framework for collective anomaly detection using a partition clustering technique to detect anomalies based on an empirical analysis of an attack's characteristics. Since the information theoretic co-clustering algorithm is advantageous over regular clustering for creating a more fine-grained representation of the data, Mohiuddin Ahmed et al. [28] extend the co-clustering algorithm by incorporating the ability to handle categorical attributes which augments the detection accuracy of DoS attacks. In addition to the clustering algorithm, JT Ren [29] conducts research on network-level traffic pattern recognition and uses PCA and SVM for feature extraction and classification. I. Paredes-Oliva et al. [30] build a system based on an elegant combination of frequent item-set mining with decision tree learning to detect anomalies.

The signature generation has been researched for years and has been applied to protocol identification and malware detection. FIRMA [31] is a tool that can cluster network traffic clusters obtained by executing unlabeled malware binaries and generate a signature for each cluster. Terry Nelms et al. [32] propose ExecScent, a system that can discover new C&C domains by building adaptive templates. It generates a control protocol template (CPT) for each cluster and calculates the matching score to find similar malware. These tools have proven to automatically generate valid signatures, but the process still needs to define the composition of the initial signature or template in advance. As far as we know, signature generation is rarely used in web attack detection. The study of pattern mining for malicious traffic is not yet mature.

In recent years, the attention-based neural network model has become a research hotspot in deep learning, which is widely used in image processing [33], speech recognition [34], and healthcare [35]. Attention mechanism has also proved to be extremely effective. Luong et al. [36] first design two novel types of attention-based models for machine translation. Since the attention mechanism can automatically extract important features from raw data, it has been applied to relation Classification [37] and abstract extraction [38]. To the best of our knowledge, as for HTTP traffic detection and pattern mining, proposed models rarely combine sequence models with attention mechanism. Hence, in this paper, we build a model based on attention mechanism, which can get rid of the dependency of artificial extraction features and do well in pattern mining.

3 Preliminaries

3.1 DeepHTTP Architecture and Overview

The “Rule Engine” mentioned in this paper is an engine that consists of many rules. Each rule is essentially a regular expression used to match malicious HTTP traffic that matches a certain pattern. Generally, the expansion of the rule base relies on expert knowledge. It requires high labor costs. And the malicious traffic that the “Rule Engine” can detect is limited. Therefore, we additionally introduce a deep learning model based on the attention mechanism, which can identify malicious traffic entries that are not detected by the “Rule Engine”. Also, the pattern mining module can automatically extract the string patterns in the traffic payload, which can greatly reduce the workload of rule extraction.

In this paper, rules can be roughly divided into seven categories according to the type of web application attack: File Inclusion (Local File Inclusion and Remote File Inclusion), framework vulnerability (Struts2, CMS, etc.), SQL Injection (Union Select SQL Injection, Error-based SQL Injection, Blind SQL Injection, etc.), Cross-Site Scripting (DOM-based XSS, Reflected XSS, and Stored XSS), WebShell (Big Trojan, Small Trojan and One Word Trojan [39]), Command Execution (CMD) and Information Disclosure (system file and configuration file).

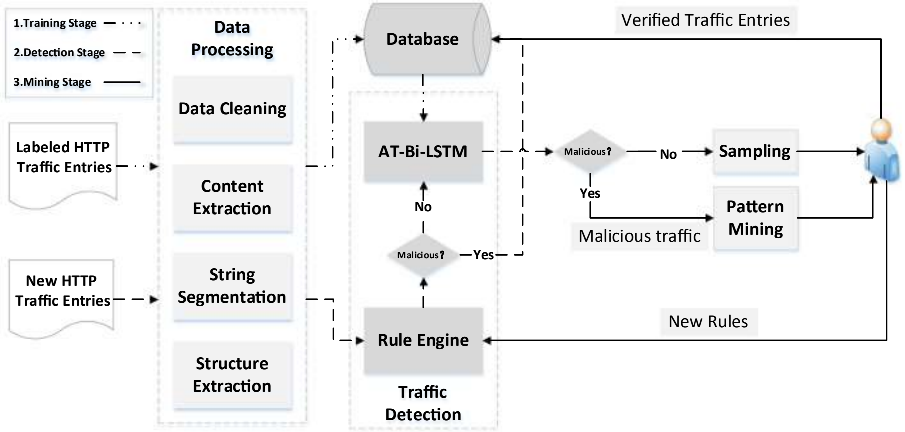


Fig. 1. DeepHTTP architecture.

DeepHTTP is a complete framework that can detect web application attacks quickly and efficiently. In this section, we introduce three stages of DeepHTTP (see Fig. 1), which are training stage, detection stage, and mining stage.

- **Training stage.** The core task of this phase is to train the model (AT-Bi-LSTM). It includes data processing and model training. First, we put the labeled dataset into the data processing module to obtain content and structure features of traffic payload. After that, we divide the processed formatted data into training, test, and verification sets and store in the database. To enhance the robustness of the model, we build data sets containing positive and negative samples in different proportions and use cross-validation to train the model.
- **Detection stage.** The pre-trained model and the “Rule Engine” are used for anomaly traffic detection. After data processing, new HTTP entries are first entered into the “Rule Engine” for detection. For the entries which are detected by the engine, we labeled the data and update them directly into the database. Other traffic entries will be entered into the pre-trained model for detection. Anomaly traffic entries detected by AT-Bi-LSTM will be used in the mining stage.

- **Mining stage.** The main works of this phase are verifying the anomalous traffic labeled by the model and then mining malicious patterns. Generally speaking, there are a large number of traffic entries that the model identifies as malicious. To improve efficiency, we first cluster and sample the data. Specifically, malicious traffic will be divided into different clusters by clustering. In each cluster, we mine malicious patterns based on attention mechanism and then generate new rules. Simultaneously, we sample a small number of entries from each cluster and perform manual verification. Verified malicious data will be updated regularly to the database and new rules will be updated regularly to "Rule Engine".

DeepHTTP is a complete closed-loop workflow. The detection model and "Rule Engine" complement each other. The timing update and feedback mechanism can continuously improve the detection ability of the system, which is the main reason for the practicability of the framework. Data processing, traffic detection model, and pattern mining method are critical parts in DeepHTTP, which will describe in the later sections.

3.2 Data Preprocessing

Data Collection. The study spends nearly half a year to collect actual traffic. Nearly 1.5 million malicious HTTP traffic samples are accumulated through vulnerability scanning, rule filtering, and manual verification. After sampling and deduplication, we eventually collect 10, 645, 12 malicious samples.

- **Rule-based collection method.** Specifically, we collect network traffic from the university network monitoring system and filter out HTTP traffic. To protect the privacy of teachers and students, we remove sensitive content from the data. Then, we use the "Rule Engine" mentioned in Sect. 3.1 to identify malicious traffic.
- **Tools-based collection method.** In order to enrich the type of malicious traffic, we use kali [40], Paros [41], W3AF [42] to perform simulation attack and vulnerability scanning. We collect relevant traffic as malicious traffic samples.
- **Model-based collection method.** As described in Sect. 3.1, after manual verification, malicious traffic entries detected by AT-Bi-LSTM are periodically updated to the data set.

Data Cleaning. We parse HTTP traffic packets and extract Uniform Resource Locator (URL) and POST body (if the request method is POST). Then, we mainly perform the following data cleaning operations:

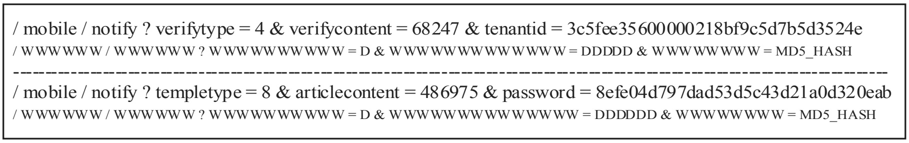
- URL decoding: Since URL data often been encoded, we perform URL decoding.
- Payload decoding: Many strings in traffic payload are encoded by different encoding methods, like MD5, SHA, and Base64, etc. For these strings, we identify the encoding type and replace them with the predefined flag (see Table 1).
- We replace garbled characters and invisible characters with null characters.
- Since the binary stream data in the Post request body does not contain semantic information, we replace this kind of data with the predefined flag (see Table 1).

String Segmentation. Text vectorization is the key to text mining. Numerous studies use n-grams [43] to extract the feature of payloads [44–46]. This method can effectively capture the byte frequency distribution and sequence information, but it is easy to cause dimension disaster. To prevent dimensional disaster, we split the string with special characters. The special characters refer to characters other than English letters and numbers, such as *"@", "!", "#", "%", "^", "&", "*", "?"*, etc. Here is an instance. Suppose the decoded data is: "*/tienda1/publico/vaciar.jsp <EOS> B2 = Vaciar carrito; DROP TABLE usuarios; SELECT * FROM datos WHERE nombre LIKE*". "*<EOS>*" is the connection symbol. After string splitting, the data is denoted as: "*/tienda1 /public /vaciar. jsp <EOS> B2 = Vaciar carrito; DROP TABLE usuarios; SELECT * FROM datos WHERE nombre LIKE*". Strings are connected by spaces. Another benefit of this approach is that it makes the results of malicious pattern mining more understandable. In this example, the malicious pattern we want to obtain from the data is *{"SELECT", "FROM", "WHERE"}*. However, if we use n-grams ($n = 3$) or character-based method [39], the result may be denoted as *{"SEL", "ELE", ..., "ERE"} or {"S", "L",..., "R"}*, which is not intuitive.

Structure Feature Extraction. To better measure the similarity of URLs, Terry Nirm, etc. [32] use a set of heuristics to detect strings that represent data of a certain type and replaces them accordingly using a placeholder tag containing the data type and string length. Inspired by this, the paper uses a similar way to extract structure features from HTTP payload. The "structure feature" mentioned in this paper refers to string type other than the meaning of the string itself. We replace string with predefined flags according to their data type. The types of data we currently recognize include hash (MD5, SHA, and Base64), hexadecimal, binary, Arabic numerals and English alphabet (upper, lower and mixed case) .etc. The main replacement rules are shown in Table 1.

Table 1. Characters replacement rules.

Encoding type	Replacement string
MD5 hash	'MD5_HASH'
SHA hash	'SHA_HASH'
Base64	'BASE64_ENCODE'
Hexadecimal	'HEXADECIMAL'
Encryption	'ENCRYPTION'
Binary	'BINARY'



```
/ mobile / notify ? verifytype = 4 & verifycontent = 68247 & tenantid = 3c5fee35600000218bf9c5d7b5d3524e
/ WWWWWW / WWWWWW ? WWWWWWWWWW = D & WWWWWWWWWWWWW = DDDDD & WWWWWWWW = MD5_HASH
--------------------------------------------------------------------------------------------------------
/ mobile / notify ? templetype = 8 & articlecontent = 486975 & password = 8efe04d797dad53d5c43d21a0d320eab
/ WWWWWW / WWWWWW ? WWWWWWWWWW = D & WWWWWWWWWWWWWW = DDDDDD & WWWWWWWW = MD5_HASH
```

Fig. 2. An example of structure extraction.

Here is an example of a structure feature extraction (see Fig. 2). Since the encoding type of the string *“3c5fee35600000218bf9c5d7b5d3524e”* is MD5 (We use “ hashID” [47] to identify the different types of hashes.), we replace it with *“MD5_HASH”*. For those string not belong to any special type, we replace each character in the string with the specified character. *“D”* for Arabic numeral and *“W”* for the English alphabet (not case sensitive). Since the string *“68247”* consists of five Arabic numerals, we replace it with five *“D”*. Obviously, by extracting structural features, we can easily find requests with different content but almost the same in data type.

4 Our Approach

4.1 Anomaly HTTP Traffic Detection

The goal of the proposed algorithm is to identify anomaly HTTP traffic based on semantics and structure of traffic entries. Figure 3 shows the high-level overview of the proposed model. The model (AT-Bi-LSTM) contains five components: input layer, word embedding layer, Bi-LSTM layer, attention layer and output layer.

Problem Definition. Let $\mathbf{R} = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_i, \ldots, \boldsymbol{R}_N\}$ be the set of HTTP traffic entries after data processing. For each traffic entry $\boldsymbol{R}_i(\boldsymbol{i} = 1, 2, \ldots, \boldsymbol{N})$, there are two sequences $\boldsymbol{S}_i^1 = \{\boldsymbol{c}_{11}, \boldsymbol{c}_{12}, \boldsymbol{c}_{13}, \ldots, \boldsymbol{c}_{1n}\}$ and $\boldsymbol{S}_i^2 = \{\boldsymbol{c}_{21}, \boldsymbol{c}_{22}, \boldsymbol{c}_{23}, \ldots, \boldsymbol{c}_{2n}\}$, which respectively represent content sequence and structure sequence. Because structure sequence is derived from content sequence, the length of both sequence is equal to *n*.

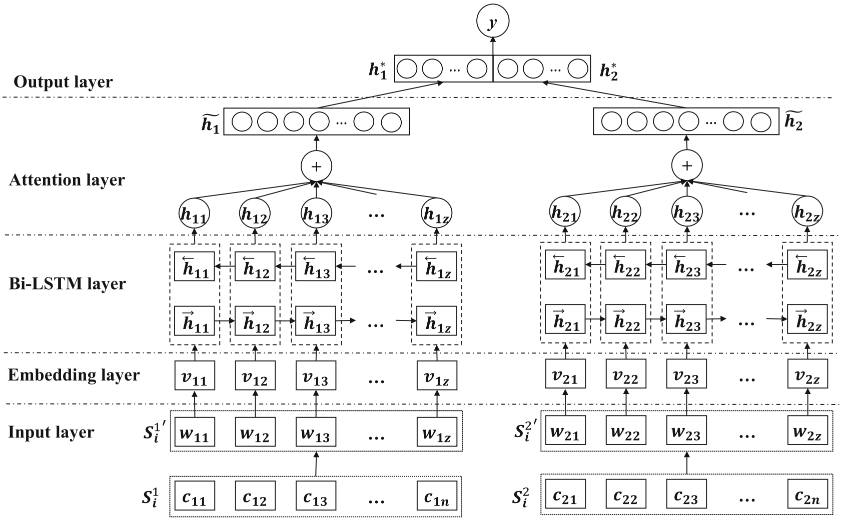

y
Output layer
h_1^*
h_2^*
$\widetilde{h_1}$
$\widetilde{h_2}$
Attention layer
+
h_{11} h_{12} h_{13} ... h_{1z}
h_{21} h_{22} h_{23} ... h_{2z}
Bi-LSTM layer
$\overleftarrow{h}_{11}$ $\overleftarrow{h}_{12}$ $\overleftarrow{h}_{13}$ ... $\overleftarrow{h}_{1z}$
$\overrightarrow{h}_{11}$ $\overrightarrow{h}_{12}$ $\overrightarrow{h}_{13}$ ... $\overrightarrow{h}_{1z}$
$\overleftarrow{h}_{21}$ $\overleftarrow{h}_{22}$ $\overleftarrow{h}_{23}$ ... $\overleftarrow{h}_{2z}$
$\overrightarrow{h}_{21}$ $\overrightarrow{h}_{22}$ $\overrightarrow{h}_{23}$ ... $\overrightarrow{h}_{2z}$
Embedding layer
v_{11} v_{12} v_{13} ... v_{1z}
v_{21} v_{22} v_{23} ... v_{2z}
Input layer
$S_i^{1\prime}$ w_{11} w_{12} w_{13} ... w_{1z}
$S_i^{2\prime}$ w_{21} w_{22} w_{23} ... w_{2z}
S_i^1 c_{11} c_{12} c_{13} ... c_{1n}
S_i^2 c_{21} c_{22} c_{23} ... c_{2n}

Fig. 3. Model architecture.

Input Layer. In this paper, we use the content and structure sequence after word segmentation as a corpus, and select words that are common in the corpus to build a vocabulary according to term frequency inverse document frequency (TF-IDF) [48]. Then, the unique index is generated for each word in the vocabulary. We convert the word sequences ($\boldsymbol{S_i^1}$ and $\boldsymbol{S_i^2}$) to final input vectors ($\boldsymbol{S_i^{1\prime}}$ and $\boldsymbol{S_i^{2\prime}}$), which are composed of indexes. The length of input vector is denoted as z, which is a hyper-parameter (the fixed length in this paper is set to 300 because the proportion of sequence length within 300 is 0.8484). The excess part of input sequence is truncated, and the insufficient part is filled with zero. Formally, the sequence of content can be converted to $\boldsymbol{S_i^{1\prime}} = \{\boldsymbol{w}_{11}, \boldsymbol{w}_{12}, \boldsymbol{w}_{13}, \ldots, \boldsymbol{w}_{1z}\}$ and the sequence of structure can be expressed as $\boldsymbol{S_i^{2\prime}} = \{\boldsymbol{w}_{21}, \boldsymbol{w}_{22}, \boldsymbol{w}_{23}, \ldots, \boldsymbol{w}_{2z}\}$. Here is an example. Given a sequence of content: *{'/', 'admin', '/', 'caches', '/', 'error_ches', '.', 'php' }*. The input vector with fix length can be denoted as *[23, 3, 23, 56, 23, 66, 0, 0, ..., 0]*. Since the index of 'admin' in vocabulary is 3, the second digit in the vector is 3. And since the length of this sequence is less than fixed length, the rest of the vector is filled with zeros.

Embedding Layer. Take a content sequence of i-th traffic entry as an example. Given $\boldsymbol{S_i^{1\prime}} = \{\boldsymbol{w}_{11}, \boldsymbol{w}_{12}, \ldots, \boldsymbol{w}_{1k}, \ldots, \boldsymbol{w}_{1z}\}$, we can obtain vector representation $v_{1k} \in R^m$ of each word $w_{1k} \in R^1 (k = 1, 2, \ldots, z)$ as follows:

$$v_{1k} = ReLU(W_e w_{1k} + b_e) \tag{1}$$

where m is the size of embedding dimension, $W_e \in R^{m\times 1}$ is the weight matrix, and $b_e \in R^m$ is the bias vector. Rectified Linear Unit (ReLU) is the rectified linear unit defined as ReLU($\boldsymbol{v}$) = max($\boldsymbol{v}$, 0), where max() applies element-wise to vector.

Bidirectional Long Short-Term Memory. We employ Bidirectional Long Short-Term Memory (Bi-LSTM), which can exploit information both from the past and the future to improve the prediction performance and learn the complex patterns in HTTP requests better. A Bi-LSTM consists of a forward and backward LSTM. Given embedding vector $\{\boldsymbol{v}_{11}, \boldsymbol{v}_{12}, \ldots, \boldsymbol{v}_{1k}, \ldots, \boldsymbol{v}_{1z}\}$ of content sequence of i-th traffic entry $\boldsymbol{R_i}$, the forward LSTM $\vec{f}$ reads the input sequence from $\boldsymbol{v}_{11}$ to $\boldsymbol{v}_{1z}$, and calculates a sequence of forward hidden states $(\vec{\boldsymbol{h}}_{11}, \vec{\boldsymbol{h}}_{12}, \ldots, \vec{\boldsymbol{h}}_{1k}, \ldots, \vec{\boldsymbol{h}}_{1z})$ ($\vec{\boldsymbol{h}}_{1k} \in \boldsymbol{R^p}$) and p is the dimensionality of hidden states). The backward LSTM $\overleftarrow{\boldsymbol{f}}$ reads the input sequence in the reverse order and product a sequence of backward hidden states $\left(\overleftarrow{\boldsymbol{h}}_{11}, \overleftarrow{\boldsymbol{h}}_{12}, \ldots, \overleftarrow{\boldsymbol{h}}_{1k}, \ldots, \overleftarrow{\boldsymbol{h}}_{1z}\right)$ ($\overleftarrow{\boldsymbol{h}}_{1k} \in \boldsymbol{R^p}$). The final latent vector representation $\boldsymbol{h}_{1k} = \left[\vec{\boldsymbol{h}}_{1k}; \overleftarrow{\boldsymbol{h}}_{1k}\right]^{\boldsymbol{T}}$ $\left(\boldsymbol{h}_{1k} \in \boldsymbol{R^{2p}}\right)$ can be obtained by concatenating the forward hidden state $\vec{\boldsymbol{h}}_{1k}$ and the backward one $\overleftarrow{\boldsymbol{h}}_{1k}$. We deal with the embedding vector of structure sequence in the same way.

Attention Layer. In this layer, we apply attention mechanism to capture significant information, which is critical for prediction. General attention is used to capture the relationship between h_t and $h_i (1 \le i < t)$:

$$\alpha_{ti} = h_t^T W_\alpha h_i \tag{2}$$

$$\alpha_t = softmax\left(\left[\alpha_{t1}, \alpha_{t2}, \ldots, \alpha_{t(t-1)}\right]\right) \tag{3}$$

where $W_\alpha \in R^{2p \times 2p}$ is the matrix learned by model, α_t is the attention weight vector calculated by softmax function. Then, the context vector $c_t \in R^{2p}$ can be calculated based on the weights obtained from Eq. (3). The hidden states from h_1 to h_{t-1} can be calculated by the following formulas:

$$c_t = \sum_{i}^{t-1} \alpha_{ti} h_i \tag{4}$$

We combine current hidden state h_t and context vector c_t to generate the attentional hidden state as follows:

$$\widetilde{h_t} = \tanh(W_c[c_t; h_t]) \tag{5}$$

where $W_c \in R^{r \times 4p}$ is the weight matrix in attention layer, and r is the dimensionality of attention state. $\widetilde{h_1}$ and $\widetilde{h_2}$ can be obtained using Eq. (2) to Eq. (5), which denote the attention vector of content and structure sequence learned by the model.

Output Layer. Before feeding the attention vector into softmax function, the paper apply dropout regularization randomly disables some portion of attention state to avoid overfitting. It is worth noting that we concatenate vector of content and structure to generate output vector for prediction. The classification probability is calculated as follows:

$$p = softmax\left(w_s\left[h_1^*; h_2^*\right] + b_s\right) \tag{6}$$

where h_1^* is the output of $\widetilde{h_1}$ after dropout strategy, h_2^* is the output of $\widetilde{h_2}$. $w_s \in R^{q \times r}$ and $b_s \in R^q$ are the parameters to be learned.

$$\hat{y} = argmax(p) \tag{7}$$

where $\hat{y}$ is the label predicted by the attention model.

Objective Function. The paper calculate the loss for all HTTP traffic entries using the cross-entropy between the ground truth $y_i \in (0, 1)$ and the predicted $p_i (i = 1, 2, \ldots, N)$:

$$L = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(p_{i1}) + (1 - y_i) log(1 - p_{i1}) \tag{8}$$

where N is the number of traffic entries, p_{i1} denotes the probability that the i-th sample is predicted to be malicious.

We train the model to minimize the objective function so that the model automatically learns the appropriate parameters. The model can automatically learn the feature expression of input data without manual feature extraction. In addition to outputting the judgment results, the model will also output attention weights which will be used as important inputs for the pattern mining part. The introduction of attention mechanism makes this model more explanatory than other deep learning models.

4.2 Mining Stage

The function of this module is to interpret the results of the model and extract the string pattern. For malicious traffic that is not detected by the rules engine but is discriminated by the model, we perform pattern mining and verification. Figure 4 shows the architecture of the mining stage.

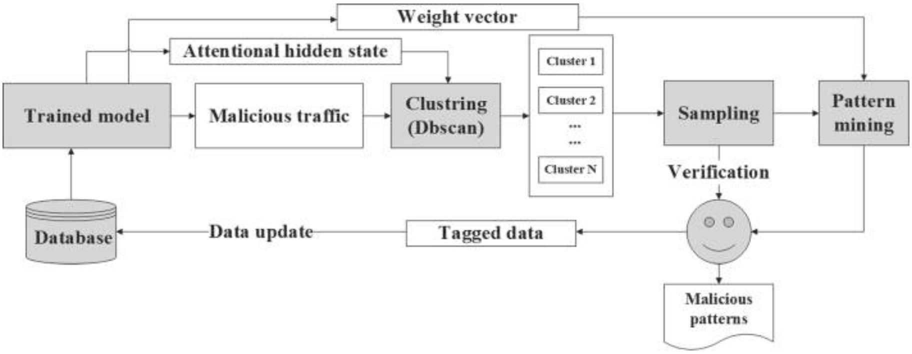


Fig. 4. The architecture of mining stage.

Clustering. We cluster traffic entries that were flagged as malicious by AT-Bi-LSTM. Specifically, we feed the attentional hidden state (obtained by Eq. (5) in Sect. 4.1) into the clustering model. The clustering method we apply is DBSCAN [49], a density-based clustering algorithm, which does not require prior declaring the number of clusters. After clustering, we obtain several clusters. Traffic entries in each cluster are similar in content or structure.

Tag Verification. In practical applications, there are massive suspicious HTTP requests every day. There is no doubt that manual verification requires a lot of time and effort. In this paper, we use clustering and sampling to reduce the workload. After clustering, we sample some entries from each cluster for verification. If the predicted labels of these samples are consistent with the ground-truth, then all the prediction results in this cluster are considered correct.

Pattern Mining. This module can mine the string pattern of the payload of malicious traffic. Experts generate new rules based on the results of pattern mining, which can reduce the workload of manual extraction. As mentioned in Sect. 3.1, the attention weight vector obtained in the attention layer can reflect the crucial parts of the payload. Therefore, for each malicious traffic entry, we dig out the key parts according to the corresponding attention weight vector. The greater the weight is, the more important the word is.

Specifically, given a cluster with N traffic entries $T = \{t_1, t_2, ..., t_N\}$, we perform pattern mining according to the following steps:

- **Get a keyword set according to attention weight.** AT-Bi-LSTM can output the attention weight vector (obtained by Eq. (3)). For each traffic entry $\boldsymbol{t_i}(i = 1, 2, \ldots, N)$, we get n keywords $\boldsymbol{K_i} = \{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k_n}\}$ according to its weight vector. The greater the weight, the more important the word is. At last, we can obtain a set of keywords $K = \{\boldsymbol{K}_1, \boldsymbol{K}_2, \ldots, \boldsymbol{K_N}\}$ identified by the model.
- **Extracting frequent patterns.** The goal of this step is to unearth words that not only frequently occur in this cluster but also recognized by the model as key parts. We calculate the co-occurrence matrix of keywords in set K. If we discovery several words in keywords set K to appear together frequently, then the combination of these words can represent a malicious pattern. The malicious pattern can be used as an effective basis for security personnel to extract new filtering rules.

5 Evaluation

5.1 Dataset

We use the method mentioned in Sect. 3.2 to build the HTTP traffic dataset. For the collected data, we perform manual verification and tagging. Finally, the total number of labeled data is 2,095,222, half of them are malicious traffic entries. The types and quantities of tagged malicious samples are shown in Table 2. Moreover, we prepare five million unmarked HTTP traffic for model testing.

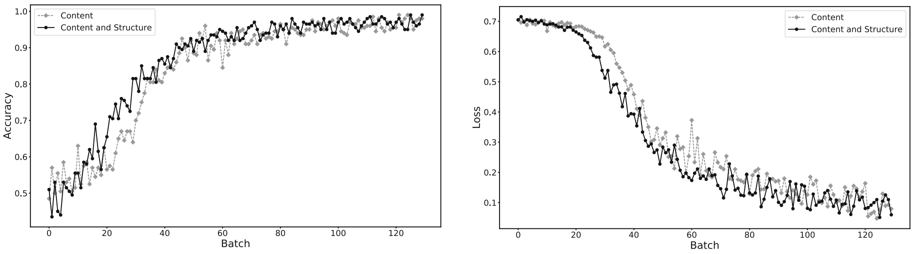
Table 2. Distribution of malicious traffic entries.

Data type	Number
Deserialization	6014
CMS	5836
File inclusion	46438
SQL injection	463776
Webshell	288050
XSS	127750
Sensitive data exposure	16656
Middleware vulnerability	47614
Struts2 vulnerability	42477
Botnet	19901
Total	1064512

5.2 Validation of Structural Feature Extraction Method

To verify the effectiveness of the structural feature extraction method, we compare the convergence speed and detection ability of the model trained by different features.

We record the loss and accuracy of each iteration of the model and draw the loss curve and the accuracy curve (Fig. 5). To balance the memory usage and model training efficiency, the best batch size is set to 200. As we observe from the figure, the model trained based on content and structural features converge faster. In other words, after fusing structural features, the learning rate has been enhanced, and it can reach the convergence state faster.



(a) Accuracy curve

(b) Loss curve

Fig. 5. Accuracy curve and loss curve.

Moreover, in unbalanced dataset, we compare the effects of models trained by different features. As shown in Table 3, the model trained based on content and structure features performs better. The reason is that structural features increase the generalization ability of the model.

Table 3. Performance of models trained by different features

Different features	Precision	Recall	F1-score	AUC
Content feature	0.9856	0.8765	0.9278	0.9382
Structure feature	0.9633	0.6643	0.7863	0.8320
Content and structure features	0.9560	0.9608	**0.9584**	**0.9795**

5.3 Model Comparison

We use 3-gram [43], TF-IDF [48], Doc2vec [50] and Character_level feature extraction method [8, 39] to obtain the feature vector of the payload. Then, we compared the effects of models between classic machine learning methods and machine learning models, including Support Vector Machine(SVM) [51], Random Forest(RF) [52], eXtreme Gradient Boosting(XGBoost) [53], Convolutional neural networks (CNNs) [54], Recurrent neural networks (RNNs) [55, 56], Long short term memory (LSTM) [57] and the proposed model AT-Bi-LSTM.

Detection in Labeled Dataset. We sample 1.1 million traffic entries from labeled dataset (as described in Sect. 3.2) to build a balanced dataset (550,000 for normal samples and 550,000 for malicious samples). To approximate the actual situation, we also sample 1.1 million traffic entries from labeled dataset to build an unbalanced dataset (1,000,000 for normal samples and 100,000 for malicious samples). Then the data set is divided into training set, test set and verification set according to the ratio of 6:2:2. The evaluation metrics consist of precision, recall, F1-score.

Table 4. Model performance in labeled dataset.

Dataset	Classifier	Precision	Recall	F-score
Balanced dataset	3-gram_TF-IDF_SVM	0.9607	0.9564	0.9585
	3-gram_TF-IDF_RF	0.9518	0.9269	0.9378
	3-gram_TF-IDF_XGBoost	0.9755	0.9683	0.9717
	Doc2vec_SVM	0.9365	0.9201	0.9274
	Doc2vec_ RF	0.9646	0.9444	0.9534
	Doc2vec_XGBoost	**0.9810**	**0.9753**	**0.9781**
	Doc2vec_CNN	0.9611	0.9467	0.9538
	Doc2vec_LSTM	0.9765	0.9538	0.9650
	Doc2vec_Bi-LSTM	0.9852	0.9791	0.9821
	Character_Level_CNN	0.9556	0.9461	0.9508
	Character_Level_LSTM	0.9895	0.9847	0.9870
	Character_Level_Bi-LSTM	**0.9954**	**0.9921**	**0.9937**
	AT-Bi-LSTM	**0.9979**	**0.9963**	**0.9970**
Unbalanced dataset	3-gram_TF-IDF_SVM	0.6573	0.5987	0.6266
	3-gram_TF-IDF_RF	0.7036	0.6835	0.6934
	3-gram_TF-IDF_XGBoost	0.7499	0.6937	0.7207
	Doc2vec_SVM	0.7531	0.6111	0.6061
	Doc2vec_ RF	0.8212	0.7484	0.7675
	Doc2vec_XGBoost	0.8844	0.8570	0.8683
	Doc2vec_CNN	0.8823	0.7851	0.8308
	Doc2vec_LSTM	0.8921	0.8235	0.8564
	Doc2vec_Bi-LSTM	0.9011	0.8221	0.8597
	Character_Level_CNN	0.9365	0.9342	0.9353
	Character_Level_LSTM	0.9485	0.9456	0.9470
	Character_Level_Bi-LSTM	0.9545	0.9574	0.9559
	AT-Bi-LSTM	**0.9661**	**0.9609**	**0.9635**

We can conclude the following conclusions according to Table 4. First, in the balanced dataset, Doc2vec_XGBoost, Character_Level_Bi-LSTM, and AT-Bi-LSTM perform well. However, in the imbalanced dataset, the detection capabilities of Doc2vec_XGBoost is not as good as deep learning models. Second, although the character-level deep learning models are comparable to AT-Bi-LSTM, the model proposed in this article is superior in interpretability. Finally, AT-Bi-LSTM is superior to all baseline models in almost all metrics. In unbalanced data sets, the superiority of the proposed model is even more pronounced.

At the same time, we record the training time of each model (see Fig. 6). Doc2vec-based deep learning models take more time because using Doc2vec to obtain sentence vectors requires additional training time. Because CNN has faster training speed, the training time of Character_Level_CNN is the least. The training time of AT-Bi-LSTM is at the middle level. It is acceptable in practical application.

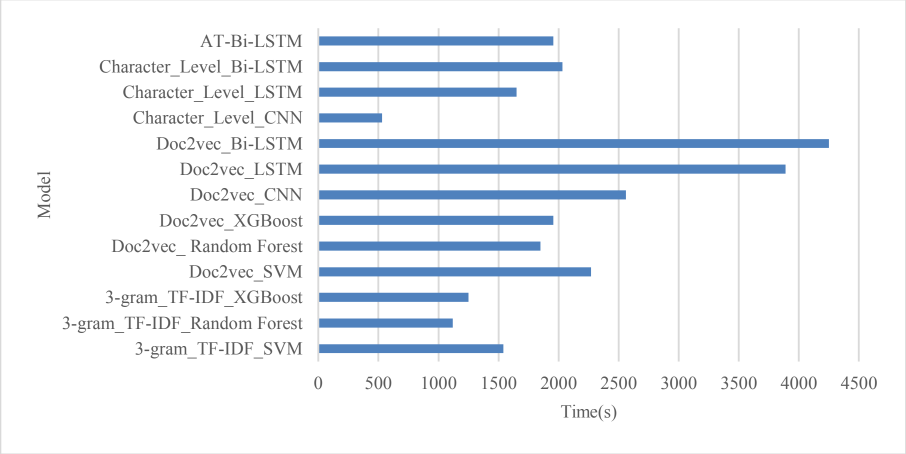


Fig. 6. Training time of models.

Detection in Unlabeled Dataset. We conduct comparative experiments using five million unlabeled traffic entries. Rules in "Rule Engine" are derived from expert knowledge so that we use the rules engine to verify the validity of the detection model. The explanation of the assessment indicators is as follows:

N_M. The number of malicious entries detected by the model.

N_RE. The number of malicious entries detected by the "Rule Engine".

N_M ∩ RE. The number of malicious entries detected by both the model and the "Rule Engine".

M-*RE*. A collection of malicious entries detected by the model but not detected by the "Rule Engine".

N_TP. The number of true positive samples in the M-*RE*.

N_FP. The number of false positive samples in the M-*RE*.

N_TP and N_FP are depend on manual verification.

Rule_Coverage_Rate (RCR) = N_M ∩ RE/N_RE. It represents the coverage of the model towards the "Rule Engine".

False_Rate (FR) = N_FP /N_M. It means the false rate of the model.

New_Rate (NR) = N_TP/N_M. It represents the ability of the model to identify malicious traffic outside the scope of the "Rule Engine".

We adopt the "Rule Engine" to extract malicious entries across the overall unlabeled traffic set. The amount of malicious traffic entries detected by "Rule Engine" (NMT_RE) equals to 217100. The result of model evaluation in the unlabeled dataset is shown in Table 5. According to the value of RCR, Doc2vec_Bi-LSTM, Character_level_CNN and AT-Bi-LSTM can basically cover the detection results of the "Rule Engine". However, Doc2vec_Bi-LSTM and Character_level_CNN have a higher false rate. Overall, AT-Bi-LSTM is superior to other models.

Table 5. Model results in the unlabeled dataset.

Model	N_M	N_RE	N_M ∩ RE	N_TP	N_FP	RCR	FR	NR
3-gram_TF-IDF_SVM	231246	217100	98965	38957	93324	0.4558	0.4036	0.1685
3-gram_TF-IDF_RF	234796	217100	102578	39875	92343	0.4725	0.3933	0.1698
3-gram_TF-IDF_XGBoost	265478	217100	119867	48057	97554	0.5521	0.3675	0.1810
Doc2vec_SVM	250164	217100	117687	47895	84582	0.5421	0.3381	0.1915
Doc2vec_ RF	302546	217100	116598	48965	136983	0.5371	0.4528	0.1618
Doc2vec_XGBoost	348951	217100	124263	53248	171440	0.5724	0.4913	0.1526
Doc2vec_CNN	458964	217100	169542	91458	197964	0.7809	0.4313	0.1993
Doc2vec_LSTM	486525	217100	189981	90259	206285	0.8751	0.4240	0.1855
Doc2vec_Bi-LSTM	589647	217100	200143	99653	289851	**0.9219**	0.4916	0.1690
Character_level_CNN	653287	217100	198756	180145	274386	**0.9155**	0.4200	**0.2758**
Character_level_LSTM	325648	217100	165478	55641	104529	0.7622	0.3210	0.1709
Character_level_Bi-LSTM	295876	217100	187569	31542	76765	0.8640	**0.2594**	0.1066
AT-Bi-LSTM	428270	217100	206809	110974	110487	**0.9526**	**0.2580**	**0.2591**

5.4 Malicious Pattern Mining

As mentioned before, one of the highlights of AT-Bi-LSTM is that it can automatically identify the malicious part of each traffic request according to attention weight vector. This is also the difference between this model and the traditional fingerprint extraction methods [14, 31]. As described in Sect. 4.2, we first cluster the malicious entries detected by AT-Bi-LSTM but not detected by the "Rule Engine", then we perform pattern mining for each cluster.

Given a cluster that consists of several traffic of cross-site scripting attack (see Fig. 7). We can get keywords for each entry according to its attention weight vector.

```
/cgi-bin/wa.exe?SHOWTPL=<script>alert(/openvas-xss-test/)</script>
/webContent/fck/wlkt.htm?xss_test"'></textarea></script><script>prompt(42873);</script>
adminDirHand="/></script><script>alert(1);</script>
itemid=1527"'></textarea></script><scr<script>ipt>alert(2014)</scr<script>ipt>
/survey/list.jsp?s_id=f65b</textarea></script><a href=//eye.hihop.cn/>webscan</a>
```

Fig. 7. Traffic samples of cross-site scripting attack.

For instance, the first traffic entry in Fig. 7 is “*/cgi-bin/wa.exe?<EOS> SHOWTPL=<script> alert(/openvas-xss-test/)</script>*” . The visualization of its attention vector is shown in Fig. 8. The color depth corresponds to the attention weight α_t (Eq. 3). The darker the color, the greater the weight value. Obviously, top 10 keywords for this entry are *{‘.’, ‘exe’, ‘<’, ‘script’, ‘>’, ‘alert’, ‘)’, ‘/’, ‘openvas’, ‘xss’}*. Based on this string pattern, we can generate a rule that identifies such malicious traffic.

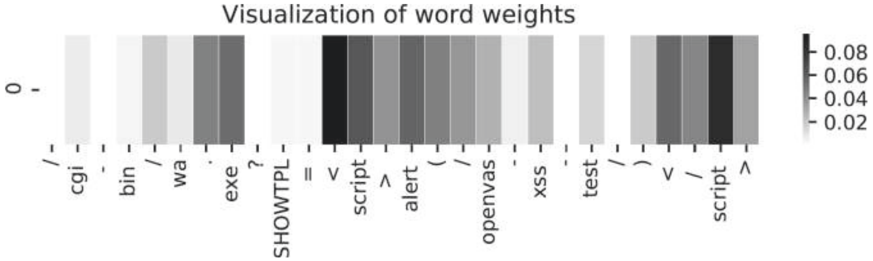


Fig. 8. Visualization of attention.

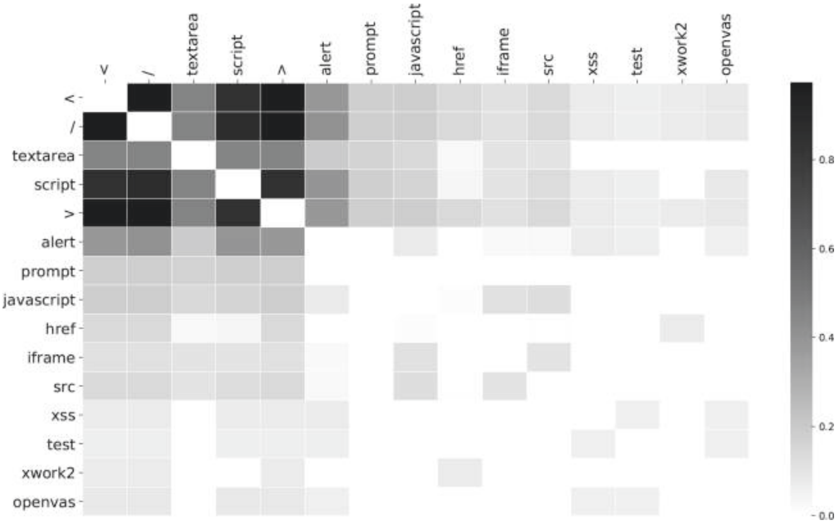


Fig. 9. Visualization of pattern mining.

To further illustrate the performance of the proposed model in malicious pattern mining, we visualize the pattern mining results of this cluster (see Fig. 9). The darker the color of the square is, the more times the words appear together. Hence, the pattern of these traffic can be denoted as {*“<”, “/”, “ script “, “>”, “textarea”, “prompt”, “javascript”, “alert”, “iframe”, “src”, “href”*}.

6 Conclusion

This paper presents DeepHTTP, a general-purpose framework for HTTP traffic anomaly detection and pattern mining based on deep neural networks. We build AT-Bi-LSTM, a deep neural networks model utilizing Bidirectional Long Short-Term Memory (Bi-LSTM), which can enable effective anomaly diagnosis. Besides, we design a novel method that can extract the structural characteristics of HTTP traffic. DeepHTTP learns content feature and structure feature of traffic automatically and unearths critical section of input data. It performs detection at the single traffic level and then performs pattern mining at the cluster level. The intermediate output including attention hidden state and the attentional weight vector can be applied to clustering and pattern mining, respectively. Meanwhile, by incorporating user feedback, DeepHTTP supports database updates and model iteration. Experiments on a large number of HTTP traffic entries have clearly demonstrated the superior effectiveness of DeepHTTP compared with previous methods.

Future works include but are not limited to incorporating other types of deep neural networks into DeepHTTP to test their efficiency. Besides, improving the ability of the model to detect unknown malicious traffic is something we need to further study in the future. With the increasing popularity of encrypted traffic, the detection of encrypted traffic attacks is also our future research direction.

Acknowledgement. This work was supported by the National Key R&D Program China under Grant 2018YFB0804701.

References

1. China Internet Network Security Report. https://www.cert.org.cn/publish/main/upload/File/2019-year.pdf
2. Estévez-Tapiador, J.M., García-Teodoro, P., et al.: Measuring normality in HTTP traffic for anomaly-based intrusion detection. Comput. Netw. **45**(2), 175–193 (2004)
3. Jamdagni, A., Tan, Z., Nanda, P., He, X., Liu, R.P.: Intrusion detection using GSAD model for HTTP traffic on web services. In: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, pp. 1193–1197. ACM (2010)
4. Tombini, E., Debar, H., Mé, L., Ducassé, M.: A serial combination of anomaly and misuse IDSes applied to HTTP traffic. In: Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC), pp. 428–437. IEEE Computer Society, Washington, DC, USA (2004)
5. w3techs Homepage. https://w3techs.com/technologies/details/ce-httpsdefault. Accessed 21 Jan 2020

6. Tony Messer's blog published in Pickaweb. https://www.pickaweb.co.uk/blog/local-business-seo-stats-chart-and-infographic/
7. Le, H., et al.: URLNet: learning a URL representation with deep learning for malicious URL detection. arXiv preprint arXiv:1802.03162 (2018)
8. Saxe, J., Berlin, K.: eXpose: a character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. arXiv preprint arXiv:1702.08568 (2017)
9. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Proceedings of INTERSPEECH, pp. 601–608 (2012)
10. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
11. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. SIGCOMM Comput. Commun. Rev. **35**(4), 217–228 (2005)
12. Samant, A., Adeli, H.: Feature extraction for traffic incident detection using wavelet transform and linear discriminant analysis. Comput.-Aided Civil Infrastruct. Eng. **15**(4), 241–250 (2010)
13. Swarnkar, M., Hubballi, N.: OCPAD: one class Naive Bayes classifier for payload based anomaly detection. Expert Syst. Appl. **64**, 330–339 (2016)
14. Nelms, T., Perdisci, R., Ahamad, M.: Execscent: mining for new C&C domains in live networks with adaptive control protocol templates. Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 2013), pp. 589–604 (2013)
15. Ma, J., Saul, L.K., Savage, S., et al.: Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1245–1254. ACM (2009)
16. Ma, J., Saul, L.K., Savage, S., et al.: Identifying suspicious URLs: an application of large-scale online learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 681–688. ACM (2009)
17. Bortolameotti, R., van Ede, T., et al.: DECANTeR: DEteCtion of anomalous outbouNd HTTP TRaffic by passive application fingerprinting. In: Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC 2017), pp. 373–386. ACM, New York (2017)
18. Juvonen, A., Sipola, T., Hämäläinen, T.: Online anomaly detection using dimensionality reduction techniques for HTTP log analysis. Comput. Netw. **91**, 46–56 (2015)
19. Ringberg, H., Soule, A., et al.: Sensitivity of PCA for traffic anomaly detection. SIGMETRICS Perform. Eval. Rev. **35**(1), 109–120 (2007)
20. El-Alfy, E.S.M., Al-Obeidat, F.N.: A multicriterion fuzzy classification method with greedy attribute selection for anomaly-based intrusion detection. Procedia Comput. Sci. **34**, 55–62 (2014)
21. Estévez-Tapiador, J.M., et al.: Measuring normality in HTTP traffic for anomaly-based intrusion detection. Comput. Netw. **45**(2), 175–193 (2004)
22. Mahoney, M.V., Chan, P.K.: Learning rules for anomaly detection of hostile network traffic. In: Third IEEE International Conference on Data Mining, pp. 601–604 (2003)
23. Radford, B.J., Apolonio, L.M., et al.: Network Traffic Anomaly Detection Using Recurrent Neural Networks. CoRR abs/1803.10769 (2018)
24. Du, M., Li, F., et al.: DeepLog: anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS 2017), pp. 1285–1298. ACM, New York (2017)
25. Erfani, S.M., et al.: High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recogn. **58**, 121–134 (2016)
26. Chiou, T.-W., Tsai, S.-C., Lin, Y.-B.: Network security management with traffic pattern clustering. Soft Comput. **18**(9), 1757–1770 (2014)

27. Le, T.T., Millaire, A., Asseman, P., De, G.P., ThĂČAĬry, C., Ducloux, G.: Novel approach for network traffic pattern analysis using clustering based collective anomaly detection. Ann. Data Sci. **2**(1), 111–130 (2015)
28. Ahmed, M., Mahmood, A.N.: Network traffic pattern analysis using improved information theoretic co-clustering based collective anomaly detection. In: Tian, J., Jing, J., Srivatsa, M. (eds.) SecureComm 2014. LNICST, vol. 153, pp. 204–219. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23802-9_17
29. Ren, J.T., Ou, X.L., Zhang, Y., Hu, D.C.: Research on network-level traffic pattern recognition. In: Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, pp. 500–504 (2002)
30. Paredes-Oliva, I., Castell-Uroz, I., Barlet-Ros, P., Dimitropoulos, X., SolAĬ-Pareta, J.: Practical anomaly detection based on classifying frequent traffic patterns. In: 2012 Proceedings IEEE INFOCOM Workshops, pp. 49–54 (2012)
31. Rafique, M.Z., Caballero, J.: FIRMA: malware clustering and network signature generation with mixed network behaviors. In: Stolfo, S.J., Stavrou, A., Wright, C.V. (eds.) RAID 2013. LNCS, vol. 8145, pp. 144–163. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41284-4_8
32. Nelms, T., Perdisci, R., et al.: ExecScent: mining for new C&C domains in live networks with adaptive control protocol templates. In: Presented as part of the 22nd USENIX Security Symposium (2013), pp. 589–604. USENIX, Washington, D.C. (2013)
33. Xu, K., Ba, J., Kiros, R., et al.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057 (2015)
34. Chorowski, J.K., Bahdanau, D., et al.: Attention-based models for speech recognition. In: Advances in Neural Information Processing Systems, pp. 577–585 (2015)
35. Ma, F., Chitta, R., Zhou, J., et al.: Dipole: diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1903–1911. ACM (2017)
36. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
37. Zhou, P., Shi, W., Tian, J., et al.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (vol. 2: Short Papers), pp. 207–212 (2016)
38. Ren, P., Chen, Z., Ren, Z., et al.: Leveraging contextual sentence relations for extractive summarization using a neural attention model. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 95–104 (2017)
39. Zhang, H., et al.: Webshell traffic detection with character-level features based on deep learning. IEEE Access **6**, 75268–75277 (2018)
40. Kali Official Website. https://www.kali.org/. Accessed 21 Jan 2020
41. Chinotec Technologies Company: Paros - for web application security assessment. http://www.parosproxy.org/index.shtml. Accessed 21 Jan 2020
42. Riancho, A.: Web Application Attack and Audit Framework. http://w3af.sourceforge.net. Accessed 21 Jan 2020
43. Damashek, M.: Gauging similarity with n-grams: language-independent categorization of text. Science **267**(5199), 843–848 (1995)
44. Kloft, M., Brefeld, U., Düessel, P., Gehl, C., Laskov, P.: Automatic feature selection for anomaly detection. In: Proceedings of the 1st ACM Workshop on Workshop on AISec (AISec 2008), pp. 71–76. ACM, New York (2008)
45. Wang, K., Stolfo, S.J.: Anomalous payload-based network intrusion detection. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 203–222. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30143-1_11

46. Zolotukhin, M., HAďmAďlAďinen, T., Kokkonen, T., Siltanen, J.: Analysis of HTTP requests for anomaly detection of web attacks. In: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, pp. 406–411 (2014)
47. hashID, a tool written in Python 3. https://github.com/psypanda/hashID
48. Joachims, T.: A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In: Proceedings of International Conference on Machine Learning, pp. 143–151 (1996)
49. Ester, M., Kriegel, H.P., Sander, J., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, no. 34, pp. 226–231 (1996)
50. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning (2014)
51. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
52. Ho, T.K.: Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1. IEEE (1995)
53. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2016)
54. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural-network approach. IEEE Trans. Neural Netw. **8**(1), 98–113 (1997)
55. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
56. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
57. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
58. Lakhina, A., Crovella, M., Diot, C.: Characterization of network-wide anomalies in traffic flows. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC 2004), pp. 201–206. ACM, New York (2004)
59. Wang, K., Parekh, J.J., Stolfo, S.J.: Anagram: a content anomaly detector resistant to mimicry attack. In: Zamboni, D., Kruegel, C. (eds.) RAID 2006. LNCS, vol. 4219, pp. 226–248. Springer, Heidelberg (2006). https://doi.org/10.1007/11856214_12
60. Iglesias, F., Zseby, T.: Analysis of network traffic features for anomaly detection. Mach. Learn. **101**(1–3), 59–84 (2015)

Social Network Security and Privacy

A Label Propagation Based User Locations Prediction Algorithm in Social Network

Huan Ma[1] and Wei Wang[2](✉)

[1] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
[2] Harbin Engineering University, No. 145, Nantong Street, Harbin, China
w_wei@hrbeu.edu.cn

Abstract. Network community detection is an important service provided by social networks, and social network user location can greatly improve the quality of community detection. Label propagation is one of the main methods to realize the user location prediction. The traditional label propagation algorithm has the problems including "location label countercurrent" and the update randomness of node location label, which seriously affects the accuracy of user location prediction. In this paper, a new location prediction algorithm for social networks based on improved label propagation algorithm is proposed. By computing the K-hop public neighbor of any two point in the social network graph, the nodes with the maximal similarity and their K-hopping neighbors are merged to constitute the initial label propagation set. The degree of nodes not in the initial set are calculated. The node location labels are updated asynchronously is adopted during the iterative process, and the node with the largest degree is selected to update the location label. The improvement proposed solves the "location label countercurrent" and reduces location label updating randomness. The experimental results show that the proposed algorithm improves the accuracy of position prediction and reduces the time cost compared with the traditional algorithms.

Keywords: Social network · Location prediction · Label propagation · Social relationships · User location probability

1 Introduction

As social networks with location-based information are increasingly popular, the users' location in social network attracts more attention than before. Location information can help to shorten the gap between the virtual and the real world, such as monitoring residents' public health problems through online network [1], recommending local activities or attractions to tourists [2, 3], determining the emergency situation and even the location of the disaster and so on [4–6]. In addition, users' offline activity area and trajectory can also be analyzed through their locations in social networks. Due to the increasing awareness of privacy protection, people will cautiously submit their personal location information or set the visibility of the location of the message in social networks,

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 165–174, 2020.
https://doi.org/10.1007/978-981-33-4922-3_12

which make it difficult to acquire their real location information. Therefore, how to accurately predict the actual location information of social network users is an important and meaningful research question.

This paper proposes location prediction algorithm for social network users based on label propagation, which solves the following two key problems:

(1) The accuracy of the traditional label propagation algorithm is not high in the user location prediction, and "countercurrent" phenomenon will appear in the iterative process, which will lead to the increase of the time overhead.
(2) Improve the accuracy of social network users' location prediction by using their offline activity location.

2 Related Work

There are three scenarios for user location prediction in social networks, such as user's frequent location prediction, prediction of the location of messages posted on the user's social network, and forecasts of the locations mentioned in messages. The main methods of location pre-diction include location prediction based on the content of message published by users, user friend relationships, and so on.

Laere et al. chose two types of local vocabulary and extracted valid words to predict the location of users [7]. Ren [8] and Han et al. [9] were inspired by the frequency of reverse documents, using the reverse position frequency (ILF) and the reverse city frequency (ICF) to select the position of the vocabulary, they assumed that the location vocabulary should be distributed in fewer locations, but with large ILF and ICF values. Mahmud et al. [10] applied some column heuristics to select local vocabulary. Cheng [1] makes the position word distribution conform to the spatial change model proposed by the Backstorm [11], secondly they make local or non-local mark on 19,178 dictionary words, and use the Labeled Vocabulary Training classification model to discriminate all words in the tweet dataset.

Backstrom [12] established probability models through physical distances between users to express the possibility of relationships between users, which has no effect on the position prediction of friends considering different degrees of tightness. Kongl [13] on the basis of Backstorm work by adding the weight of the edge to predict the user's position, where the weight of the edge is determined by a social tight coefficient. Li [14] considered the location of user neighbors, and captures the information of users' neighbors that intuitively consider the location of users. The user location is allocated randomly, then the user's location is iteratively updated from the user's neighbors and the location name mentioned, and then the parameters in the update are improved by measuring the prediction error of the known location of the user. Davis Jr et al. [15] thought that the most frequent user' locations that appear in the user's social network as a basis for predicting their location. Jurgens et al. [16] extend the concept of location prediction into location label propagation, which is made by the location of the label space to explain the location of label propagation, they think that the position of the user through the iterative process that many times.

Li et al. [17] thought that the literature assume the user has only one home location is a defect, they think that users should have the relationship with a number of positions, so they have defined the location information of a user and user set as the set of locations, and these users about the system is not only a geographical location the range is not a point, is not a temporary and user related position, but a long-term position, so they set up a MLP in the paper (Multiple Location Profiling Model) to establish a model containing a plurality of position information of the position of archives to the user, and this model is to the location file according to the target user relationships and their tweets content released.

The label propagation algorithm can effectively deal with large data sets, so in this paper, we are in the position of the user prediction based on label propagation algorithm, but with the label propagation algorithm in-depth study, we found that the label propagation algorithm will position the label "countercurrent" label update and node location is random, this algorithm cannot guarantee the accuracy of prediction of the position of the user, in order to improve the accuracy of location prediction algorithm and reduce the time overhead, this paper pro-poses a label propagation based on user location prediction algorithm (Label Propagation Algorithm-Location Prediction, LPA-LP).

3 Related Concept and Problem Definition

Definition 1 Social Network**.** A social network can be represent by a graph $G = (V, E, A)$, where V represents the collection of the users who are in the social network, and $n = | V |$. E represents the collection of the relationship between users and $m = |E|$, and A represents the collection of the activities and $a = |A|$. Beyond that, L represents the set of locations, including users' locations and activities' locations, and $n_l = |L|$, U_0 is the set of the users whose locations are known, on the contrary, U_n is the set of users whose locations are unknown.

Definition 2 Shortest Path Length**.** It refers to the shortest path between the two nodes i and j in the social network graph. It means the minimum number of paths through the node i to the node j. It can be used $d(i, j)$ to represent the shortest path length between two nodes.

Definition 3 K-Hopping Neighbors**.** It means that the user to its neighbor needs a k hopping to achieve, that is to say, the shortest path length of the two node is k.

Definition 4 K-Hopping Public Neighbors**.** $G = (V, E, A)$ is a social network diagram, where V represents the user set in the graph, $E = (v_i, v_j, w_{ij})$ represents the set of relations between the user nodes with weights, w_{ij} represents the weight of the edges between nodes. The k-hopping public neighbors set of the nodes is defined as follows:

$$\Gamma_k\left(v_i, v_j\right) = \left\{v \middle| d(v_i, v) = d\left(v_j, v\right) = k\right\} \ = \Gamma(v_i, k) \cap \Gamma\left(v_j, k\right), k \geq 1 \qquad (1)$$

In the formula (1), $\Gamma(v_i, k)$ represents the set of k-public neighbor of node v_i, and represents the set of node v_j, represents the set of k-public neighbor between v_i and v_j.

Definition 5. Similarity of k-hopping public neighbors. The value of k is determined by the network itself, it can be defined on formula (2).

$$\bar{k} = \frac{\sum_{i \neq j} k_{\max} |\Gamma(i) \cap \Gamma(j)|}{|V|} \tag{2}$$

In the formula (2), $k_{\max}|\Gamma(i) \cap \Gamma(j)|$ represents the max public neighbor hops between two nodes. The k value in the network refers to the average of any two nodes in the network. The similarity of the two node k-hopping public neighbors is defined by formula (3).

$$S(v_i, v_j) = \frac{|\Gamma(v_i, k) \cap \Gamma(v_j, k)|}{|\Gamma(v_i, k) \cup \Gamma(v_j, k)|}, k \geq 1 \tag{3}$$

Definition 6 Similarity of Nodes. It means denominator size of the similarity between the k-hopping public neighbors between nodes subtracts the two nodes. It can be defined by formula (4).

$$\gamma = \frac{|\Gamma(v_i, k) \cap \Gamma(v_j, k)|}{|\Gamma(v_i, k) \cup \Gamma(v_j, k)| - 2}, k \geq 1 \tag{4}$$

Definition 7. The max degree between nodes and users set. If the user is divided into different sets $L_1, L_2, \ldots, L_e$ according to their locations, nodes are set up by users who are not labeled as location labels. The max degree of users divided into different sets according to their location is the degree and the maximum of some nodes in the nodes. It can be defined by formula (5).

$$d(v_i, L_i) = \max\{d(v_i, L_1), d(v_i, L_2), \ldots, d(v_i, L_e)\} \tag{5}$$

Definition 8 K-Hopping Weight**.** We believe that the most important impact on user location is its 1 hop neighbors. Moreover, the offline location of users also has a great impact on user location, and its weight can also be set to 1. For $k > 1$, when setting the weight of the edge, it will be attenuated according to the speed of 1/5, that is, the weight of the edge of the 1 hop neighbor is 1, the weight of the 2 neighbors is 0.8, and so on.

Now given the location prediction problem definition: In the social network G, the unknown location information of the user u, according to the location information and the users of their k-hopping neighbors, to predict the unknown location information of the user u in the prediction of the probability of the position of L.

4 Label Propagation Based User Location Prediction Algorithm

In this section, a correlation algorithm for location prediction for users of unknown location information in social networks is proposed. This paper proposed a location prediction algorithm based on label propagation (Label Propagation Algorithm-Location Prediction, LPA-LP), the algorithm is mainly divided into two parts, one part is to run before the label propagation algorithm of data preprocessing algorithm, the other part is the use of label propagation of location prediction algorithm.

Algorithm 1Data Preprocessing

```
Input: G-dataset, L-location label set
Output: pre-processed user sets C
Initialization: set C is empty
while $S(v_i, v_j) \geq \gamma$ do
for each $v \in U_n$ do
if $\Gamma(v) \neq \varnothing$ do
  set k = 1
$\Gamma(v)_i \leftarrow BFS(v)$
k = i
end for
    get $\Gamma_k(v_i, v_j)$ for every nodes based on formula(1)
    select all nodes with $k_{max}$
    calculate the similarity of k-neighbors based on formula(3)
    choose the k-neighbors as the begin set
    update the nodes' label in the begin set
    calculate node similarity based on formula(4)
   end while
```

Algorithm 1 is pretreated before running the label propagation algorithm to initialize the data set, according to the Definition 5, the node with its maximum similarity and the *k* hop neighbor as the set of starting processing for the user location prediction, and according to the known label to the data in the collection of the label, which is in order to be able to quickly and accurately using the label propagation algorithm for unknown location information in a social network user node location prediction. After preprocessing the data set, location prediction algorithm based on label propagation can be used to predict the location of users who have not tagged location labels in the processed data set. Algorithm 2 gives a description of the location prediction algorithm (LPA-LP) based on the label propagation.

Algorithm2LPA-LPalgorithm description

```
Input:pre-processed user sets $C_i$
Output:estimated user location, E
Initialization:set E is empty
    choose the begin set $C_i$
    while the nodes label change do
      for each node do
        sort the nodes as the k to every set Ci to made F
          end for
      for $v \in F$ do
$F_v(t) = f\left(F_{v_{i1}}(t),\ldots,F_{v_{im}}(t),F_{v_{im+1}}(t-1),\ldots,F_{v_{in}}(t-1)\right)$
    update the nodes' label in the begin set build E'
    end for
      E=E'
end while
```

In Algorithm 2 location prediction algorithm based on label propagation in the iterative process of user location labels are updated, and the location information of the user location information of neighbors and user participation in the offline activities are taken into account, which significantly improves the prediction accuracy of the locations of users, and in the operation of label propagation algorithm for data sets are preliminary the treatment improve the performance of the label propagation algorithm of user location prediction algorithm, the following will be proved by experiments.

5 Experiment Result and Analysis

In this section, we will analyze the experimental results, the experimental results are divided into two parts, one part is the results of algorithm time overhead and the other is the accuracy of user locations prediction algorithm.

5.1 Data Set Description

In this paper, we use the dataset is NLPIR microblogging corpus. We extracted several datasets from the dataset. In order to compare the accuracy of the improved algorithm for user location prediction and improve the execution efficiency of the algorithm, we extract different scale datasets from the data set to compare the experimental results. The detail of our data sets are described in Table 1.

5.2 Experimental Results Analysis

The location prediction algorithm based on the label propagation (LPA-LP) is an improvement on the preprocessing of the data set and the selection strategy of the location label in the iterative process. It can avoid the "countercurrent" phenomenon of the position label and reduce the randomness to update the location tag, and improve the efficiency and the accuracy of the prediction. The whole experiment is divided into two

Table 1. Data sets description

Dataset	Users number	Relations number	Activities number
A	2748	12121	452
B	4025	61881	983
C	5194	77567	2392
D	9940	107979	4938

parts. The first part is using label propagation algorithm to predict user location on these four datasets of different sizes. The second part is using LPA-LP algorithm to predict location on four different scale datasets.

In the process of user location prediction, probabilistic LPA algorithm and LPA-LP algorithm with random or update the node label to a certain extent, the running times of the two algorithms may produce different results, so the choice between the four data sets of different size on the running times of experimental results for the 10, 30, 50, 70, 100 and mean value. The time required for the experiment to run on different scale data sets is shown in Fig. 1, 2, 3 and 4.

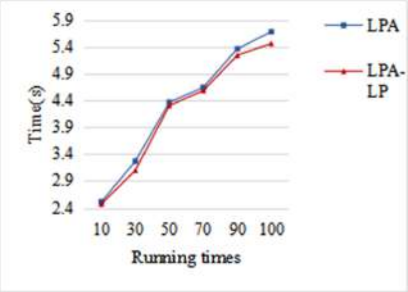


Fig. 1. Time overhead comparison with dataset A

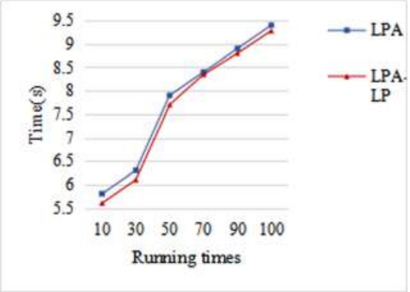


Fig. 2. Time overhead comparison with dataset B

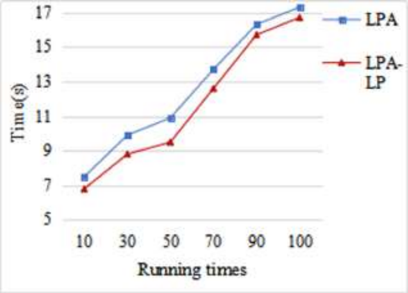


Fig. 3. Time overhead comparison with dataset C

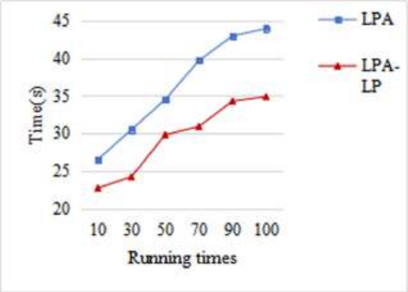


Fig. 4. Time overhead comparison with dataset D

From these four figures, we can know that the running time of different dataset is similar between the improved algorithm LPA-LP and the algorithm LPA when the dataset have less than 5000 nodes, when the nodes are more than 9000 in dataset, we can see that the running time of the improved algorithm LPA-LP is obviously less than the algorithm LPA. It shows that the LPA-LP algorithm can be effectively applied to large-scale data sets.

In addition to comparing the running time of the algorithm, it is necessary to compare the accuracy of the algorithm. The results of the experiment are shown in Table 2.

Table 2. Algorithm accuracy comparison

Dataset	Accuracy	
	LPA	LPA-LP
A	59.3%	64.4%
B	62.2%	67.3%
C	66.5%	69.5%
D	70.7%	78.4%

6 Conclusion

This paper proposes a location prediction algorithm for social network users based on label propagation. The algorithm first obtains k-hop public neighbors at any two points in the social network graph, and uses the node with the largest similarity and its k-hop neighbors as the initial set of label propagation, and calculates the degree of the node to these sets. In each iteration, the node adopts the strategy of asynchronous update, and selects the node with the highest degree to update the position label, so as to avoid the "countercurrent" phenomenon of the position label and reduce the possibility of randomly updating the position label. Relevant experiments show that the algorithm proposed in this paper improves the accuracy of user location prediction and reduces the time cost of the algorithm.

References

1. Cheng, Z., Caverlee, J., Lee, K.: You are where you tweet: a content-based approach to geo-locating Twitter users. In: The 19th ACM Conference on Information and Knowledge Management, pp. 759–768. ACM, Toronto (2010)
2. Yuan, Q., Cong, G., Ma, Z., et al.: Who, where, when and what: discover spatio-temporal topics for Twitter users. In: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 605–613. ACM, Chicago (2013)
3. Noulas, A., Scellato, S., Lathia, N., et al.: Mining user mobility features for next place prediction in location-based services. In: 13th Industrial Conference on Data Mining, pp. 1038–1043, IEEE, New York (2013)
4. Rakesh, V., Reddy, C.K., Singh, D., et al.: Location-specific tweet detection and topic summarization in Twitter. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1441–1444. ACM, Niagara (2013)
5. Ao, J., Zhang, P., Cao, Y.: Estimating the locations of emergency events from Twitter streams. Procedia Comput. Sci. **31**, 731–739 (2014)
6. Lingad, J., Karimi, S., Yin, J.: Location extraction from disaster-related microblogs. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1017–1020. ACM, Rio de Janeiro (2013)
7. Van Laere, O., Quinn, J., Schockaert, S., et al.: Spatially aware term selection for geotagging. IEEE Trans. Knowl. Data Eng. **26**(1), 221–234 (2014)
8. Ren, K., Zhang, S., Lin, H.: Where are you settling down: geo-locating Twitter users based on tweets and social networks. In: Hou, Y., Nie, J.-Y., Sun, L., Wang, B., Zhang, P. (eds.) AIRS 2012. LNCS, vol. 7675, pp. 150–161. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35341-3_13
9. Han, B., Cook, P., Baldwin, T.: Geolocation prediction in social media data by finding location indicative words. In: 24th International Conference on Computational Linguistics, pp. 1045–1062. ACM, Mumbai (2012)
10. Mahmud, J., Nichols, J., Drews, C.: Where is this tweet from? Inferring home locations of Twitter users. In: Sixth International AAAI Conference on Weblogs and Social Media, pp. 73–77. AAAI, Dublin (2012)
11. Backstrom, L., Kleinberg, J., Kumar, R., et al.: Spatial variation in search engine queries. In: Proceedings of the 17th International Conference on World Wide Web, pp. 357–366. ACM, Beijing (2008)

12. Backstrom, L., Sun, E., Marlow, C.: Find me if you can: improving geographical prediction with social and spatial proximity. In: Proceedings of the 19th International Conference on World Wide Web, pp. 61–70. ACM, North Carolina (2010)
13. Kong, L., Liu, Z., Huang, Y.: SPOT: locating social media users based on social network context. Proc. VLDB Endow. **7**(13), 1681–1684 (2014)
14. Li, R., Wang, S., Deng, H., Wang, R., Chang, K.C.: Towards social user profiling: unified and discriminative influence model for inferring home locations. In: The 18th International ACM SIGKDD Conference, pp. 1023–1031. ACM, Beijing (2012)
15. Davis, Jr C., Pappa, G., de Oliveira, D., de L Arcanjo, F.: Inferring the location of twitter messages based on user relationships. Trans. GIS **15**(6), 735–751 (2011)
16. Jurgens, D.: That's what friends are for: inferring location in online social media platforms based on social relationships. In: Seventh International AAAI Conference on Weblogs and Social Media, pp. 237–240. AAAI, Massachusetts (2013)
17. Li, R., Wang, S., Chang, C.: Multiple location profiling for users and relationships from social network and content. Proc. VLDB Endow. **5**(11), 1603–1614 (2012)

Perosonalized Differentially Private Location Collection Method with Adaptive GPS Discretization

Huichuan Liu, Yong Zeng(✉), Jiale Liu, Zhihong Liu, Jianfeng Ma, and Xiaoyan Zhu

Xidian University, Xi'an 710126, Shaanxi, China
hc_liu@stu.xidian.edu.cn, {yzeng,liuzhihong,jfma,
xyzhu}@mail.xidian.edu.cn, liujialehenu@163.com

Abstract. In recent years, with the development of mobile terminals, geographic location has attracted the attention of many researchers because of its convenience in collection and its ability to reflect user profile. To protect user privacy, researchers have adopted local differential privacy in data collection process. However, most existing methods assume that location has already been discretized, which we found, if not done carefully, may introduces huge noise, lowering collected result utility. Thus in this paper, we design a differentially private location division module that could automatically discretize locations according to access density of each region. However, as the size of discretized regions may be large, if directly applying existing local differential privacy based attribute method, the overall utility of collected results may be completely destroyed. Thus, we further improve the optimized binary local hash method, based on personalized differential privacy, to collect user visit frequency of each discretized region. This solution improve the accuracy of the collected results while satisfying the privacy of the user's geographic location. Through experiments on synthetic and real data sets, this paper proves that the proposed method achieves higher accuracy than the best known method under the same privacy budget.

Keywords: Local differential privacy · Geographical location · Privacy security

1 Introduction

With the development of mobile Internet technology, various mobile platforms such as mobile phones, tablets, smart watches and other devices have brought many conveniences and joys to people's lives. Sensors such as Accelerometer, GPS, Gyroscope and Magnetometer could capture information about the user's surroundings and provide a richer and more interesting interface for human-computer interaction. Among them, geographic location sensing has been widely equipped on smart devices. As a form of information that could reflect the user's trajectory and lifestyle, it is widely used by major application service providers in the recommendation system to provide users with personalized advertisement.

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 175–190, 2020.
https://doi.org/10.1007/978-981-33-4922-3_13

However, due to the sensitivity of the geographic location itself, and the fact that background applications may collect user data at any time, the uploaded user trajectory data may reflect the user's sensitive information, such as the user's income, beliefs, daily habits, illness and other information [1]. Users may dislike their private data that could expose their activity being analyzed. Besides that, improper data management may result in the disclosure of user privacy data, thereby causing legal problems.

In order to ensure privacy of user uploaded data in analysis process, many researches have been conducted and most differential privacy based methods for solving privately analysis can mainly be divided into two categories. The first category [2–6] is to disturb the collected data before data sharing and publishing. This type mainly uses differential privacy settings. The other category [7–9] mainly focuses on the data collection process and disturbs the data before users upload their private data. Among them, the former category couldn't provide protection against database intrusions or application service providers' threats to user privacy. In reality, the database interface provided by the server is very likely to have problems. For example, in March 2018, a security breach on Facebook enables third-party application software to download unpublished private photos of users without permission, affecting up to 6.8 million users. It is conceivable that with the expansion of business and the growth of code volume, security vulnerabilities are inevitable. The privacy protection of the second category, which is based local differential privacy model, can also essentially prevent third-party analysts from threatening privacy, and it can also prevent the inappropriate use of user privacy data by the enterprise itself, so it has a stronger privacy protection. In this paper, we follow the second category research line and adopt a variant of local differential privacy as our privacy model.

Most existing attribute collection methods [10–12] assume that the user attributes to be collected are discrete, which means, for GPS data, the continuous GPS signal must be quantified before being applied to an existing collection method. But in fact, due to the non-uniformity of the geographical location itself, completely uniform density quantization without any knowledge of the whole user density distribution, will cause very low signal-to-noise ratio. In addition, in order to provide more fine-grained geographic location collection, the number of quantized geographic location areas is large, so local differential privacy based location collection methods would cause overwhelming noise, completely destroying the utility of the data collection results.

This paper proposes a new geographic location collection method. The method is divided into two modules, each of which takes exclusive user sets as input. The first module is a location division module, which is responsible for sending location-related query requests to users in the corresponding user set. On the premise of localized differential privacy, the location area is divided, in the form of quadtree, to establish a quantitative level of location. The second module is the location collection module. It collected the its users' disturbed location set on the division results of the first module, and estimate the true user location distribution as the final result. The main innovations of our method are as follows:

Adaptive location discretization. Unlike the previous work, the method in this paper does not need to assume that the input geographical location are discrete. We propose a local differential privacy based method that can interactively make queries to users and could adaptively discretize the GPS data according to the user access density of each region. This module divides the area as finely as possible while ensuring the signal-to-noise ratio of the collected result, which balances the grainess of region and signal-to-noise ratio.

Adoption of personalized differential privacy. In our experiments, we found that the geographic location collection scheme that conforms to local differential privacy introduces a lot of noise and makes the overall utility of the final collection results low. Therefore, we adopt the personalized local differential privacy model and modified existing attribute collection algorithms, achieving collection result with higher utility.

2 Related Work

Since local differential privacy needs to disturb user data before the user uploads the data, a mechanism that conforms to local differential privacy generally runs on the user side. Local differential privacy will disturb each user's data, and the variance of the noise of the aggregate result is proportional to the number of samples. In order to avoid noise overwhelming the real signal results, the data collection method that conforms to local differential privacy will only count the frequency of frequent item sets. In order to reduce the impact of noise on the data collection process, and to optimize the communication overhead and computational efficiency, researchers have conducted a lot of researches on the implementation of data collection mechanisms that conform to local differential privacy. Here we briefly introduce the design of methods that have inspired our work.

In 2014, a statistical method RAPPOR that conforms to local differential privacy is proposed. This method encodes the user's attribute set through the bloom filter and randomly disturbs all bits of the bloom filter. On the basis of local differential privacy, the disturbed bloom filter is uploaded to the data collector. On the collector side, the collector sums the set times of all bits of the bloom filter uploaded by all users, and use the least square method to estimate the frequency of occurrence of each attribute. In 2016, RAPPOR [8] was further improved, no longer need to assume that user attributes belong to a known limited set, so that RAPPOR can count the frequency of frequent occurrences of arbitrary unknown attributes. Their improved method is comprised of two modules. The first module is the same as the original RAPPOR method, using bloom filter results to estimate the frequency of attributes. The second module is used to calculate attribute sets that belong to frequent items. It cuts the string encoding of all attribute names into multiple fixed-length character segments, and uses the expected maximum algorithm to estimate the probability of occurrence of all character segment pairs. The connection of the character combination is stored in a graph. Each character segment corresponds to a node in the graph. When the occurrence probability of the character segment pair exceeds a certain threshold, the two nodes are connected. Since all character segments of each frequent element must also be frequent, fully connected subgraphs of a specific length in the graph then correspond to frequent item sets. Finally, the first module could estimate the frequency of items in the frequent attribute set.

In 2015, a local differential privacy based method—binary local hashing method [9] is proposed, which is completely different from RAPPOR and based on the principle of compressed sensing theory. This method randomly generates a ±1 vector with a fixed length of m for each attribute of the user attribute set, and uses this vector as the binary representation of the attribute. Since the expectation of two different vector dot product is 0, and the dot product of the same vector is m, the method randomizes the input vector while keeping the expectation of each value in the vector unchanged, and then sums all the uploaded user vector And by dot multiplying the sum vector with any representation vector of an attribute, we can get an unbiased estimate of the frequency of the attribute.

In 2017, researchers [10] summarized methods such as random response, RAPPOR, and binary local hash method, and proposed an error analysis framework for automatically optimizing random response probability parameters. But these two methods can only estimate the attribute frequency of a known and limited set, and cannot deal with the unknown or unlimited number of attribute sets.

In 2018, a frequent item set discovery framework, called PrivTrie [11], based on prefix trees was proposed. They believed that the reason RAPPOR improved method [8] has excessive computational overhead and sensitivity to noise interference, is that graph is not suitable for storing the relationship between character segments. Therefore, they propose to use the prefix tree structure to describe the coupling relationship between character segments. In addition, their paper proposes a method that can make the same query to users of different branches of the prefix tree at the same time and still ensure differential privacy security. It can make more query requests to a limited set of users, thereby improving the accuracy of estimated attribute frequency.

In addition, in 2016, researchers [12] first applied the concept of local differential privacy to the field of geographic location collection research, and its scheme adopted a binary local hash method for location data collection. As the direct use of localized differential privacy would result in low signal-to-noise ratio, researchers proposed the concept of personalized local differential privacy, which is different from local differential privacy in that the new concept only requires that the probability distribution on the user-specified attributes are approximate rather than the whole attribute set. In addition, the scheme assumes that all geographic locations have been quantified as discrete areas. This scheme is a geographic location collection scheme based on the concept of local differential privacy derivation, which is known to have high data utility. Therefore, we use this work as a comparison to verify the utility of the data collection results of our work, and in paper, we refer to it as PSDA.

3 System Overview

In order to guarantee the user's data privacy during data collection, our method adopts the local differential privacy [13] as the privacy protection model. The principle of localized differential privacy is to randomly disturb the user's data before uploading it. After the collector collects a certain number of users' disturbed data, the collector then estimates the distribution of real users. There are mainly two problems in the scheme design:

(1) Suppose the size of the user set to be collected is N, the noise magnitude added by local differential privacy is orders of, and the noise added by centralized differential privacy is generally a constant. Therefore, compared to centralized differential privacy based method, data collection methods that conform to local differential privacy need to be designed to ensure that the attribute whose frequency is to be estimated must be frequent. As a result, before estimating the frequency of geographic

location access, our method first needs to calculate the frequent item sets, and the process of calculating frequent item sets also needs to satisfy the local differential privacy.

(2) There are huge differences in user attitudes towards privacy. On the one hand, capturing this difference meets the personalized privacy requirement; on the other hand, it adaptively reduces the magnitude of added noise. Therefore, in our method, it is necessary to adopt a privacy concept that can reflect the privacy protection needs of different users according to the characteristics of geographic location data, so as to improve the availability of data.

In response to the problem in (1), our method first divides the user set into two disjoint set, the first set is used to calculate frequent itemsets of geographic location. As original GPS data is continuous, and there is a certain unevenness in the distribution, so first of all, it is necessary to quantify the continuous geographic location into discrete areas, and adjust the quantization granularity of different areas according to each area's user access frequency. More fine-grained quantification need to be performed on the area with higher user access frequency; the second user set is used to collect the disturbed frequency of user visits in each geographical area, and estimate the true geographic distribution of users.

In response to the problem in (2), our method adopts the concept of personalized local differential privacy, using the tree structure to organize the calculated frequent area sets, and allows users to personalize their privacy requirement, which can greatly improve the accuracy of the estimation result.

In terms of system architecture, this chapter is divided into a geographic location division module and a geographic location collection module. The relationship between these two modules is shown in Fig. 1.

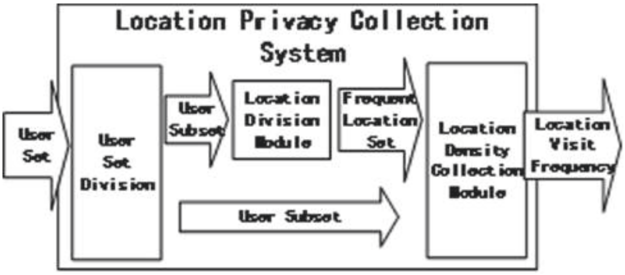


Fig. 1. Architecture of our location privacy collection method

4 Module Design

4.1 Location Division Module

This section introduces the design of the geographical location division module. The map division method used in our method uses the quadtree division method adopted by previous researchers [4, 14, 15], and the division method is shown in Fig. 2. The largest square represents the entire map. By recursively dividing the map with a quadtree, multiple quantization areas are obtained. In the location density collection module, the results of the map division will be used to quantify the continuous geographic location of the user, and then the data collection method for discrete attributes can be adopted.

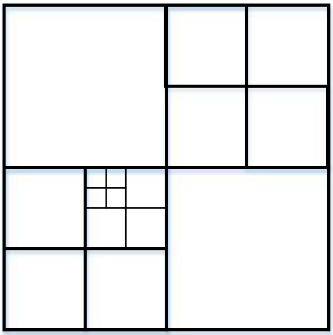


Fig. 2. Schematic diagram of geographical location division method based on quadtree

Because the local differential privacy based method can only collect the frequency of frequent itemsets, it is necessary to ensure that the frequency of user access in each sub-region finally obtained is higher than a certain threshold to reduce the impact of noise. Therefore, the problems solved in this section are summarized as follows: Under the limitation of local differential privacy, the map is reasonably segmented using a limited set of user data, so that the number of users in each sub-region is higher than a certain threshold and as close as possible to the threshold.

Before introducing the scheme, first we introduce the data structure used in the algorithm. The TreeNode structure is used to store tree node information, where Cell represents the area corresponding to the tree node, children represents the child nodes of the tree node, number represents the number of users who select the node as a geographic protection area. As our Location Density Module exploits personalized differential privacy, user_index is used to store the set of users who designate this TreeNode as their privacy protection area. Count is used to save the weights of the four child nodes of the node, and parent represents the parent node of the node.

```
struct TreeNode {
    Cell c
    TreeNode* [] children
    int number
    int[] users_index
    CellCount count
    TreeNode* parent
}
```

The algorithm for segmenting the map is shown in Algorithm 1. It draws on the design of Privtrie [11], which was designed for calculating frequent discrete attribute, and we modified the algorithm process to make it adaptively calculating discretization level of continuous GPS data.

Algorithm 1：DivideTreeNode(**rt**, ***D***, ε, batch_size)

Algorithm 1 ： DivideTreeNode(rt, D, ε , batch_size)

```
Input: the tree root node rt， user subset D, local differential privacy budget $\varepsilon$, batch_size
Output: map division tree rooted with rt
1:  F=∅
2:  CS= set of four sub-areas of rt
3:  count=0
4:  while D!=∅:
5:      choose batch_size users from D, represented as G
6:      delete G elements from D
7:      $F = F \cup G$
8:      for every user u in G do
9:          count+=IsInCell(r.Cell,u.Location, $\varepsilon$ )
10:             if evaluate(count, F.size) > threshold then
11:             for every cell cnode in CS do
12:                 root.Children.append(DivideTreeNode( cnode,D, $\varepsilon$))
13:             break
14:     return root
```

Lines 1–3 are the initialization of parameters. Lines 5–7 indicate that batch_size users are randomly sampled from the set of users assigned to the current node. In the 9–10 line, IsInCell is used to simulate the process of making a query request to the sampled user, and the implementation of the IsInCell function is given in Algorithm 2. Line 10 simulates the process that the data collector uses the evaluate function to remove noise and determine whether the frequency of user access to the node is a certain threshold. We choose max(, 0.001|D|) as threshold, among which, means the variance of evaluate result. Since the evaluate result follows normal distribution, its variance could be calculated easily. If evaluate result is greater than the threshold, then in line 12, corresponding areas to the child nodes are further recursively divided; if it is less, return to line 5, adds more users, and repeat the process of lines 7–13 until D is the empty set.

```
Algorithm 2 : IsInCell(Cell c, Location $l$, double $\varepsilon$)
Input: Location Area $c$, user location $l$, local differential privacy budget $\varepsilon$
Output: 0 or 1
sample from the distribution, and get the result b
    $\Pr[output = 1] = \begin{cases} \mathrm{p} = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1}, \text{ if } l \in \mathrm{c} \\ \mathrm{q} = \frac{1}{e^{\frac{\varepsilon}{2}}+1}, \text{if } l \notin \mathrm{c} \end{cases}$
return b
```

The information collection process given in Algorithm 2 exploits the randomized response mechanism, which has been proved to satisfy local differential privacy [7]. We simply show the proof of local differential privacy here.

There are four situations here, which are:

$$\frac{\Pr[output(l) = 1]}{\Pr[output(l') = 1]} \begin{cases} 1, & if\ l \in c\ \ and\ \ l' \in c' \\ e^{\frac{\varepsilon}{2}}, & if\ l \in c\ \ and\ \ l' \notin c' \\ e^{-\frac{\varepsilon}{2}}, & if\ l \notin c\ \ and\ \ l' \in c' \\ 1, & f\ l \notin c\ \ and\ \ l' \notin c' \end{cases}$$

Thus we can easily see that each IsInCell algorithm satisfies 0.5ε-local differential privacy. Furthermore, in algorithm 1, every user sent bit vector contains at most one 1-bit, and all others 0-bit, so algorithm 1 satisfies ε-local differential privacy. On the server side, The implementation of evaluate function is

$$\text{evaluate}(count, n) = \frac{count - n \cdot q}{p - q}$$

Finally, the algorithm given in Algorithm 1 can get the quadtree corresponding to the map area division, and the leaf nodes in the tree have a one-to-one correspondence with each quantized area.

4.2 Personalized Location Privacy

Since the map has been recursively divided into multiple areas, and the areas are in a tree-like, hierarchical relationship, our method allows users to specify their privacy protection areas. Note that user-specified privacy protection areas are considered not to be private data and it could be obtained directly by the server. Assume that the geographical division module divides the map as shown in Fig. 3.

In order to reduce the error caused by quantization, the user's location data will only be quantized to any element in the set of leaf nodes, in our example, {2, 3, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17} numbered nodes corresponding areas. Assume that a user is quantified to area 11, he is allowed to choose his privacy protection level in 4 levels of differential privacy protection. The numbers of the privacy protection areas

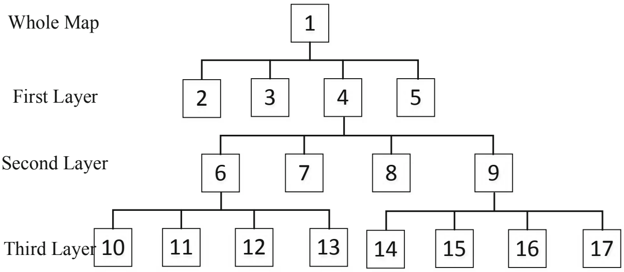


Fig. 3. Example of map division result

corresponding to the four levels are 11, 6, 4, 1, respectively, that is, a user can choose any ancestor node of his location node as his privacy protection area.

For example, when the user selects area 4 as its privacy protection area, according to the definition of personalized local differential privacy, we needs to ensure that on all leaf nodes under area 4, including {7, 8, 10, 11, 12, 13, 14, 15, 16, 17}, local differential privacy needs to be satisfied. The advantage of personalized differential privacy is that the user's data collection process only needs to ensure the differential property in the privacy protection area specified by the user, which doesn't need to consider the probability distribution on the locations outside the privacy protection area, in this example, {2, 3, 5} area.

4.3 Location Density Collection Module

Since the privacy protection areas designated by users are different, firstly, users are divided according to their designated differential privacy protection areas, and a data collection method is called individually for each user subset. This section introduces the design of location collection module.

This module uses the improved method of the binary local hash method proposed by researchers [9, 10, 12] and in order to improve the utility of collection results, this module exploit personalized differential privacy model. Assuming that each user to be collected has designated his privacy protection area, suppose the geographic location of a user u is u.l and privacy protection area is u.L. The collection process is shown in Algorithm 3.

```
Algorithm 3: Location Collection Process
Input: Collector designated parameter m and g, quantization location set D, user set U
Output: All quantization locations' estimated frequency
1: $d$=|D|
2: collector generate a $m \times d$ sized matrix $M$, each item in matrix is randomly chosen from {1,2,3,…,$g$}, and each column corresponds to a location
3: collector initializes a zero matrix $z$, sized $m \times g$
4: collector initializes a $d$ sized zero vector $f$, to save all locations' estimated frequency
5: for every user $u$ in U do
6:     collector randomly generates a number $j$ from {1,2,3,…,m}
7:     collector sends $j$-th row of $M$ to user $u$
8:     user $u$ computes $r$=LocalRandomize(u.l, u.L, $M_{j,.}$), and sends $r$ to collector
9:     collector computes z[j][r]= z[j][r]+1
10:  for every location $l$ in $D$ do
11:      $f_l$ = EstimzteFrequency($M_{.,l}, z$)
12:  return $f$
```

In the first step, the collector generates a random matrix. It should be noted that this matrix does not need to be kept secret. It can be obtained by sharing the key between the data collector and the user and generated from a random stream, which reduces communication overhead of sending the j-th row of matrix M in the row 7. The matrix z in the second step is used to save the user's aggregate statistical results. Steps 6 to 9 are basically the same as the binary local hash mechanism [9, 12]. The difference is that the return value r of LocalRandomize in our method is no longer, but a value in {1, 2, 3,…, g}. Corresponding to that, in step 7, our method takes r as an index, add 1 to the r-th column of the j-th row of the aggregate result z.

The implementation of LocalRandomize and EstimateFrequency are shown in Algorithm 4 and Algorithm 5 respectively.

```
Algorithm 4: LocalRandomize
Input: user location $l$, user designated privacy protection area $L$, $j$-th row $R$ of matrix $M$, location quantization set D
Output: disturbed user location index from {1,2,3,…,g}
1: $e$=R[$l$]
2: user randomizes $z$ following the distribution,and get the result $v$
   $\Pr[v=z]=\begin{cases}\dfrac{e^{\varepsilon}}{e^{\varepsilon}+g-1}, & z=e\\ \dfrac{1}{e^{\varepsilon}+g-1}, & z\neq e\end{cases}$
3: return $v$
```

Since for every user, randomized response mechanism is invoked, and the proof is the same as Algorithm 2.

```
Algorithm 5 ： EstimateFrequency
Input: the location encoding $c$, aggregate matrix $z$, user number $N$ that designate the location as their privacy protection area
Output: the location's estimated visit frequency
1: $p = \frac{e^{\varepsilon}}{e^{\varepsilon} + g - 1}$, $q = \frac{1}{g}$
2: count=0
3: for i=0;i<c.size;i++:
4:     count+=z[i][c[i]]
5: return $\frac{\text{count} - N \cdot q}{p - q}$
```

The basic idea of the frequency estimation process in Algorithm 5 is the same as the randomize response mechanism. The difference is that the user aggregation result here is a matrix instead of a vector. Since each column of the random matrix generated by the collector can be regarded as a encoding of a location area, each element is randomly chosen from {1, 2,…, g}. So when estimating the frequency, only the same indexed aggregation value as the target encoding needs to be count. So in line 4, we first take the value of the column c[i], and use c[i] as index to take the corresponding aggregation frequency value in z. After eliminating the bias in line 5, we can get the estimated frequency of the target attribute.

It should be noted that in our method, Location Collection Process needs to be invoked for every set of users that designate the same privacy protection area. But this wouldn't be a efficiency bottleneck, because every user still only needs to participates in one collection. After all users location data has been collected, add the estimated results in each collection and then the total corresponds to the location's real visit frequency.

5 Experimental Validation

5.1 Experiment Setup

In our experiment, we use Brinkhoff [16] and the Portugal taxi trajectory dataset as the users' location data set.

Brinkhoff is trajectory generator that has been widely adopted as benchmark [17, 18]. It takes the map in the real world as the input, and establishes a trajectory generator, which can generate trajectory data sets of any size according to the characteristics specified by the user. In the experiment, the German Oldenberg is used as the map, and a total of 1,000,000 trajectory data are generated as the trajectory data set of the experiment.

Protugal taxi trajectory dataset was drawn from the ECML/PKDD 2015, and we randomly chose 1,000,000 trajectory data from original 1,673,686 trajectories.

Since the goal of our method is to collect the users' geographic location data as accurately as possible, we compare the collected user location distribution with real user data distribution to evaluate the geographic location collection method proposed in this paper. The evaluation indicators adopted in this article are the same as PSDA work and are as follows:

(1) KL distance. We calculate the distribution of the original data set on the geographical location division results, and then calculate the distribution of the collected location access probability distribution. In order to measure the distance between the two distributions, KL divergence is used as the evaluation metric.
(2) The accuracy of top-K areas with the highest density. We calculate the K locations with the highest frequency of density in the original data set, then calculate the K locations with the highest frequency of access in the estimation result, and calculate the accuracy of the estimation result.

5.2 Experiment Results

The performance of this scheme and PSDA scheme on the KL distance evaluation index on different data sets is shown in Fig. 4.

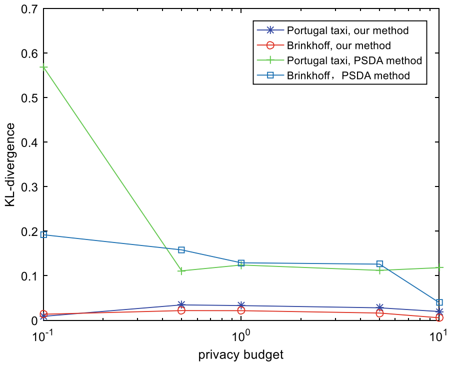


Fig. 4. KL divergence between original data set and collected results.

It can also be verified that under the same local differential privacy budget, out method could achieve lower KL divergence and higher top-K accuracy than PSDA method. In addition, it should be noted that in Fig. 5, when differential privacy budget, the geographical location is divided and the size of the division location set is less than K = 100, so the accuracy rate of the K regions with the highest access density is 100%. It can be seen that the accuracy of the experimental results in Fig. 6 does not increase with the increase in differential privacy budget. According to the analysis, there are two reasons for this phenomenon:

(1) The top-K indicator only cares about the frequency of the area with a larger frequency, and the collection result of the area with a higher frequency itself has higher signal-to-noise and is less affected by noise.

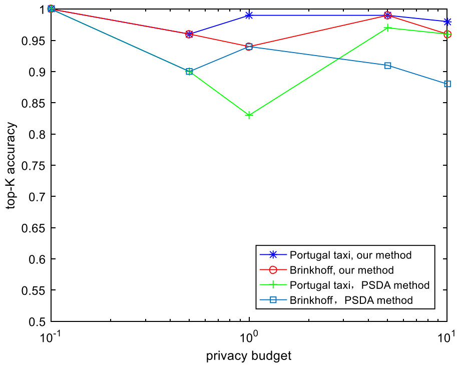


Fig. 5. Top-K accuracy of collected location results. K = 100.

(2) The location collection module also uses the result of the geographic location division module. As the differential privacy overhead increases, the variance of the noise also decreases, so the threshold of the leaf nodes in the division process also decreases. As a result, the leaf nodes are further divided, making the location set larger. In the experiments of the Portuguese taxi data set, the change of the size of the divided location set with the differential privacy budget is shown in Fig. 6.

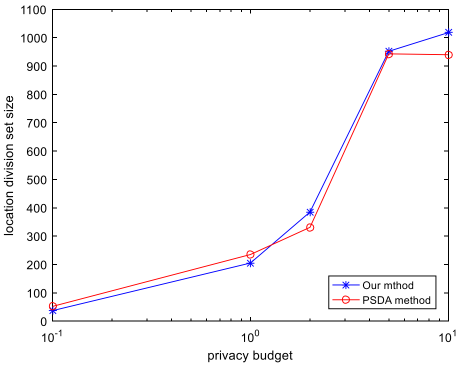


Fig. 6. Change of location division result size with differential privacy budget.

It can be seen from Fig. 6 that the size change of the location set obtained by this scheme and PSDA scheme is basically the same. When the differential privacy budget is low, the number of geographically divided areas is also low, which can compensate for the increase in noise, even if signal-to-noise ratio of each collected location density reduces. It should be noted that in the experiments corresponding to Fig. 4 and Fig. 5, PSDA scheme also has this effect, but because the noise amplitude of their method grows too fast, the change in the size of the location set is not fast enough to compensate for the increase of noise. Therefore, its accuracy shows a significant downward trend, which also proves that the method proposed in our paper could achieve better collected results utility.

In order to further illustrate the influence of the original location set size and location division results size on the accuracy of the final collection results, experiments are carried out on different sizes of original datasets. The experimental results are shown in Table 1. Note that original data size's unit is million.

Table 1. Change of evaluation with dataset size (batch_size = 1000)

Original dataset size/million	0.2	0.4	0.6	0.8	1
KL divergence	0.0208	0.0639	0.0868	0.136	0.222
Top-K	0.93	0.96	0.97	0.98	0.97
Location division set size	133	538	1546	2653	5239

As can be seen from the results in Table 1, as the scale of the data set increases, the number of regions obtained by dividing the map by the location division module has increased significantly, and the relative proportion of the growth rate is far faster than the growth rate of the scale of the data set, resulting in that the signal-to-noise ratio averaged in each area is reduced. With the increase in the size of the data set, the KL divergence indicator showed a significant increase, but the top-k accuracy rate remained almost unchanged. The reason for this result is that the KL divergence represents the accuracy of the collection results of all regions, and the top-K accuracy represents the accuracy of the collection results of high-frequency sub-regions, so the latter itself is less affected by noise. In summary, it can be concluded that if the goal of collecting data only considers high-frequency attributes, the system can achieve high-precision collection results without special settings; if the data to be collected needs to consider the frequency of all attributes, we need to adjust the size of batch_size according to the size of the user set to be collected, so that the number of regions divided by the geographic location division module increases in proportion to the size of the data set, so as to ensure the relative stability of the signal-to-noise ratio.

6 Conclusion

In this paper, we explain the necessity of privately collecting user locations from the perspective of users and service providers, and then divides the private collection method into a location division module and a location density collection module, and explains functions and principles of the two modules. Finally, the utility and accuracy of the method are tested using the Brinkhoff trajectory generator and the Portugal taxi trajectory data set. The results shows that out method could achieve better utility than the best method known so far.

References

1. Fawaz, K., Feng, H., Shin, K.G.: Anatomization and protection of mobile apps' location privacy threats. In: 24th USENIX Security Symposium, pp. 753–768. USENIX (2015)
2. Chen, R., Fung, B., Desai, B.C.: Differentially private trajectory data publication. arXiv preprint arXiv:1112.2020 (2011)
3. Chen, R., Acs, G., Castelluccia, C.: Differentially private sequential data publication via variable-length n-grams. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 638–649. ACM (2012)
4. Zhang, J., Xiao, X., Xie, X.: PrivTree: a differentially private algorithm for hierarchical decompositions. In: Proceedings of the 2016 International Conference on Management of Data, pp. 155–170. ACM (2016)
5. He, X., Cormode, G., Machanavajjhala, A., Procopiuc, C.M., Srivastava, D.: DPT: differentially private trajectory synthesis using hierarchical reference systems. In: Proceedings of the VLDB Endowment, pp. 1154–1165. Springer (2015)
6. Gursoy, M.E., Liu, L., Truex, S., Yu, L., Wei, W.: Utility-aware synthesis of differentially private and attack-resilient location traces. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 196–211. ACM (2018)
7. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1054–1067. ACM (2014)
8. Fanti, G., Pihur, V., Erlingsson, Ú.: Building a rappor with the unknown: privacy-preserving learning of associations and data dictionaries. Proc. Priv. Enhanc. Technol. **2016**(3), 41–61 (2016)
9. Bassily, R., Smith, A.: Local, private, efficient protocols for succinct histograms. In: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, pp. 127–135. ACM (2015)
10. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: 26th USENIX Security Symposium, pp. 729–745. USENIX (2017)
11. Wang, N., et al.: PrivTrie: effective frequent term discovery under local differential privacy. In: 2018 IEEE 34th International Conference on Data Engineering, pp. 821–832. IEEE (2018)
12. Chen, R., Li, H., Qin, A.K., Kasiviswanathan, S.P., Jin, H.: Private spatial data aggregation in the local setting. In: 2016 IEEE 32nd International Conference on Data Engineering, pp. 289–300. IEEE (2016)
13. Warner, S.L.: Randomized response: a survey technique for eliminating evasive answer bias. J. Am. Stat. Assoc. **60**(309), 63–69 (1965)
14. Samet, H.: The quadtree and related hierarchical data structures. ACM Comput. Surv. **16**(2), 187–260 (1984)
15. Ho, S.S., Ruan, S.: Differential privacy for location pattern mining. In: Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS, pp. 17–24. ACM (2011)
16. Brinkhoff, T.: Generating network-based moving objects. In: Proceedings of the 12th International Conference on Scientific and Statistical Database Management, pp. 253–255. IEEE (2000)
17. Agarwal, P.K., Fox, K., Munagala, K., Nath, A., Pan, J., Taylor, E.: Subtrajectory clustering: models and algorithms. In: Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pp. 75–87, May 2018
18. Orakzai, F., Calders, T., Pedersen, T.B.: k/2-hop: fast mining of convoy patterns with effective pruning. Proc. VLDB Endow. **12**(9), 948–960 (2019)

Systems Security

Analysis on the Security of Satellite Internet

Huan Cao[1], Lili Wu[2], Yue Chen[2], Yongtao Su[1], Zhengchao Lei[2](✉), and Chunping Zhao[3]

[1] Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
caohuan@ict.ac.cn

[2] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China
leizhengchao@cert.org.cn

[3] Beijing Sylincom Technology Co., Ltd., Beijing, China

Abstract. Satellite Internet (SI) is a new way to provide internet access all over the world. It will bring great convenience to international communication. Compared with the traditional communication networks, SI has a significant change in network architecture and communication model, which will have an important impact on national information network security. For example, the global interconnected SI consists of a large number of small satellites and each satellite has multi-beams to cover a vast area, which leads to the disorderly flow of information across the border, and greatly increases the difficulty of network protection. Therefore, it is necessary to closely track the development of SI and analyze security problems brought by SI. In this paper, we analyze the security risks of SI from the perspective of national security, network security and equipment security, and thirteen security issues have been summarized to provide reference for the healthy development of SI industry.

Keywords: Satellite internet · Network security

1 Introduction

In recent years, the world's space powers have proposed low-earth-orbit (LEO) satellite constellation plans, which has triggered a boom in satellite internet (SI) development. Concerning the development of SI, the white paper published by China Center for Information Industry Development (CCID) points out that the world is on the eve of the dense launch of man-made satellites [1]. It is estimated that the low Earth orbit (LEO) satellites will deploy a total of about 57000 by 2029. A space resource race of satellite orbits is quietly beginning, countries all over the world have joined in the space race of SI, and the earth surface will be covered by a large number of LEO satellites intensively. Therefore, security problems brought by this will become a new challenge [2–4]. With the construction of SI becoming a national strategy all over the world, the industry has entered a period of rapid market growth [5, 6], and it specifically reflected in the following aspects:

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 193–205, 2020.
https://doi.org/10.1007/978-981-33-4922-3_14

- Fighting for frequency and orbit resources: The competition for frequency and orbit resources among countries has become increasingly white-hot. According to the data submitted to international telecommunications union (ITU), satellite companies in France, United States and United Kingdom have the largest number of resources such as key frequency bands and orbital heights. For example, OneWeb has submitted at least seven materials of network resources to ITU, including THEO, STRIPE, 102, etc., covering 8425 km / 8575 km, 1200 km and other medium and low orbital altitude, as well as Ku / Ka / V and other frequency bands; SpaceX submitted twelve materials to ITU, including usasat-ngso-3a-r / 3b-r / 3C, 3D / 3E / 3F / 3G / 3H / 3I, usasat-ngso-3j / 3K / 3l, covering 345.6–1325 km orbital altitude and Ku / Ka / V frequency bands.
- Large-scale network deployment: SI constellation construction has entered the stage of large-scale network deployment. SpaceX plans to launch 42000 satellites, and 482 broadband Starlink satellites have been launched by June 5, 2020. In addition, OneWeb has launched 74 satellites in the past two years [7].
- International operation: The service providers of SI have been striving for landing rights in countries around the world. For example, OneWeb initially obtained market access authorization in about 19 countries in 2019.

SI can be mainly used for emergency rescue, rural and remote area coverage, maritime market (including cruise ships, merchant ships, fishing boats, yachts, etc.), aviation market, military and government applications [8]. Compared with the terrestrial mobile communication system (TMCS), the SI will face the following new security challenges:

- Due to the limited computing and storage capacity, the satellites in SI constellation don't support high-complexity encryption protocols and algorithms, resulting in the weak protection of traffic data.
- The topological structure of the LEO satellite networks are constantly changing, the openness of the satellite's orbit makes it very difficult to be supervised.
- Communication satellite is a highly integrated product, its components are supplied by many manufacturers. There may be security holes and design defects in all aspects of integration. Especially, the technology of on-orbit satellite reprogramming is not mature, which makes it very difficult to make up for the security holes of on-orbit satellites.
- Satellite communication has the characteristics of wide coverage [9], which can broadcast data to a large number of user terminals in a large range. When the SI network is attacked, the impact is greater than that of the TMCS, so it is easier to become the target of hackers.

In summary, the security problems faced by the SI are more severe than those of the TMCS. If the SI is attacked, it will have a wider range of influence and cause greater damage. Therefore, it is necessary to carry out the research on the security problems faced by the SI.

2 Related Work

The research on the security problems of SI is still in its infancy. 3GPP puts forward the network architecture of non-terrestrial networks (NTN) [10], but there is no systematic analysis on the security problems of NTN. Sat5G analyzes the security threats of the integration of satellite network and 5G networks, mainly including the following three main aspects [11]:

1. Security threats of satellite connections as transport network for backhaul
   One of the main security threats perceived by the terrestrial network is the tampering or eavesdropping of the data transmitted (the control plane signaling or the user plane data) over the backhaul connection. In addition, another threat perceived by terrestrial networks in case of sharing of the satellite network is the tampering and eavesdropping of traffic via the shared network.
2. Security threats of satellite connections as transport network among 5G core networks
   In this case, the two terrestrial networks usually are not in the same trust domain, and the intermediate satellite network is not considered to be part of the trust domain of either of the two terrestrial networks. At the same time, it is very common for satellite networks to be shared among multiple terrestrial networks. The security threats perceived by the terrestrial network are tampering, eavesdropping and unauthorized traffic redirection (i.e. traffic 'hijacking') [12, 13].
3. Security threats to content delivery via satellite

Security threats related to content delivery networks (CDN) are DDOS at-tacks; Content leakages, such as unauthorized access to content, which is aggravated by local caching and the use of MEC servers; Deep linking, in this case, all media slices can be accessed by accessing a manifest file due to use MEPG DASH.

However, Sat5G has made a preliminary analysis of the security issues of SI, but it is not comprehensive enough. This paper summarizes and analyzes the security issues faced by SI in the future from the aspects of national security, network security and equipment security based on the existing research.

3 Analysis of SI Security

3.1 Overview of Security Issues of SI

The system architecture of SI can be divided into user segment, space segment and ground segment. The user segment includes various satellite terminals; the space segment includes satellite constellation [14], which can be divided into constellation with inter satellite link (ISL) and constellation without ISL [15]; the ground segment includes gateway station (GS), operation management and control system (OMCS), measurement and control system (MCS), network management system (NMS), etc. According to the characteristics of SI, the possible security problems in SI are summarized in Table 1.

Table 1. Security issues of SI.

Classification	ID	Security problem	Description
National security	(1)	National and military security threats	• Illegal organizations can steal strategic information of target countries by deploying earth observation payload on LEO satellites • LEO satellite provides communication platform for future information warfare weapons
	(2)	Frequency and orbit resource preemption	To occupy limited orbit resources by planning LEO satellite constellation
	(3)	Interference in astronomical exploration	The launch of a large number of LEO satellites can cause serious interference to astronomical observation
Network security	(4)	Identity impersonation	• Disguised as a satellite terminal (ST) to access the SI and destroy the network • Disguised as a satellite to trick legitimate STs into accessing a false network to obtain the ST's location or identification information
	(5)	Data eavesdropping	Illegal organizations illegally receive and analyze transmitted traffic data or signaling data through wireless links (feedback link, user link, ISL)
	(6)	Data integrity issues	Modify, insert, replay, delete user or signaling data to destroy data integrity
	(7)	Information interception	Illegal interception of user location or identification information transmitted by ST through wireless links
	(8)	Signal interference	Attackers interfere with satellite wireless links by emitting high-power electromagnetic waves
	(9)	Denial of service	Interfere with satellite or gateway, and interfere with data or signaling physically or by protocol, which makes SI unable to provide normal services for legitimate ST
	(10)	Anonymous attack	Attackers attack the satellite node in space, but the satellite cannot determine the attackers

(*continued*)

Table 1. (*continued*)

Classification	ID	Security problem	Description
	(11)	Malicious occupation of satellite bandwidth resources	Sending illegal signals to the satellite through wireless link, because the satellite will not check the legitimacy of the signals, so the illegal signals will occupy the bandwidth resources of the satellite
Equipment security	(12)	Malicious satellite control	By issuing malicious instructions or injecting viruses to satellite nodes from ground facilities or space to achieve the goal of controlling satellites
	(13)	Malicious consumption of satellite resources	Malicious consumption of satellite propellant resources to achieve the goal of reducing satellite life

The distribution of the above thirteen security issues in the SI is shown in Fig. 1.

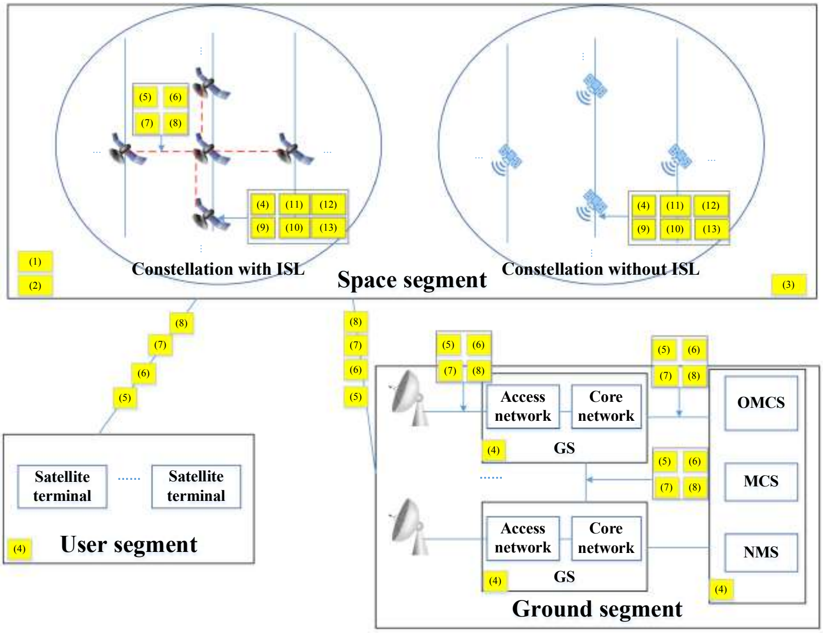


Fig. 1. The distribution of security issues in SI system

3.2 National Security Issues

National and Military Security

The security threats include national strategic information security and military security threats.

National Strategic Information Security

SI involves a large number of satellites, and the orbit altitude is concentrated between 300 km and 2000 km. If the corresponding satellites equipped with high-resolution scanning observation payloads, such a large number of satellites will expose the important military infrastructure of countries all over the world and threaten national security. Recently, earth observation industry (EOI) company is promoting the development of a new very low earth orbit (VLEO) satellite constellation. Its propulsion system and innovative design will enable the satellite to run in a very low orbit. In order to support continuous monitoring service, the initial constellation consists of 30 satellites with an average revisit time of two hours. The company plans to launch its first satellite by the end of 2022. EOI company's mission is to enable defense and intelligence agencies and commercial customers to easily access ultra-high resolution images at affordable prices to support a range of applications such as resource management, environment and disaster assessment, asset monitoring, logistics planning, infrastructure mapping, public security, homeland security, insurance and real estate. For example, Fig. 2 shows the image of naval ship captured by EOI company's VLEO satellite.

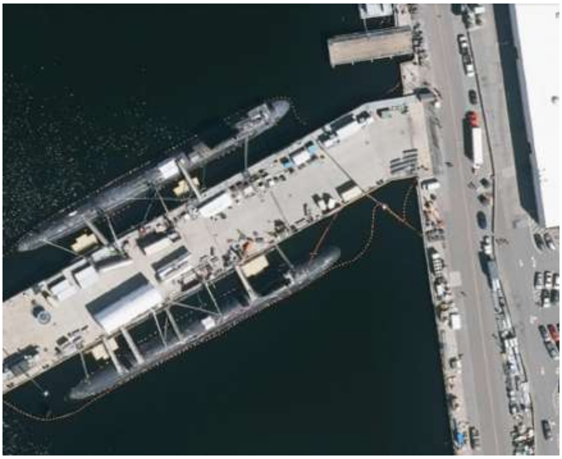


Fig. 2. Image of naval ship captured by EOI's VLEO satellite

Military Security Threat

1. The low cost and high launch success rate (LSR) of SI will pose new challenges to the technical field of anti-satellite weapons.

The existing kinetic energy anti-satellite weapons (KEAW) rely on the momentum of high-speed moving objects to destroy the target, which has a strong lethality for satellites with high cost and low LSR. However, for the large-scale and low-cost SI, traditional KEAW are facing many challenges. Taking Starlink constellation of SpaceX as an example:

a. The traditional KEAW are all disposable. It means that a large number of KEAW need to be manufactured and maintained to deal with the threat of the Starlink constellation of 42000 satellites, and the cost will be astronomical.
b. The traditional KEAW adopts the hard-kill method. The method will generate a large number of space debris, which may hit more satellites, causing uncontrollable and irreversible chain reaction, making the whole earth surrounded by satellite debris.
c. If we give up the hard-kill method and study more advanced weapons such as soft-kill method, it will cost a lot of money to tackle key technical problems, and the development cycle will very long.

2. The cooperative operation of SI and drone swarm will pose great challenges to the national defense system.

With the accelerated evolution of the war form, a large number of intelligent equipment appear in the war. As an important part of intelligent warfare, Unmanned Aerial Vehicle (UAV) cluster warfare poses great challenges to the traditional defense system. However, the UAV cluster warfare relies on the communication link among each UAV to achieve real-time information interaction, and also relies on the control system to achieve collaborative command, so the overall viability and combat ability of the UAV cluster depends on the security and controllability of the communication link and control system. If the communication link or the control system is jammed by enemy, the UAV cluster will likely to be completely annihilated. Starlink constellation can make up for this defect, it can provide large bandwidth, low delay and wide coverage communication services through a large number of LEO satellites without being affected by any terrain and climate. It can help the UAV cluster get rid of the dependence on land-based communication system, and significantly improve the overall combat effectiveness of the cluster through flight control, situation awareness, information sharing, target allocation and intelligent decision-making, which makes it more difficult for the national defense system to deal with the threat of UAV cluster warfare.

Frequency and Orbit Resource Preemption

According to international regulations, all countries have the right to explore outer space peacefully. Radio frequencies and satellite orbits are limited natural resources and must be used equally, reasonably, economically and effectively. Effective interference control mechanisms should be adopted to make full use of frequency and orbit resources. Effective interference control mechanisms should be adopted to make full use of the limited resources. According to the ITU rules [16], orbit resources are mainly allocated in the principle of first come, first served, and the later declarers cannot cause adverse interference to the satellites of the first declarers. The LEO constellation system should not only launch the satellite in accordance with ITU regulations, but also provide relevant services to the public in accordance with the specified time and proportion, so as to legalize the frequency usage. In other words, the development and utilization of LEO constellation can not only occupy the limited space resources of LEO satellite, but also help to seize the priority use right of spectrum, which has an important impact on the channel use range of battlefield communication.

Generally, at the end of a geostationary earth orbit (GEO) satellite's life, it will increase more than 200 km by using its final energy and enter the grave orbit to complete its self-destruction, so as to release the original working orbit. But the LEO satellite communication system is different, it needs many small satellites to maintain a complete network to provide communication services. When some small satellites cannot work normally or reach the end of their life, it is necessary to launch new satellites to supplement the network, so they will always occupy the frequency and orbit resources. However, the near earth orbit resources can only hold about 60000 satellites, the United States, the United Kingdom, France, Canada, Norway, the Netherlands and other countries have taken the lead in the deployment of SI. SpaceX alone plans to launch 42000 LEO satellites, which will further compress the available orbit resources of other countries.

In addition, all countries except the United States have gaps in the supervision technology of SI, and corresponding laws are not perfect. Once the design and construction of SIs are completed, it will lead to difficulties for countries to effectively supervise the communication services provided by SI, leaving huge security loopholes.

Interference in Astronomical Exploration

Due to the huge scale of SI constellations, astronomical observation will become more difficult as small satellites in constellations are launched one after another. Starlink will launch 42000 satellites, with an average of about 400 satellites observed at any time and at any place. Although they are invisible to the naked eye in orbit, they have a great influence on the astronomical research of optical, infrared and radio telescopes, and are easy to leave traces in the astronomical images. The large-scale integrated Sky Survey Telescope (such as China sky eye) will be greatly affected, which will reduce our ability to observe and warn near Earth Asteroids.

In addition, LEO constellations have the most interference for astronomers who detect dark matter and dark energy, because the signals detected by related instruments are very weak. A large number of LEO satellites will interfere with the space observation of various countries to a certain extent when passing over them, and affecting the corresponding research.

3.3 Network Security Issues

Identity Impersonation

Due to the lack of identity authentication mechanism in SI's user link, feedback link and ISL, there are three problems of identity impersonation in the following aspects:

1. If the transmission mechanism adopted by the communication system is public, the attacker can calculate the uplink signal according to the downlink signal of the satellite, and then use the satellite communication equipment to disguise as a legitimate ST to access the network and illegally obtain network services.
2. The attacker disguised himself as a satellite network and induced legal STs to access the satellite network to obtain relevant user identification information and location information.
3. The attacker disguised himself as adjacent satellites in the same orbit or different orbit to induce the target satellite to establish an ISL with it, so as to obtain the relevant data transmitted by the ISL.

Data Eavesdropping and Data Integrity Attack

Due to the openness of wireless communication of user link, ISL and feed link of SI, the data transmitted through satellite network can be easily eavesdropped. In addition, data encryption will increase the cost of satellite terminal equipment and reduce the utilization rate of satellite link resources. Many satellite communication networks do not encrypt the transmitted data, so it is very easy to cause data leakage.

The most possible attack methods are as follows:

1. The attacker uses a kind of satellite data receiving card to steal data, which is similar to the computer network card with low cost.
2. The attacker makes use of retired equipment abandoned by manufacturers to perform network attacks.
3. The attacker can use the VLEO or LEO satellite in the overseas satellite constellation to eavesdrop the service data on the user link and feeder link of the domestic satellite system.
4. If the satellite constellation built in a country has an ISL, which uses microwave communication, the attacker can control the foreign satellite to approach the target satellite as close as possible to implement data eavesdropping.

Data eavesdropping is often combined with data integrity attack. The attacker often implements data eavesdropping, then inserts, modifies, falsifies the stolen data, and finally send it to the data receiver to achieve the purpose of destroying data integrity.

Information Interception

If the orbit of a foreign satellite is lower than that of a domestic satellite (for example, the lowest orbit of SpaceX is about 300-500km), the attacker can use the attack means similar to the terrestrial pseudo base station to carry out network attack. For example, the torpedo attack method in 4G system can be used in SI, the attacker can use legitimate ST to launch multiple paging to the attacked ST, and this will expose the user identification information, which can be intercepted by the LEO satellite and terrestrial equipment owned by the attacker, so as to track the user's location and bring great security threat.

Signal Interference

This kind of attack is the most common but effective, and it is often used in wars. Interference can be divided into blocking interference and noise interference. Strong interference signals will cause the satellite to be unable to receive the signal normally and provide the service for the legitimate ST.

The possible attack methods are as follows:

1. If the orbit of the overseas satellite is lower than that of the domestic satellite, the attacker can deliberately transmit signals on the working frequency band of the feeder link, user link or ISL of the domestic satellite system to cause interference or interruption of the domestic satellite service.
2. Satellite transponders can be divided into on-board processing transponders and bent-pipe transparent transponders. On-board processing transponders can rely on channel coding, advanced modulation technology, spread spectrum technology, coherent intermodulation product cancellation, etc. to resist interference attacks. But the bent-pipe transponder has a simple structure and does not process any communication signals, so it is easy to encounter signal interference attack. Attackers can interfere with satellites by transmitting signals from high-power transmitters.

Denial of Service

The attack mode against terrestrial network is also applicable to satellite network, such as DDoS attack. Attackers make use of software to simulate massive satellite terminals to send legitimate and bogus requests, which leads to the failure of satellites to provide effective services to legitimate STs. This kind of attack is difficult to defend due to the diversity of satellite communication links. Each ST's client has a receiving and transmitting system. If the transceiver fails to get effective processing when it has problems, it will lead to unstable connection and generate a large number of connection requests. In addition, access requests will also increase greatly when satellite links suffer from signal fading caused by severe weather. However, satellites cannot blindly defend these requests, and the system design of satellite system will not defend these requests as well as the network firewall. Because satellites cannot distinguish whether these requests come from legitimate STs or malicious attackers, which leads to denial of service problems.

Anonymous Attack

Space belongs to the global commons and has no national boundaries. Therefore, it is possible for attackers to launch anonymous attacks against the target satellite in space. Moreover, it is difficult for the attacked satellite to determine and trace the attacks due to the long distance and limited information. On the one hand, there are many factors that lead to satellite failure, such as changes in space environment, design defects, device problems and even debris effects. Attacks are not the only reason for the failure of satellites in orbit. On the other hand, it is difficult for the ground station to accurately judge what is happening in space limited by distance, meteorology, technical capability and other conditions. The combined effect of these factors enables the attacker to find reasonable excuses to deny the attack.

Malicious Occupation of Satellite Bandwidth Resources

Satellite is a typically resource limited system, on-board computing resources, wireless resources are very scarce, so they are not suitable for complex communication payloads. Most of the on-orbit satellites adopt the bent-pipe transponder without signal unpacking, so it is not possible to determine whether the received data is from a legitimate user. When the attacker sends his own illegal signal, the satellite will still forward the signal to the GS. At this time, if the attacker builds a receiving system to demodulate, decode the data and extract useful data, the purpose of privately communicating with the aid of the satellite is achieved, and a complete method of stealing satellite resources is formed. Moreover, attackers will use their own encryption algorithm to effectively encrypt communication data.

3.4 Equipment Security Issues

Malicious Satellite Control

Due to the lack of network security standards for commercial satellites, coupled with the complex supply chain of satellites, satellite manufacturing uses ready-made technologies to maintain low cost. The wide availability of these components means that hackers can analyze their security vulnerabilities. In addition, many components use open source technology, and hackers may insert backdoors and other vulnerabilities in satellite software, making satellites vulnerable to security risks that are maliciously controlled by attackers.

The means to control the satellite maliciously are as follows:

1. The attacker can capture the target satellite in space and drag the captured satellite out of the working orbit, causing the whole satellite constellation unable to provide continuous services. Moreover, the attacker can inject virus into the captured target satellite after it has been dragged off the working orbit, and then it will be pushed back to the working orbit, causing the virus to spread throughout the whole SI. The technology of capturing on-orbit satellites is already available and has been used in the service of extending the life of on-orbit satellites in orbit. Once this technology is used by hackers, the target satellites can be captured arbitrarily.
2. Satellites are usually controlled by the GSs. These stations are vulnerable to the same network attacks as computers. Although the satellite control attack is not as simple as stealing other people's email, but it can be realized. If there are security loopholes that can be exploited by hackers in the GS, the hackers may invade these stations, and then they can send malicious instructions to control the satellite, or they can use special tools to trick the satellite, and finally achieve the purpose of attacking the SI. For example, the attacker can carry out further attacks after controlling the target

satellite: the attacker can use the broadcast channel of the target satellite to send a large amount of garbage data or spread viruses to the whole SI; shutting down the target satellite to make it unable to provide normal services; if the hackers control the target satellite and it has a propeller device, they can change the orbit of the satellite and hit it on the ground, other satellites or even the international space station.

Malicious Consumption of Satellite Resources

Attackers can also directly affect the life of satellites by consuming propellants, depleting the write life of charged erasable programmable read-only memory (EEPROM) and other attacks.

4 Conclusion

The rapid development of SI has brought some security risks. On the one hand, we should actively develop SI industry, giving full play to the unique advantages of SI, which is not affected by geographical obstacles and disasters; on the other hand, in view of the different levels of security threats faced by SI, it is necessary to carry out forward-looking research on the satellite network security, so as to fill in the regulatory gaps.

Acknowledgement. This research was supported by GF science and technology foundation enhancement program, National computer network and information security management center (XDA-Y07–02).

References

1. CCID. research on the development of China's satellite Internet industry. In: CCID. https://zhuanlan.zhihu.com/p/144513640 (2020)
2. De Azúa, J.A.R., Calveras, A., Camps, A.: Internet of Satellites. IoSat), Analysis of Network Models and Routing Protocol Requirements. (2018)
3. Su Y., Liu Y., Zhou Y., Yuan J., Cao H., Shi J.: Broadband LEO Satellite Communications: Architectures and Key Technologies (2019).
4. Cao H., Su Y., Zhou Y., Hu J.: QoS Guaranteed Load Balancing in Broadband Multi-Beam Satellite Networks (2019).
5. Anpilogov V.R., Gritsenko A.A., Chekushkin Y.N., Zimin I.V.: A Conflict in the Radio Frequency Spectrum of LEO-HTS and HEO-HTS Systems (2018).
6. Tong, X., et al.: Normalized Projection Models for Geostationary Remote Sensing Satellite: A Comprehensive Comparative Analysis (January 2019). IEEE Trans. Geosci. Remote Sens. **57**(57), 9643 (2019)
7. Foust J.: SpaceX's space-Internet woes: Despite technical glitches, the company plans to launch the first of nearly 12,000 satellites in 2019 (2019).
8. 3GPP.: Study on using Satellite Access in 5G. TR 22.822 (2018). https://3gpp.org/DynaReport/38-series.htm.

9. Sacchi, C., Rossi, T., Murroni, M., Ruggieri, M.: Extremely High Frequency (EHF) Bands for Future Broadcast Satellite Services: Opportunities and Challenges. IEEE Trans. Broadcast. **65**(65), 609 (2019)
10. 3GPP.: Solutions for NR to support non-terrestrial networks (NTN). TR 38.821 R16 (2018). https://3gpp.org/DynaReport/38-series.ht.
11. Sat5G. Extending 5G Security to Satellites. D4.5 (2019). https//www.sat5g-project.eu/public-deliverables.
12. 3GPP.: Security architecture and procedures for 5G system. TS 33.501 (2019). https://3gpp.org/DynaReport/33-series.htm.
13. 3GPP.: System Architecture for the 5G System (5GS); Stage 2. TS 23.501 (2019). https://3gpp.org/DynaReport/23-series.htm.
14. Jiang J., Yan S., Peng M.: Regional LEO Satellite Constellation Design Based on User Requirements (2018).
15. Xia S., Jiang Q., Zou C., Li G.: Beam Coverage Comparison of LEO Satellite Systems Based on User Diversification (2019).
16. ITU.: Radio Regulations 2016 Edition (2016).

A Survey on Cyberspace Search Engines

Ruiguang Li[1,2](✉), Meng Shen[1], Hao Yu[1], Chao Li[2], Pengyu Duan[1],
and Lihuang Zhu[1]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
lrg@cert.org.cn

[2] National Computer Network Emergency Response Technical Team/Coordination,
Center of China, Beijing, China

Abstract. This paper introduces the concept of cyberspace search engine, and makes a deep survey on 5 well-known search engines, say Shodan, Censys, BinaryEdge, ZoomEye and Fofa, by querying official websites, analyzing APIs, and making academic research. We discuss the following items in details: Supporting internet protocols, Total amounts of detected devices, Device information, Scanning frequency, System architecture, The third party databases, Probes distribution, etc. We give a comprehensive comparison of the detecting abilities and working principles of the cyberspace search engines.

Keyword: Cyberspace search engines

Cyberspace search engines, such as Shodan, Censys, BinaryEdge, ZoomEye and Fofa, are new Internet applications in recent years. They search various types of online devices in cyberspace, such as webcams, routers, intelligent refrigerators, industrial control devices, etc. They are becoming powerful tools to detect network resources. At present, mastering the network resources is valuable for cyberspace governance and network security protection. Therefore, global security companies and scientific research institutions pay great attention on the development and utilization of cyberspace search engines. This paper will carry out a comprehensive investigation and analysis on the detection capabilities and working principles of 5 well-known search engines.

1 Introduction

Network resources exploration is to send probe packets to the remote network devices, and to receive and analyze the response data, so as to get the information of remote devices, such as opening ports and services, operating systems, vulnerability distribution, device types, organizations, the geographical position, and so on. The detecting protocols are mainly on the transport layer and the application layer in the TCP/IP stacks. The detection methods of transport layer include SYN scan, TCP connection scan, UDP scan, FIN scan, ICMP scan, etc. Application layer detection mainly uses the special fields of internet protocols, special files, hash values, certificates, and so on.

The working principles of cyberspace search engines are very different from the Web search engines such as Google, Baidu. Web search engines collect, store and analyze

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 206–214, 2020.
https://doi.org/10.1007/978-981-33-4922-3_15

Web page for information querying, while the cyberspace search engines adopt the network resource detecting technology. By sending the detection packet to the remote devices, it can obtain the important information of the target, and conduct comprehensive analysis and display. Global security companies and research institutions have developed a number of search engines, in which the following are most well-known: Shodan (www.shodan.io) Censys (Censys.io) from the US, BinaryEdge (www.binaryedge.io) from Europe, and ZoomEye (www.zoomeye.org) Fofa (www.fofa.so) from China. Some of these engines are commercially available, while others offer none-profit services.

We are very interested in the detection abilities and the working principles of these search engines, so we made a comprehensive investigation on Shodan, Censys, BinaryEdge, ZoomEye, Fofa, by querying official websites, analyzing APIs, and making academic research. The main contents include: Supporting internet protocols, Total amounts of detected devices, Device information, Scanning frequency, System architecture, The third party databases, Probes distribution, etc.

2 Supporting Internet Protocols

Mastering various types of Internet protocol formats is the basis for the exploration of cyberspace search engines. Different devices in the internet have different protocols. In order to facilitate the comparative study, we first carry out a classification of various network devices.

We got all types of devices from the search engine's official websites, and classify all devices into 11 categories: Network Equipments, Terminal, Server, Office Equipment, Industrial Control Equipment, Smart Home, Power Supply Equipment, Web Camera, Remote Management Equipment, Blockchain, Database, shown as Fig. 1.

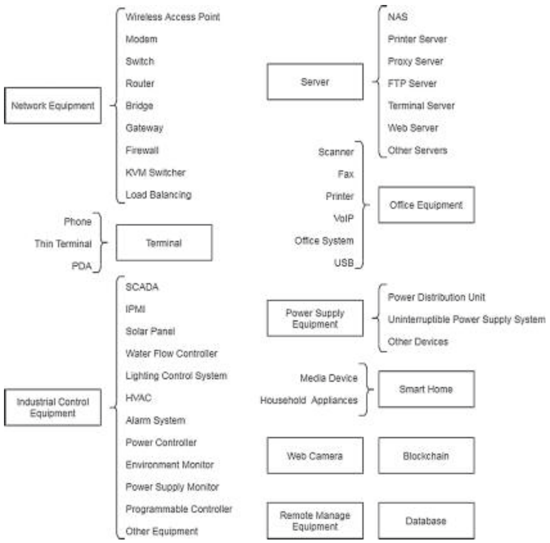


Fig. 1. Device categories

On this basis, we obtained the lists of all engines' supporting protocols from the official websites, user manuals, the APIs, and some technical forums. We classify them into 11 categories according to Fig. 1, shown as Table 1, where "-" means there is no such agreement.

Table 1. Supporting internet protocols

	Shodan	Censys	ZoomEye	Fofa	BinaryEdge
Network equipment	10	1	54	7	8
Terminal	19	1	227	6	13
Server	67	10	154	20	63
Office Equipment	12	5	31	6	11
Industrial Control Equipment	26	5	16	23	17
Smart Home	9	–	3	7	9
Power Supply Equipment	4	1	3	2	4
Web Camera	3	–	8	–	3
Remote Management Equipment	13	5	31	8	11
Blockchain	5	–	4	21	4
Database	17	6	19	16	15
Total	185	34	550	116	158

Shodan's API interface contains supporting protocols that can be directly queried [1]. Censys's protocols information comes from the official forum [2]. ZoomEye's protocols information comes from the NMAP-Services file in the user's manual [3]. Fofa's protocols information comes from the technical forum [4]. BinaryEdge's protocols information comes from the API documentation [5]. As you can see in the table, Shodan and ZoomEye have mastered more types of network protocols, covered all protocol categories, and presumably have better device detecting capabilities. Due to the different statistical caliber of network protocols, there may be some deviation in the comparison results.

3 Total Amounts of Detected Devices

Based on the analysis in Sect. 2, we investigate the total numbers of detected devices of different search engines. Typically, the official websites will claim the total numbers of detected devices, but sometimes we need to do more auxiliary analyzing.

The total amount of Shodan comes from the official website query tool CLi.shodan.io [6]. All the data records after January 1, 2009 can be inquired by the command line tool, so we can calculate the total number of detected devices.

The official website of Censys provides data statistics function [7]. We divide the IPv4 address space into 256 parts, and retrieve each address block with Censys, and calculate the manufacturer's brands of specific types in the returned results, and then obtain the total number as a summary. The total amount of ZoomEye, Fofa and BinaryEdge are from the official website [5, 8, 9].

Table 2. Comparison of the total amount of detectable devices

	Shodan	Censys	ZoomEye	Fofa	BinaryEdge
Total amounts	436489751	111368143	1190860679	270363	89871839

The total numbers of detected devices for each engine are shown in Table 2. As you can see from the table, ZoomEye (nearly 1.2 billion) and Shodan (over 0.4 billion) have the strongest detecting capabilities.

It should be noted that, because of the lack of industry standards in the field of network devices classification, there are statistical caliber of the comparison results.

4 Device Information

Cyberspace search engines need to present the detected device information in a comprehensive way for users to use. One device stands for a file or a record. By analyzing the files or the records, we can get the device information architecture. Typically, the device information architecture includes such important information as domain names, opening ports, services, geographic locations, countries, device types, affiliation, and so on.

We collect, analyze and draw the device information architecture of the above search engines, and make a comparison. We can classify all the device information into: Equipment information, location information, port information, loopholes, probe point information, tag information, network equipment information, WEB information, file transfer, email protocol information, remote access to information, database information, industrial control protocol information, message queues, clustering information. This will be of great value to developers and users of the cyberspace search engines.

Taking Censys as an example, by analyzing the official documents of Censys [10], we get the tree diagram of Censys' device information architecture, as shown in Fig. 2. All these information will be reflected on Censys' web pages. In the below figure, the vulnerability information and probe point information are represented as dotted lines because Censys does not provide such information.

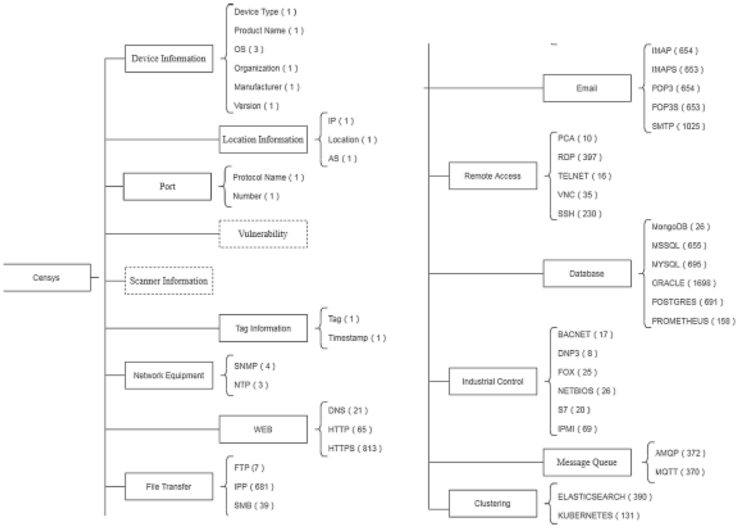


Fig. 2. Device information architecture for censys

5 Scanning Frequency

The cyberspace search engines constantly scan and probe the whole network, discover the new connected devices, and periodically update the detected devices. As a complete scan of the whole network consumes lots of computing and storage resources, so search engines usually set a scanning frequency. Scanning frequency is an important index for the detecting ability. The higher the frequency, the stronger the search engines' performance.

We measured the scanning frequencies of Shodan, Censys, ZoomEye and Fofa. More than 130 IP addresses (opening HTTP, HTTPS, TELNET, FTP and SSH services) were randomly selected. By checking the update status of these IP addresses every day, we can get the scanning intervals of each engines, as shown in Table 3 below.

Table 3. Comparison of scanning frequencies

Protocol (port)	Shodan	Censys	ZoomEye	Fofa
HTTP (80/TCP)	10 days	2 days	389 days	39 days
TELNET (23/TCP)	24 days	2 days	-	-
HTTPS (443/TCP)	9 days	1 day	26 days	102 days
FTP (21/TCP)	13 days	2 days	173 days	74 days
SSH (22/TCP)	10 days	3 days	24 days	60 days

In the above table, "-" means it hasn't been scanned for a long time. As can be seen from the table, that the scanning frequencies of Shodan and Censys are significantly higher than that of ZoomEye and Fofa. We can include that Shodan and Censys have more powerful performance.

6 System Architecture

We are very interested in the system architectures of the cyberspace search engines, so we conducted an extensive academic research. Typically, the architecture of search engine can be divided into three modules: information acquisition module, data storage module and information retrieval module. The information acquisition module is responsible for collecting the information of various devices in the cyberspace. The data storage module is responsible for storing the massive device information collected, and the information retrieval module is responsible for providing statistical and querying services.

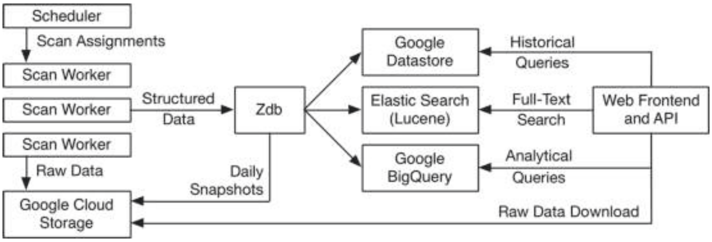


Fig. 3. Censys system architecture1

Figure 3 shows the system architecture of Censys [11]. In the above figure, the Scan Worker is responsible for information acquisition. The Scheduler allocates scanning tasks to multiple scanning modules. The scanning module will save the detection results to Zdb database, and all the information will be stored in Google Cloud. In the information retrieval module, Censys provides elastic Search for full-text retrieval. Google Datastore offers history retrieval and Google BigQuery offers statistics retrieval.

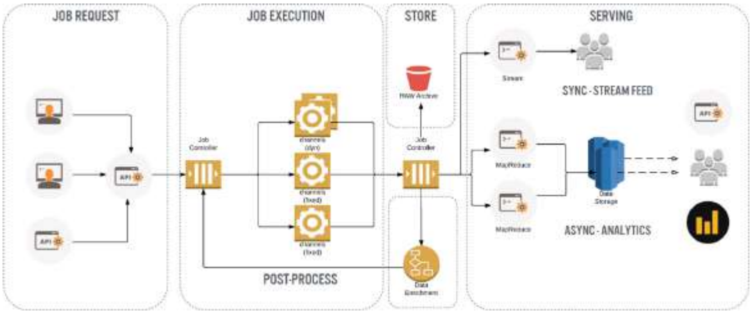


Fig. 4. BinaryEdge system architecture2

BinaryEdge system architecture is shown in Fig. 4 [12], which is divided into four parts: task submission, task execution, storage and service. Task submission uses HTTP, command line, third party and other forms of API for data acquisition. In the task execution stage, the task is sent to multiple channels, including port scanning, Screen shotter, OCR and other technologies. In the storage stage, the collected information will be divided into original data and processed data, and stored in the database. During the service stage, the processed data will be sent to users through a real-time information flow, or deeply analyzed by MapReduce, Kibana, or InfluxDB.

7 Third Party Databases

Many cyberspace search engines work with third-party databases, such as IP databases, domain name databases, and geographic location databases. We investigated the third-party databases associated with commercial search engines, as shown in Table 4 below:

Table 4. Search engines associate third-party databases

	Shodan	Censys	ZoomEye	Fofa	BinaryEdge
IP database	Randomly generated	Randomly generated	-	-	-
Domain database	-	Alexa	-	-	Passive DNS
Address database	-	GeoIP	IPIP	GeoIP	GeoIP

In the table, the IP addresses of Shodan and Censys are randomly generated and do not rely on the third-party IP database. We haven't found the information of ZoomEye, Fofa and BinaryEdge. As for the domain name database, Censys used the domain datas provided by Alexa Top 1 Million Websites, while BinaryEdge used the passive DNS resolution service. We haven't found the information of Shodan, ZoomEye and Fofa. As for geographic location databases, Censys, Fofa and BinaryEdge all use the database of GeoIP, while ZoomEye uses the database of IPIP.net.

8 Probes Distribution

Cyberspace search engines often need to deploy many probes because there are many security devices (such as firewalls) in cyberspace, making it difficult to detect the network edges. Only by deploying widely distributed probes, can we minimize the impact of security devices and find more edge nodes as possible.

We conducted an extensive research, focusing on the open-source tools and third-party organizations. GreyNoise and BinaryEdge have done well.

GreyNoise is a tool for collecting and analyzing scanning traffics [13]. It found the probes of 96 search engines, including Shodan, Censys, BinaryEdge and ZoomEye, as shown in Table 5 below.

Table 5. Probes distribution marked by GreyNoise

	Shodan	Censys	BinaryEdge	ZoomEye
United States	31	398	368	-
Canada	-	-	37	-
Britain	1	-	236	-
Netherlands	10	-	86	-
Iceland	2	-	-	-
Romania	1	-	-	-
Greece	-	-	1	-
Germany	-	-	239	-
India	-	-	29	-
Singapore	-	-	27	-
Japan	-	-	-	16

BinaryEdge recorded the contents of received packets(including IP, ports and payloads) which it received by deploying honeypots all around the world. Because the honeypots do not actively interact with other devices, the data received in the honeypots are most likely send by the proves. Table 6 shows the global probe distribution of Shodan, Censys and BinaryEdge recorded by BinaryEdge during a period of 2000 days.

Table 6. Proves distribution marked by BinaryEdge

	Shodan	Censys	BinaryEdge
The United States	17	321	146
Canada	-	-	24
The British	1	-	90
In the Netherlands,	11	-	36
Iceland	2	-	-
Romania	1	-	-
Germany	-	-	115
India	-	-	8
Singapore	-	-	9

9 Conclusion

We made a comprehensive research and analysis on the well-known cyberspace search engines such as Shodan, Censys, BinaryEdge, ZoomEye and Fofa. We deeply analyze the items of Supporting internet protocols, Total amounts of detected devices, Device information, Scanning frequency, System architecture, The third party databases, Probes distribution. This paper give an objective evaluation of the detecting abilities and the working principles of the cyberspace search engines by querying official websites, analyzing APIs, and making academic research. We believe this paper will greatly help those who are developing and using cyberspace search engines.

References

1. https://api.shodan.io/shodan/protocols
2. https://support.censys.io/hc/en-us/articles/360038762031-What-does-Censys-scan-
3. https://www.zoomeye.org/doc? The channel = user# d - service
4. https://www.freebuf.com/articles/ics-articles/196647.html
5. https://docs.binaryedge.io/modules/
6. https://cli.shodan.io
7. https://censys.io/ipv4/report? Q = &
8. https://www.zoomeye.org/component
9. https://fofa.so/library
10. https://censys.io/ipv4/help/definitions? Q = &
11. Durumeric, Zakir, et al. "A search engine backed by Internet-wide scanning." Proceedings of the 22ND ACM SIGSAC Conference on Computer and Communications Security.
12. https://www.slideshare.net/balgan/binaryedge-presentationbsides
13. https://greynoise.io/

Brief Introduction of Network Security Asset Management for Banks

Yumo Wang(✉) and Qinghua Zhang

China Everbright Bank CO., Ltd, Beijing 100034, China
wangyumo@cebbank.com

Abstract. During the digital development process, enterprises have accumulated a lot of network asset including hardware, software and websites. Effective management of network asset can reduce the internet risk. Network asset is the primary object of information security. Therefore, the essential content of enterprise information security operation is ensuring the security of network assets sufficiently. This paper has investigated researches about detection, management and applications of network assets. The difficulty and current solutions have been summarized by the review. Moreover, this paper puts forward a solution of network asset management according to the bank situation.

Keywords: Network asset · Host detection · Security management

1 Introduction

With the development of technologies in information security, the demand of management of network assets is increasing in banks. During the digital construction of banks, many network assets have been accumulated including domain name, IP, system, application and so on. The network assets are the main object of information security management in banks. The purpose of managing security assets is to support the information security operation of banks, so it is essential to collect and analyze the security information of network assets. This paper conducts the research on network asset management from three aspects: detection, management and applications. A construction method of security network assets management for bank is proposed (Fig. 1).

According to the controllability of assets, the network security assets of banks are usually divided into two parts: internet and intranet. From the perspective of safety management, both of internet and intranet assets are protection objects that need to be focused on. In the management of network security assets, there are generally three aspects: detection, management and application. Detection means to discovery the security assets in cyberspace. Effective management of assets can only be achieved by timely and accurately detection. At the same time, the method of detection and monitoring are similar. Periodic updating of asset information is also an important part of safe asset management. Management means to clearly counting the proven safety assets, accumulating the detection results, so as to form an asset library that can support information security operation and provide data support for the further development of security works. The

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 215–221, 2020.
https://doi.org/10.1007/978-981-33-4922-3_16

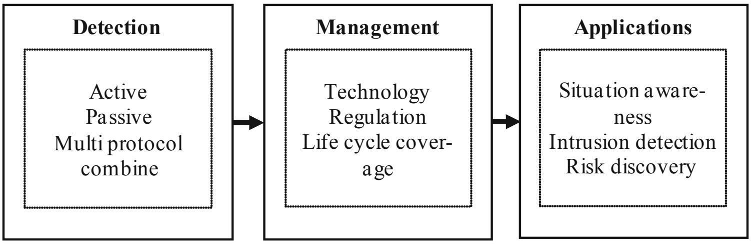


Fig. 1. Network asset management process

most important part of asset management is conducting two aspects of constructions: information and regulation. Application means using the managed network security asset data in multiple dimensions in order to embody value of it. The most typical application scenario is active risk discovery. The ability of active risk discovery for security assets can make security operation more accurate and targeted.

In view of the previous three aspects of network security asset management design, this paper conducts a literature review.

2 Detection of Network Security Assets

Detection is the starting point of network security asset management. At present, there are three common asset detection methods: active, passive and information hunting based on search engine. With the help of network scanning tools, the active way can obtain information by scanning the host, which has strong pertinence, but it will occupy part of the resources of the target host. The passive way means to the aggregation of transaction information through the carding of network traffic, which an important method in the asset discovery of intranet. Information hunting based on search engine is a non-invasive asset detection method, which can expand the collection field. However, it also depends on the data collection ability of the searching platform [1]. The detection work needs to consider different levels of assets. For the IaaS level, it mainly relies on scanners, network detection, NAT mapping table and other methods to detect network security assets. For PaaS and SaaS level, methods like traffic carding, DNS domain name aggregation are used to gather asset application information [2]. Through the acquisition of network fingerprints, the details of assets can be collected in order to identify website components, application services, communication protocols, which is able to assist the identification of vulnerabilities [3]. The design of active scanning scanner for IP requires different port scanning of TCP and UDP protocols to obtain more comprehensive host information [4, 5]. There are four scanning methods for asset discovery: ICMP, TCP connect, TCP SYN and TCP FIN. In practice, these methods are usually combined to obtain more accurate asset opening information [6]. The complex and changeable asset information needs to be monitored dynamically, and the comprehensive information including multiple dimensions as host, system, internal information should be gathered [7]. In the complex network environment, big data technology can support the asset discovery process and provide technical means for the excavation of massive information [8]. At the same time, vulnerability is also a key information in asset scanning. Periodically vulnerability mining is very important for banks [9]. In order to discovery vulnerability efficiently, automatic tools like Nmap, Xscan, Goby, Nessus are needed to discover the assets and vulnerabilities [10, 11].

3 Management of Network Security Assets

Management is the core content of network security asset management. The method of network security asset management can be divided into two aspects: technology construction and regulation construction. Empirically regulation is more important than technology in network asset management. At present, there are many problems in network security asset management, including insufficient handover, lack of sharing mechanism between different systems, untimely updating, and lack of evaluation process [12]. Though, it is very important for banks to overcome many obstacles in asset management, the management work should be appropriate considering the current situation of banks [13]. Technology construction is an indispensable method for the current network security asset management, which makes the assets fine management and strengthens the achievements of regulation construction [14]. Cloud platform is able to make the deployment of network asset management system more efficient and enable the dynamic monitoring update of asset information [15]. The integrated asset management platform usually includes account management, IP address information, resource check, electronic reminder, baseline verification, vulnerability scanning and other functions to achieve comprehensive technical function support [16, 17]. The management of network security assets needs to cover the whole life cycle of assets. The detection and monitoring needs to contain several processes like asset addition, allocation, change and invalidation. Network security assets need dynamic management, especially focusing on the changes of assets in its whole life cycle. In particular, it is necessary to check and recover the assets in time when it is out of use [18, 19]. The asset information management system based on block-chain technology makes the asset information more complete and consistent. The unchangeable characteristic of block-chain makes the asset data management process more reliable and controllable [20]. The management of network security asset data also requires multi-source fusion technology to integrate data from different sources in order to gather comprehensive information of the asset. Based on the segmentation and vectorization of address information, the cosine similarity between feature vectors is applied to assist the automatic matching and fusion of asset information [21, 22].

4 Applications of Network Security Assets

Application for security operations reflects the true value of network security asset management. The purpose of network security assets management is to find risks actively. Situational awareness system is a very practical tool in the current information security operation whose construction progress is highly associated with asset management. To enable active risk detection, many functional parts rely on the network asset management including attack detection, software and hardware information collection, external threat information and so on [23, 24]. This kind of active risk discovery has a good effect on the security of dumb terminals. For example, asset monitoring for dumb terminals such as video monitoring equipment can assist in detecting network intrusion [25]. Artificial intelligence is a potential technology in situation awareness in which asset data plays an important role and can provide data materials for situation awareness work [26]. Big data technology can also assist the network asset management in security operation. Big data technology provides sufficient storage and rapid searching for massive asset information data and enables multiple applications [27]. Big data technology provides an over-all support for comprehensive asset information management and risk discovery [28]. Vulnerability management also needs network asset management system. The whole processes of vulnerability management starts from discovering assets and includes classification, scanning, repair, tracking, compliance and periodically repetition. In the case of the asset management of FIFTH THIRD BANK in the United States, both management of network security assets security and level of compliance continuity should be paid attention in order to provide a more comprehensive guarantee for the business [29]. Asset lifecycle management can also make each data clear and controllable by assisting the work of data privacy protection which should cover the process generation, use and extinction [30]. Based on the analysis of the network flow, asset baseline is established in order to focus on the dynamic changes in data to guarantee the security of assets [31].

5 Design of Network Security Assets Management System

Based on the analysis of the relevant literature on network security asset management, current technologies and theories of network security asset management are isolated, which may be caused by the complexity of asset. Discrete management can be flexibly applied in small-scale and relatively monotonous information management but it is difficult to support complex scenarios such as information security operation with many factors. Therefore, the key of effective management of network security assets is the fusion of multi-source data. Large number of fragmented asset data need to be gathered and mixed together in order to obtain the whole picture of assets. Common asset information includes hardware, software, network, application system, organization management and so on, which involves many aspects of information about network assets (Fig. 2).

Key marking of security assets need to be focused on and be supplemented when necessary. The lack of key attribute marks will hinder the of asset management. For instance, the lack of information of the person in charge of a system will make the responsibility identification unclear. Information attributes can be roughly divided into five aspects: network, software, application, management and vulnerability. In practice, due to the partial accumulation of asset information, the management of security assets does not need start with nothing. Asset information with different attributes is generally stored in different departments of a bank. Therefore, the core problem of banks in asset management is to integrate the fragmented information comprehensively and integrate it to support the security operation. For the supplement of asset information, both detection and docking should be considered. Detecting and supplementing asset information is as important as integrating asset information from multiple channels. Moreover, asset detection is also a method of asset monitoring, which is the most important step in the whole life cycle management to protect asset information timely and accurately.

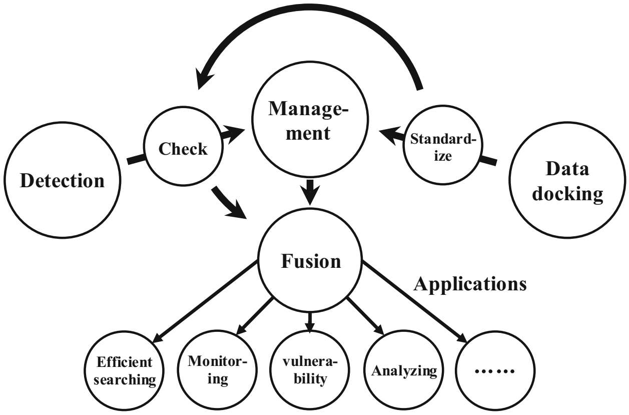


Fig. 2. Design of network asset management

The purpose of safety asset management is to find risks actively. In the multi-dimensional application of network assets, it can include: asset governance, asset full perspective, vulnerability warning, compliance inspection and so on. Asset governance means to discover unregistered assets, which is the most practical application in safe asset management. The asset full perspective means the association and display of asset data from different sources in order to provide multi-directional information for security operation. Vulnerability warning means to match the system, middleware, database, framework and other asset data in vulnerability notification. Auto POC verification tool can make the vulnerability matching more effectively. Compliance inspection means using the recorded asset information to automatically check whether assets meet the baseline regulation. With the support of comprehensive, timely and accurate asset information, security operation can be carried out more effectively.

6 Conclusions

Based on the literature review of bank safety asset management, this paper summarizes the detection, management and multi-dimensional application of asset information. A network asset management method suitable for banks is put forward. The conclusions are as listed as follows:

1) The detection of network security assets is the starting point. Comprehensive, timely and multi-dimensional detection methods can make the asset management work more effective.
2) Management of network security assets is the core. With the support of technology construction and regulation construction, network security assets can make the information security operation easier.
3) The aim of asset management is to discover risks actively and multi-dimensional application reflects the true value of management achievement. The network risk facing banks can be minimized.

4) At present, banks need to take the problem of fragmental management of data into consideration in network security asset management. It is a practical solution to fully and timely docking and fusing multi-source information from different systems.

References

1. Wang, C., Guo, Y., Zhen, S., Yang, W.: Research on network asset detection technology. Comput. Sci., 24–31 (2018)
2. Zhang, H., Wang, S., Jin, H., Deng, X.: Detection of operator network asset security management and control technology and solutions. Guangdong Commun. Technol. 5–9 (2019)
3. Yao, M., Lu, N., Bai, Z., Liu, Y., Shi, W.: Building method of device fingerprint search engine for network asset vulnerability assessment. J. Electron. 2354–2358 (2019)
4. Pei, Z., Li, B., Wang, X.: Logic processing design of IP and port scanning system. Network Secur. Technol. Appl. 26–27 (2017)
5. Ding, Y., Gao, Q., He, L.: Design and realization of assets sacn system based on complement protocol. J. Shanghai Univ. Technol. 196–200(2010)
6. Yu, X.: Design and realization of TCP/IP network scan strategy. J. Wuhan Vocational Techn. College, 54–56 (2009)
7. Li, J., Liu, P., Cai, G.: Dynamic network asset monitoring based on traffic perception. Inf. Secur. Res. 523–529 (2020)
8. Deng, X., Jin, H., Wang, S., Zhang, H.: Research on active discovery of IP assets in enterprise open network environment. Guangdong Commun. Technol. 2–4 (2019)
9. Lin, P.: Research on web risk scanning of internet information assets of postal enterprises. Postal Res. 15–17 (2008)
10. Chen, Z.: Case analysis and practice of web penetration. Network Secur. Technol. Appl. 22–24 (2020)
11. Wang, K., Li, Z., Wang, R., Gao, W., Wang, W., Wang, J.: Vulnerability scanning based on Nmap&Nessus. Commun. Power Technol. 135–136 (2020)
12. Zou, H.: Exploring the strategy of strengthening network asset management in the communication industry. Modern State-owned Enterprise Res. 49–50 (2015)
13. Li, Y.: On the role and importance of IP address planning and management in large and medium-sized enterprises. Commun. World, 20–21 (2019)
14. Wang, W.: Study on computer network security management and maintenance in hospital informatization construction. Technology, 115–116 (2020)
15. Zhang, X., Yuan, S., Ma, Z., Zhang, M., Gao, F.: Cloud-oriented asset security management scheme. Post Telecommun. Des. Technol. 12–15 (2019)
16. Xiao, Y., He, M., Wang, L.: Application research and practice of telecom operators' network asset security management technology. Guangdong Commun. Technol. (2018)
17. Song, J., Tang, G.: Research and application of network security situational awareness technology. Commun. Technol. 1419–1424 (2018)
18. Yang, X.: Thoughts on implementing dynamic management of network assets in the communication industry. Chinese Foreign Entrepreneurs, pp. 68–69 (2014)
19. Xie, R.: Lean management of optical network assets. Commun. Enterprise Manage. 24–27 (2017)
20. Zhang, S.: Network security technology based on blockchain. Inf. Technol. Inform. 129–131 (2019)

21. Chen, J.: Pre-matching scheme of network asset resources based on weighted cosine similarity. Telecommun. Technol. 46–49 (2018)
22. Lei, B.: About operation and maintenance management of IP addresses in enterprise networks. Network Secur. Technol. Appl. 106–107 (2019)
23. Yue, J.: Building an e-government network health evaluation platform based on situational awareness technology. Inform. China, 44–48 (2018)
24. Xia, Z., Li, L.: Research and design of network security situational awareness system. Inf. Commun. 147–148 (2017)
25. Li, H., Huang, X.: Illegal access detection of wireless routing equipment based on asset identification technology. China Secur. 101–105 (2019)
26. Xiao, X., et al.: A review of research on security situation prediction technology based on artificial intelligence. Inf. Security Res. 506–513 (2020)
27. Zhao, C., Sun, H., Wang, G., Lu, X.: Network security analysis of power information system based on big data. Electron. Des. Eng. 148–152 (2019)
28. Ma, Y.: Research on information security and protection of computer networks in the era of big data. Wind Sci. Technol. 82 (2020)
29. Hua, R.: Vulnerability management five: ten best practices. Instrument and Instrument, 60–62 (2016)
30. Liu, Z.: Theory and practice of Internet finance users' privacy data security. Secur. Cyberspace, 11–15 (2020)
31. Cai, G., Liu, P., Li, C.: Analysis of traffic security baseline of government websites. Inf. Secur. Res. 537–542 (2020)

Embedded Security-Critical Device Resource Isolation

Xuguo Wang(✉), Shengzhe Kan, and Yeli Xu(✉)

AsiaInfo Technologies (Chengdu), Inc., Chengdu 610000, China
{wangxg6,xuyl5}@asiainfo-sec.com

Abstract. At present, embedded devices have been widely used in people's daily life, which makes more convenience for the public. However, embedded devices still have security problems, such as automatic driving accidents that may cause casualties. In the field of embedded device security, there are many studies, for instance, OPENTEE for ARM handheld devices, providing a secure execution environment for payment devices, and SGX for Intel desk top devices, for security-critical applications, such as bank teller systems, build a safe operating environment. But it is a lack of correlation between these new and existing technologies. In our opinion, through the combination of mature technology accumulation and embedded devices, the antivirus industry can create a more secure user environment. In this paper, we propose a source isolation method to make the resources used by key processes exclusive. This method uses antivirus software and builds a more comprehensive embedded security system in critical security applications. The experimental results show that the proposed method is effective and safe.

Keywords: Embedded system · Security-Related · Resource isolation · Virus

1 Introduction

Embedded devices are becoming more and more popular in people's daily life [11], from aerospace, submarine missiles, to TV phones, watches and earphones, etc. They are everywhere, and their roles are becoming more and more important. For example, to reduce vehicles Automatic braking system for security risks.

An embedded device is usually composed of one or more hardware main bodies, and there is usually an embedded system specially developed for hardware running in the embedded device, and the system runs specifically for the device. These programs are divided into critical tasks and non-critical tasks according to different functions. For example, in the on-board embedded system, the automatic braking system is a critical task [14], and the on-board imaging system is a non-critical task. The execution of the critical task must be guaranteed, and it must be completed within the specified time from response to execution and completion. Otherwise serious consequences will occur, and non-critical tasks will be delayed for a few cycles without serious consequences. This article mainly focuses on the security issues of embedded systems with very high penetration rate-on-board systems [13].

W. Lu et al. (Eds.): CNCERT 2020, CCIS 1299, pp. 222–234, 2020.
https://doi.org/10.1007/978-981-33-4922-3_17

The original in-vehicle systems were not connected to the Internet. They usually included imaging systems, air conditioning and ventilation systems, etc. In the disconnected era, people did not have an intuitive understanding of the consequences of such devices being controlled by hackers. However, with the development of in-vehicle systems, for example, Tesla and other electric vehicle manufacturers have taken the in-vehicle systems as their main selling point. They are more humane and smarter, and even allow the driver to let go of his hands and let the in-vehicle system replace people to realize autonomous driving. Such systems are usually networked, and they are connected to non-specially constructed general-purpose networks, so the vulnerabilities in the system will be easily enlarged. According to the description in [2], such systems are more resistant to network security. Poor, and the particularity of the system makes it impossible to use common system security defense measures [3], what will happen after being controlled by hackers? At the Def Con hacking conference in August 2015, hackers claimed that they had 6 ways to control Tesla's vehicle and make it stop. Just imagine if the vehicle happened to be driving on a highway and suddenly stopped, it is likely that there will be more cars connected. Collision, causing immeasurable losses.

Do we could simply install anti-virus software in the vehicle system to solve these problems [9]? the answer is negative. Because embedded devices have relatively large limitations in storage capacity and computing performance [1], and modern on-board systems are a very complex hybrid system, they have as many as 80 control units on average [10]. Distributed in the management of multiple key tasks in the system, the operation of anti-virus software will inevitably consume storage, and may also lock up the resources occupied by key tasks. If the execution of key tasks is affected, such as an automatic braking system, it will be slightly delayed by 1 s. Zhong, stop the car, this may have caused the car crash.

In many hardware platforms, key tasks can be placed in a specific execution environment. For example, on the ARM platform, the key tasks are executed in the security domain of OPENTEE, and on the X86 platform, the key tasks are placed in the security domain of SGX. To execute, in this article, we pass Put the antivirus software in an isolated client similar to OpenTEE or SGX to run, and allocate independent resources, such as memory, cache, etc., for this isolated client. The isolated client is built through hardware virtualization features. Hardware virtualization has a very broad foundation. It is implemented on both ARM and X86 platforms, so the cost of implementation and promotion and use are of practical significance.

2 Related Work

Virtualization technology is the foundation of resource isolation technology. This technology guarantees from the bottom layer that the upper layer virtual machines can run their own processes and perform various tasks without interfering with each other. Resource isolation mainly isolates resources are CPU, memory, network and other resources. There are two types of resource isolation according to the implementation scheme: software isolation and hardware isolation.

2.1 Resource Software Isolation

Software resource isolation technology, also known as the first type of virtualization technology, mainly isolates the resources of multiple virtual machines through software simulation. This technology has the following problems and the current research status of solving these problems:

1) The issue of the authority of privileged instructions. Before the concept of hardware virtualization, the execution of all instructions was simulated by software, including privileged instructions. The way of simulation execution greatly reduced the performance of the system.
2) Compression of the process address space. Although the process can be run effectively through software simulation, the run is based on the premise that the address space is compressed, because the process-specific address is allocated to the kernel for use. If you want to fully control the process execution, you must compress the address. space. The consequence of compressing the address space is that the virtual machine's own management program needs to reserve part of the address space to store important data structures, such as IDT, GDT, etc., through these structures to manage the running image better and easier. However, if a large number of programs are compiled before that, these compressed addresses must be accessed according to the execution mode of the operation. In order to be able to process the request, performance will inevitably decrease, and if not processed, the system will crash.
3) Interrupt virtualization. The current research on this aspect has not proposed a good method to improve the way of software simulation to improve the efficiency of the mechanism.

The first type of virtualization technology mentioned above has insurmountable performance defects due to its own architecture. As shown in Fig. 1, VMM running in the user mode is the key to this problem and cannot be improved. This architecture cannot meet the real-time characteristics of the vehicle system.

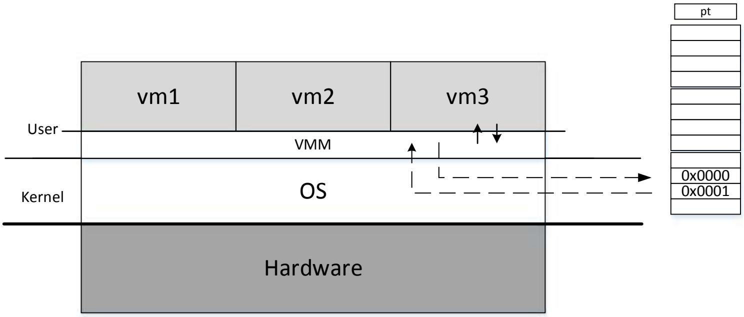


Fig. 1. Basic architecture of resource software isolation

2.2 Resource Hardware Isolation

At present, there are not many studies on resource hardware isolation technology for embedded devices, and most of them are concentrated in some research universities.

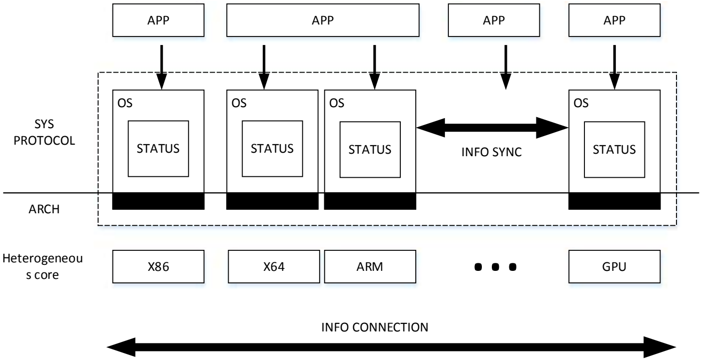


Fig. 2. Multi-core multi-operating system model

Based on the above-mentioned embedded multi-core operating system resource isolation model, this paper proposes a client solution based on a single multi-core, supporting ARM virtualization extension features, running a complete operating system per core, and running a specific strategy for each operating system.

2.3 Current Status of Research on Resource Isolation of Vehicle System

At present, most vehicle systems use SoC systems [13], in which the processor architecture uses ARM architecture [12], which provides conditions for us to implement resource isolation and sharing. By isolating the ARM processors of the SoC system according to different usage requirements For example, the driving control system is assigned to run in an independent large core, so that non-critical tasks will not interfere with him, and the control system can respond to user control in a very short time as a key task. Isolating the multimedia system into a dedicated multimedia core can achieve a better experience without affecting key tasks such as the control system.

At present, there are very few studies on resource isolation of vehicle systems that use ARM hardware virtualization extension technology. In the paper of [16], ARM's TrustZone is used to track and protect the system, but TrustZone cannot provide a complete system simulation. Need to make a lot of changes to the original system, it is more difficult to reform the system. In [17] the paper also uses TrustZone to provide a security zone for the system. The TrustZone method is more suitable for use in the context of processes, such as payment systems, fingerprint recognition systems, etc.

2.4 Current Status of Anti-virus Software Running on Vehicle Systems

Due to the particularity of the on-board system itself, it requires extremely high security. Few manufacturers consider security. By hardware isolation of resources, we can provide a completely isolated environment for the control system and install antivirus the software provides an operating environment.

The client computer installed with anti-virus software is a low-priority client. It runs only when the vehicle is stopped or is being maintained or repaired. Through such a symbiosis method, it can not only provide strong security for the vehicle, but also requests for exclusive resources that would destroy critical mission.

3 Modeling of Vehicle Resource Isolation System

3.1 Isolation Model Design

Based on the hardware characteristics described in the ARM platform architecture analysis above, the following hardware isolation model of vehicle system resources is designed.

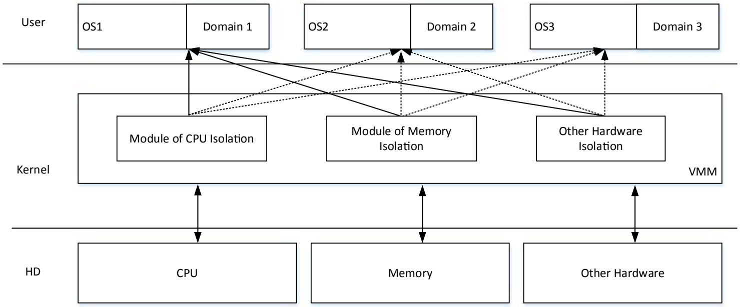


Fig. 3. Resource hardware isolation model

The entire general model is divided into three layers, namely: 1. Hardware isolation layer; 2. VMM management layer; 3. Application program operation layer.

All operating systems on this vehicle-mounted system platform run on the user layer, and the CPU isolation problem of the operating system is completed through the ARM virtual extension mechanism, so that all user-mode operating systems have their own CPU, memory and hardware devices, which are isolated from each other. Do not interfere.

3.2 Implementation and Verification of on-Board System

In the system design, we divide the system into three layers: 1. Hardware abstraction layer. 2. The virtual machine management layer is the VMM layer. 3. The application layer is the system where the control system, multimedia system and anti-virus software are installed. This chapter mainly designs the virtual machine management layer, namely the VMM layer, which provides a complete virtual environment for various applications through the VMM layer, and assigns different permissions to different applications.

3.2.1 CPU Isolation Design

The isolation of the physical processor is the most critical design [15], because if the processor resource isolation design is not perfect, then there may be the possibility of non-critical tasks affecting critical tasks, which will greatly affect the stability and reliability of the system. A big threat, especially when used in vehicle systems, has unimaginable consequences (Fig. 4).

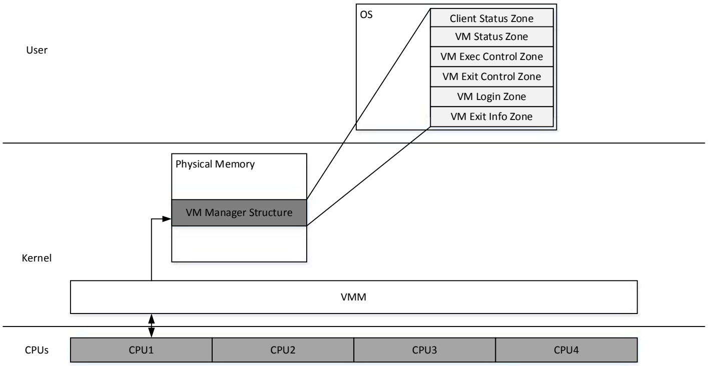


Fig. 4. Processor isolation model

Most modern ARM processors are multi-core architectures, such as ARM-v7 or ARM-v8 processors that can provide virtualization extensions, most of which are multi-core processors, and each processor can run in parallel and used in isolation, So that the applications in each client are isolated from each other without interference.

The processor isolation architecture is as shown in the figure above. The VMM corresponding to each physical processor is described by a virtual machine management structure, which includes the following parameters: 1. Client status area. 2. Host status area. 3. The VM executes the control domain. 4. The VM exits the control domain. 5. The VM enters the control domain. 6. The VM exits the information domain.

Each running client is described by this abstract structure, stored in memory, and pointed to by a special register.

3.2.2 Device Isolation Design

The vehicle system has about 80 devices with an average value that need to be controlled, so the design of device isolation is also particularly important, especially the allocation of different device controls for different clients, and the static setting of priorities.

The isolation design of the entire in-vehicle equipment is less difficult, but the workload is huge, because the consistency of the peripheral equipment is poor, and various bus standards are used, such as AHB (high-performance bus), ASB (system bus) and APB (peripheral) Bus), etc., so our current isolation scheme only isolates the AHB standard, and other standards will be considered and supported later.

1. AHB Device principle (Fig. 5)

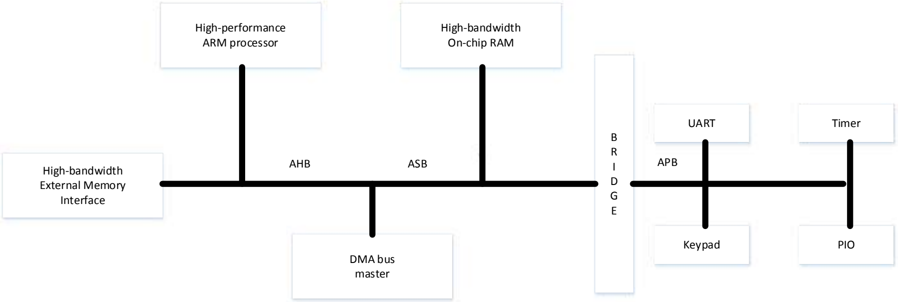

High-performance ARM processor
High-bandwidth On-chip RAM
High-bandwidth External Memory Interface
AHB
ASB
DMA bus master
BRIDGE
UART
Timer
APB
Keypad
PIO

Fig. 5. ARM's typical AMBA bus architecture

From the bus architecture in Fig. 6, AHB is in the high-speed device link. It is used by most SoC systems and is mainly used in high-performance, high-clock frequency system structures, such as high-speed RAM, NAND FLASH, etc. In the connection between DMA and Bridge, this bus is also a device bus used for critical tasks, so this isolation prioritizes the completion of device isolation on the AHB bus to ensure the reliability and availability of key tasks for vehicle equipment.

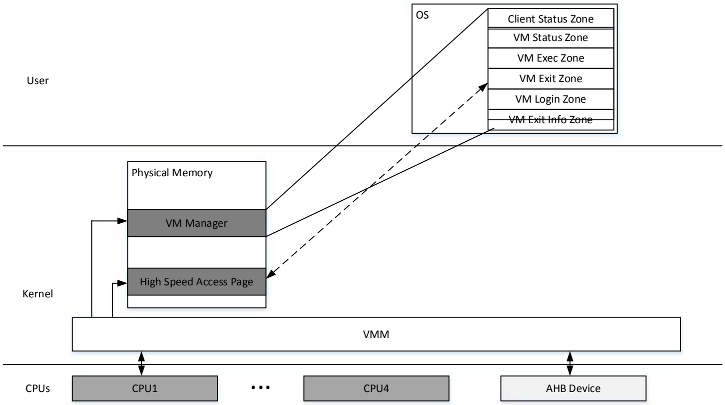


Fig. 6. AHB Device Isolation

A research statement on software reliability mentioned that errors in device drivers account for 70% or more of the entire system. The isolation of the above device drivers can also effectively reduce the system error rate and improve system reliability.

2. High-speed memory isolation

ARM has added the feature of Second-stage of translation (secondary translation) to the hardware, and supports the conversion of the physical addresses of all clients into actual physical addresses, instead of using shadow page tables (Fig. 7).

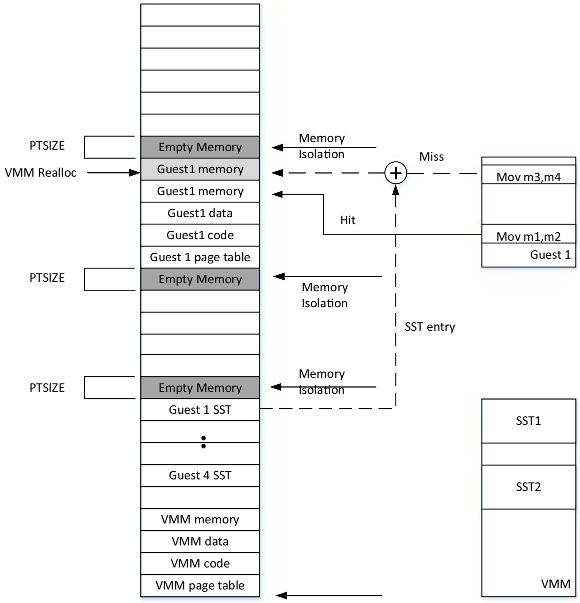


Fig. 7. High-speed memory isolation

In this paper, in the process of initializing the client by VMM, each client is allocated a completely isolated memory area. Each client maintains its own page table. When the client's GPA (Guest Physical Address) to HPA (Host Physical Address) After the query of Address) hits, with the assistance of ARM virtualization technology, there is no need to switch to the host machine. In this way, performance can be improved and the interference between different clients can be isolated.

3.3 Test and Analysis

In this vehicle-mounted resource hardware isolation system, it mainly involves the involvement and realization of CPU isolation, peripheral equipment and memory isolation, etc. This article mainly verifies the correctness of several aspects by writing a set of test cases: 1. CPU isolation characteristic test. 2. Peripheral equipment isolation characteristic test. 3. Memory resource isolation characteristic test. 4. Performance test. The tests are in two Linux clients. The test case judges whether the expected function is completed according to the result output by the terminal.

3.3.1 Test Environment

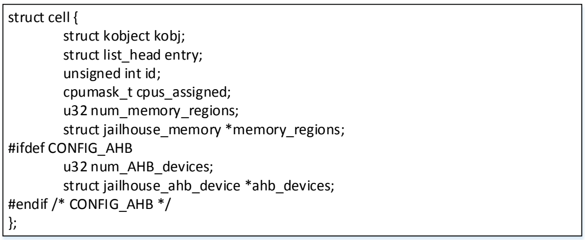
At present, the vehicle-mounted platform can already run on the NVDIA Jetson TK1 kit, but in order to obtain the visualization parameters for testing, this article runs the platform in the Qemu environment. In order to maintain consistency, all parameters are simulated NVDIA Jetson TK1 parameters (Table 1).

Table 1. TestBed

Type	Version
Kernel	Linux 4.2.0-16-generic
CPU	NVDIA Jetson TK1
AHB Bus Frequency	1000 MHz
Disk	Samsung SSD 850 EVO 120 GB
Memory	DDR3 4096 MB*2
QEMU	qemu-2.3.0

3.3.2 CPU Isolation Testing

The CPU isolation of VMM allocates a unique CPU representation structure in the memory for each client when the client is started. Each structure is maintained by VMM. In this way, the isolation is completed and the resource usage of each client will be mapped to this structure, and the allocated resources are represented by the following structure (Fig. 8):

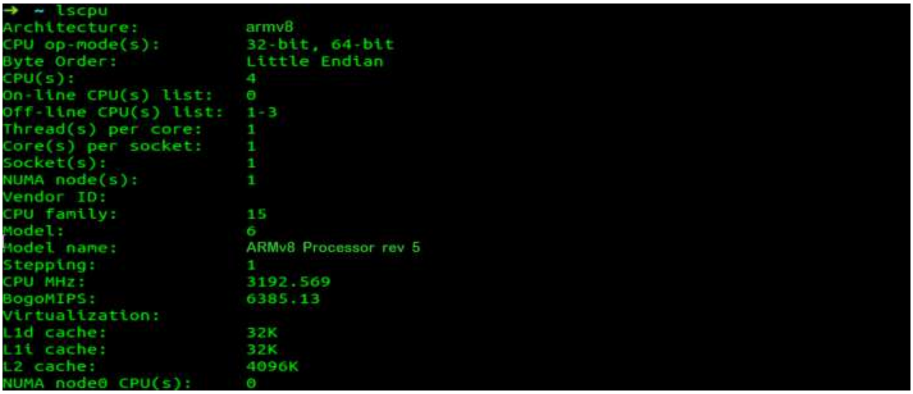


```
struct cell {
        struct kobject kobj;
        struct list_head entry;
        unsigned int id;
        cpumask_t cpus_assigned;
        u32 num_memory_regions;
        struct jailhouse_memory *memory_regions;
#ifdef CONFIG_AHB
        u32 num_AHB_devices;
        struct jailhouse_ahb_device *ahb_devices;
#endif /* CONFIG_AHB */
};
```

Fig. 8. Cell Structure

In the configuration of the client linux-arm-demo, there are a total of 4 cores, and the client occupies a total of 3 CPU. After enabling the client, 3 CPU outputs are obtained through lscpu output (Fig. 9).

There are 4 CPUs in total. After the client is started, 3 CPUs 1–3 are occupied.

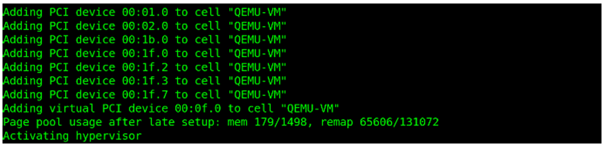


```
→ ~ lscpu
Architecture:          armv8
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0
Off-line CPU(s) list:  1-3
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             1
NUMA node(s):          1
Vendor ID:
CPU family:            15
Model:                 6
Model name:            ARMv8 Processor rev 5
Stepping:              1
CPU MHz:               3192.569
BogoMIPS:              6385.13
Virtualization:
L1d cache:             32K
L1i cache:             32K
L2 cache:              4096K
NUMA node0 CPU(s):     0
```

Fig. 9. CPU testing info

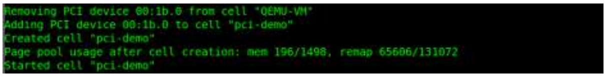
3.3.3 PCI Device Isolation Testing

After enabling QEMU-VM, all AHB devices in all systems are traversed and output to the console, as shown in Fig. 1. Enable the client PCI-demo. The configuration file of this demo only applies for a PCI device from VMM, as shown in Fig. 2. According to the expected idea, the client successfully applies for a PCI device from QEMI-VM: 00:1b.0. After applying, use the device, and after using it, release the device, as shown in the figure below, the device is successfully returned to QEMU-VM (Figs. 10 and 11).



```
Adding PCI device 00:01.0 to cell "QEMU-VM"
Adding PCI device 00:02.0 to cell "QEMU-VM"
Adding PCI device 00:1b.0 to cell "QEMU-VM"
Adding PCI device 00:1f.0 to cell "QEMU-VM"
Adding PCI device 00:1f.2 to cell "QEMU-VM"
Adding PCI device 00:1f.3 to cell "QEMU-VM"
Adding PCI device 00:1f.7 to cell "QEMU-VM"
Adding virtual PCI device 00:0f.0 to cell "QEMU-VM"
Page pool usage after late setup: mem 179/1498, remap 65606/131072
Activating hypervisor
```

Fig. 10. List PCI Devices In different VMs

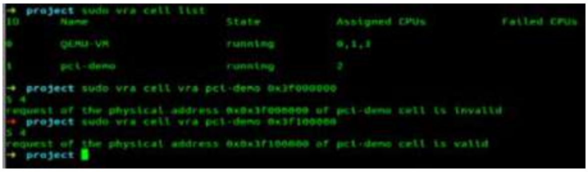


```
Removing PCI device 00:1b.0 from cell "QEMU-VM"
Adding PCI device 00:1b.0 to cell "pci-demo"
Created cell "pci-demo"
Page pool usage after cell creation: mem 196/1498, remap 65606/131072
Started cell "pci-demo"
```

Fig. 11. Remove the PCI Device from a VM

3.3.4 Memory Isolation Testing

Enter the address belonging to the client and return the valid result. Enter a physical address that does not belong to the client, and invalid is returned. As shown in the figure, the output result of the vra program is the same as the expected result, indicating that the memory isolation is executed successfully, and different clients can only access their assigned physical address space, but cannot access the address space of other clients (Fig. 12).

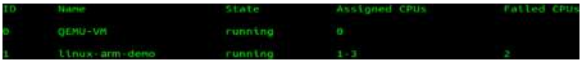


```
→ project sudo vra cell list
ID      Name                State           Assigned CPUs           Failed CPUs

0       QEMU-VM             running         0,1,3

1       pci-demo            running         2

→ project sudo vra cell vra pci-demo 0x3f000000
5 4
request of the physical address 0x0x3f000000 of pci-demo cell is invalid
→ project sudo vra cell vra pci-demo 0x3f100000
5 4
request of the physical address 0x0x3f100000 of pci-demo cell is valid
→ project
```

Fig. 12. Memory usage

3.3.5 Install a Simple Behavior Detection Engine Testing

First run the linux-arm-demo client in VMM, which is used to run the anti-virus software runtime environment. As shown in Fig. 1, next run a behavior detection engine in the client. As shown in Fig. 2. As shown in Fig. 3, the client runs a centos image, which contains basic shell tools that can perform read and write operations. This also provides a runtime basis for the behavior detection engine that needs to be run in this experiment (Fig. 13).



```
ID      Name                State           Assigned CPUs           Failed CPUs

0       QEMU-VM             running         0

1       linux-arm-demo      running         1-3                     2
```

Fig. 13. VMs engine running status

The behavior detection engine can start, run, and detect. After starting the mirror, you can read and write files in the mirror. It can be seen from Fig. 3 that the behavior detection engine can run the program normally, download the centos image and start it, and the image contains the engine that can read and write basic files, reaching expectations.

4 Conclusion

In this article, we propose a new mechanism to make the new security technologies works with the existing ones. The experimental results shows it. But the system is a rudimentary form. If the virus detection and killing mechanism want to perfectly operated in modern vehicle systems, many other measures are needed to involved, to improve the reliability of the system, such as the division of resources and priority. For the classification of processes' levels, the system itself also needs to pass the certification of security standards, but this scheme effectively uses the current general virus detection and killing mechanism, which reduces the time cost of this research and makes it easier to use previous successful experience.

Acknowledgments. I would like to thank AsiaInfo Technologies (Chengdu), Inc for its support in this research, not only for the sufficient arrangement of time, but also for the support of hardware and personnel. At the same time, I would like to thank my chief Yeli Xu, my boss Liang Yu, and my colleague Shengzhe Kan assisted in the development.

References

1. Zhou, S.: On embedded network security technology. Network Security Technology and Application (in Chinese)
2. Wang, Z.: Research on network security technology based on embedded operating system. Network Security Technology and Application (in Chinese)
3. Du, G., Wang, L.: About the security analysis of embedded network firewall. Network Security Technology and Application (in Chinese)
4. Lutz, R.: Analyzing software requirements errors in safety-critical. Embedded Syst., 126–133 (1993). https://doi.org/10.1109/isre.1993.324825
5. Kane, A.: Runtime Monitoring for Safety-Critical Embedded Systems (2015)
6. Nejati, S., Alesio, S.D., Sabetzadeh, M., et al.: Modeling and analysis of CPU usage in safety-critical embedded systems to support stress testing. In: International Conference on Model Driven Engineering Languages and Systems. Springer, Heidelberg (2012)
7. Popek, G.J., Goldberg, R.P.: Formal requirements for virtualizable third generation architectures. Commun. ACM **17**(7), 412–421 (1974)
8. Krammer, M., Martin, H., Karner, M., Watzenig, D., Fuchs, A.: System Modeling for Integration and Test of Safety-Critical Automotive Embedded Systems, 2 (2013). https://doi.org/10.4271/2013-01-0189
9. Ding, Y.: Thinking of virus killing mechanism of embedded devices. Information Security and Communication Secrecy (in Chinese)
10. Luo, J., Hubaux, J.-P.: Embedded Security in Cars. Embedded Security in Cars – Securing Current and Future Automotive IT Applications (2005). https://doi.org/10.1007/3-540-28428-1_7
11. Babar, S., Stango, A., Prasad, N., et al.: Proposed embedded security framework for internet of things (IoT). In: IEEE International Conference on Wireless Communication. IEEE (2011)
12. Othman, N.A., Aydin, I., Karakose, M.: An efficient embedded security system for reduce car accident to build safer world based on IoT. In: 2019 International Artificial Intelligence and Data Processing Symposium (IDAP) (2019)
13. Ali, S., Al Balushi, T., Nadir, Z.: Embedded Systems Security for Cyber-Physical Systems (2018)
14. Kevan, T.: Facing the embedded security crisis. Desktop Eng. **23**(9), 16–19 (2018)
15. Nathi, R.A., Sutar, D.S.: Embedded payload security scheme using CoAP for IoT device. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN) (2019)
16. Ye, H.: Security protection technology of cyber-physical systems. Int. J. Security Appl. **9**, 159–168 (2015)
17. Dong, P., Han, Y., Guo, X., Xie, F.: A systematic review of studies on cyber physical system security. Int. J. Security Appl. **9**, 155–164 (2015)

Author Index