PROCEEDINGS OF SCIENCE

The Detection of Web Abnormal Scan Behaviors based on Cluster Analysis

Dongming Bai
Research Institute of Petroleum Exploration & Development-Northwest(NWGI) PETROCHINA
Lanzhou, 730020, China
*Email:*baidm@petrochina.com.cn

Mei Feng
Department of Computer Application Technology, Research Institute of Petroleum Exploration & Development
Beijing, 100083, China
*Email:*fm@petrochina.com.cn

Liang Chen[a]; Xiaodong Guo[b]
Research Institute of Petroleum Exploration & Development-Northwest(NWGI) PETROCHINA
Lanzhou, 730020, China
Email:[a]chen_l@petrochina.com.cn; [b]guoxd@petrochina.com.cn

Scan is the most common technical means used by hackers to identify site vulnerabilities as an attack entry to a website. Local and lightweight scan can often avoid the detection for network layer security protection. The establishment of detection algorithms against such hidden abnormal scan can enable timely identification of the vulnerability of an application site so to establish a precise active protection strategy. Through the comparison on the access behaviors of various users based on the behavioral characteristics of abnormal scan summarized and the clustering algorithm of the subdomain of the site, the occurrence time of abnormal scan and the location of the subdomain can be detected. The results show that the higher the degree of overlap of characteristic operation indexes, the higher the probability of being an abnormal scan behavior. This helps greatly reduce false positives during the overall detection of the website. Based on the output of the clustering-based detection model, it provides a strong basis for enhancing the protection of the application system and repairing security vulnerabilities caused by the inherent logic errors and the incomplete system functionality.

ISCC2017
16-17 December 2017
Guangzhou, China

PoS(ISCC 2017)023

1.Introduction

Web scan plays a very important part in the preparatory work for hackers to initiate network attacks on web application systems, which is not destructive, but can obtain system data or hide latent APT attacks. Web scan can not only enable a quick collection of information about the site structure, Web links and so on, but also works as an initial detection of the security vulnerabilities on the web application page, which provides a foundation for the subsequent intrusion behaviors of the hacker such as web application system vulnerability analysis and network penetration.

This paper describes the behavior characteristics of abnormal web scan in the first place, and then analyzes and refers to the idea of attacking the mainstream detection methods and makes up for its shortcomings. To follow that, an IP address set of good-intention users is established, based on the data analysis and mining of the web log. Finally, the comparison on characteristic indexes of various users, and the cluster analysis algorithm for differentiated applications are adopted to detect web abnormal scan behaviors.

2.Web Scan

Web scan is a software technology for automatic page information collection from websites via computer programs. Using the loop and recursive methods, it traverses the nested relationship of webpages, reads the webpage link and page content on the site per page and per layer, and thus get access to the page address or site content of the entire site.

3.Feature Extraction

3.1HTTP Protocol and Related Contents

(1) HTTP attribute

While using the HTTP protocol to access website, the site's application software also records the user's access behavior and generates a web log. Many fields are stored in the log, from which the "Status" and "Method" fields are particularly important in the web scan detection.

Status identifies the HTTP request state, whose code is divided by 100-600, where 100-300 represents the normal state and 400-600 represents the abnormal state. Method identifies the webpage access mode, where GET represents the page that has been read and POST represents submitting data to the site.

(2) URL

URL is the abbreviation of Uniform Resource Locator, which is defined as the address of the identification resource on the internet. The HTTP protocol uses URL to access the web application. URL consists of the root domain, subdomain, page name and parameter domain.

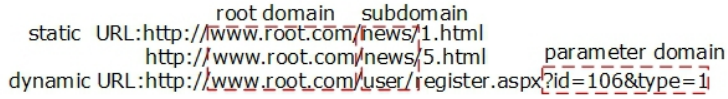


Figure 1: URL constitution

3.2Behavior Classification

Per motivation, the site scan behavior can be classified as follows: one for the purpose of content collection and the other for the purpose of vulnerability detection. The site scan behavior for the purpose of vulnerability detection is to explore the logic errors or

shortcomings in the site source code, which is often taken advantage of by hackers and is difficult to be timely found. Through the comparison in Table 1, it is observed that the web scan behavior for the purpose of vulnerability detection is comparatively challenging.

	Content collection	Vulnerability detection
Form	Software automation	Software automation
Behavior	Read page information	Read page information and submit test data
Focus	Collect website content	Collecting vulnerability information
Target	Content retrieval service	Security detection and network attack

Table 1: Web Scan Behavior Comparison

3.3Feature Comparison

As a typical representative of the normal scan behaviors, a search engine scan will first get the site's robots file, and then traverse site content according to the root domain or subdomain defined in the file. However, the abnormal scan will increase HTTP requests in POST mode on the website form page, to simulate the SQL injection, cross-site request forgery, malicious upload and other attacks. At the same time, the short address, statement splicing, script construction, parameter deformation and other approaches are used to generate a URL to detect web applications. These behaviors increase the frequency of HTTP POST mode and state anomalies, which will leave a record in the web log as a result (Fig. 2).

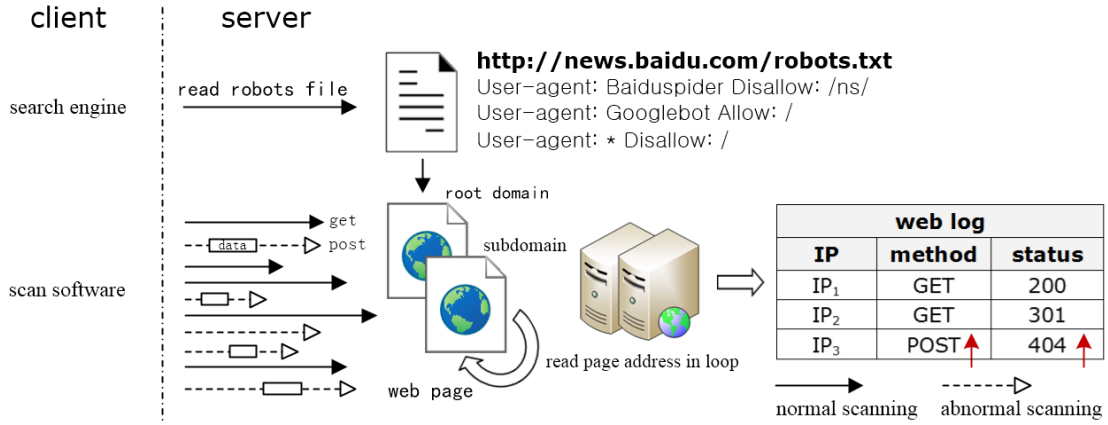


Figure 2: Overall Frequency of Requests within A Day

Consequently, the behavioral characteristics of the normal scan and the abnormal scan are obtained. The number of concurrent requests in the normal and abnormal scan behaviors will be increased greatly. The number of subdomains in the web application is usually large, and the starting position of the abnormal scan is unpredictable. The POST mode and the error code of abnormal scan will obviously increase, compared with normal scan.

Category	Starting position	Concurrent request	HTTP Status	HTTP Method
Normal scan	Usually starting at the root domain	Enlarge	Maintain normal	GET
Abnormal scan	Root domain or subdomain	Enlarge	More than 400; increase	GET and POST

Table 2: Web Scan Behavior Comparison

4.Anomaly Detection Algorithm

Threshold, K-means, Isolation Forest and local outlier factor are the most representative algorithms for anomaly detection, with which many researches and applications have been done in many fields.

4.1Threshold

This algorithm calculates the threshold by standard deviation and historical mean to

detect abnormality; it reduces misdiagnosis rate and omits judgement rate by balancing various methods. Fig. 3 shows an example of network data detection, the maximum and minimum thresholds of a time are calculated by the following formula:

$$T = u - c \times \sigma, T = u + c \times \sigma \tag{4.1}$$

Where μ is the expected historical data, reflecting the average level of the current monitoring point; σ is the standard deviation, reflecting the normal fluctuation range of the monitoring point; c is the weight, which can be adjusted according to demand. The algorithm complexity can be reduced in time and space, mainly through the calculation of the threshold size, so that the application of this method in specific scenarios can be completed. It can be concluded that this method is not universal. For example, for complex and changeable monitoring indexes, the accuracy of the algorithm is poor, so it is necessary to establish an algorithm for automatic threshold selection.

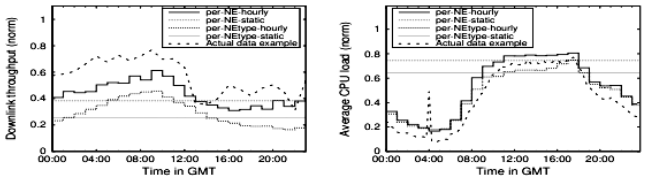


Figure 3: Comparison of Results at Different Thresholds

4.2K-means

Based on the K-means [2], the data points in the learning data set are clustered as the normal and abnormal classes by the weighted distance formula of their different KPIs (such as BPS, PPS, source-destination IP address logarithm and other performance indexes):

$$d(x, y) = \sqrt{\sum_{i=1}^{m} \left(\frac{x_i - y_i}{S_i} \right)^2} \tag{4.2}$$

The specific anomaly detection method is shown in Fig. 4, where the point far from the normal class (exceeding the threshold d_{max}) or close to the anomaly class is detected as an outlier.

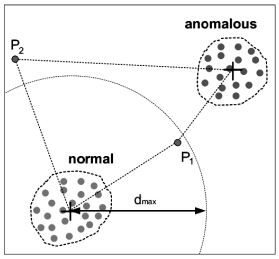


Figure 4: Outlier Determination

Fig. 5 shows an example of flood attack detection. As can be seen from the above figure, the network bandwidth suddenly increases during the attack. In the corresponding figure below, the distance between each point and outlier cluster exceeds the threshold, and the anomaly is detected successfully.

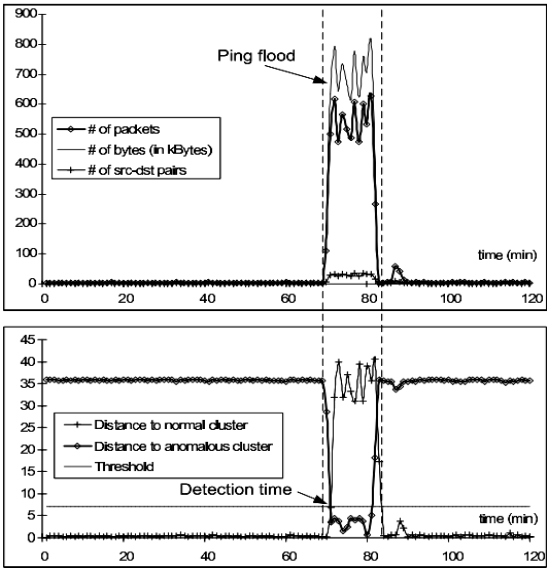


Figure 5: Detection Example

4.3Isolation Forest

In the method of isolated forest [3], a feature is randomly chosen to perform data segmentation, and a value is selected randomly between the maximum and minimum values of the feature. The data smaller than the value is categorized into the left branch, and the data equal to or larger than the value is categorized into the right branch. Then the above steps are repeated in the two branches until the data cannot be subdivided or the binary tree reaches the depth limit to measure the leaf nodes that have obvious difference from other leaf nodes. Similarly, the sample number at the leaf node where x is located is $T.size$, and $h(x)$ represents the path length (depth) of the data x.

$$h(x) = e + C(T.size) \tag{4.3}$$

Where, e represents the number of edges that data x passes through from the root node to the leaf node of iTree; $C(T.size)$ is a correction value representing the average path length of the binary tree. The calculation formula of $C(n)$ is as follows, where $H(n-1)$ is the Euler constant, and the closer the *Score* is to 1, the higher the probability of being an outlier will be.

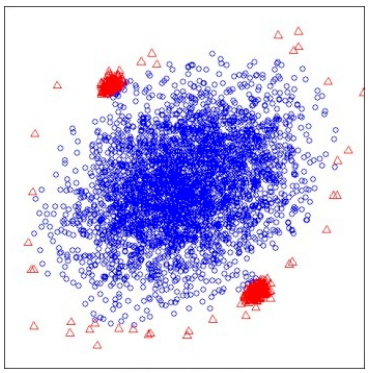


Figure 6: Detection Example

4.4Local Outlier Factor

The local reachability density algorithm [4,5] detects outliers by measuring the relative local density of the nearest points of the data. The distance between k nearest points and Point p is denoted as *k-distance(p)*. The reachable distance between *reach-dist (p,o)* of the data point p and the data point o is the maximum value between the k-distance of data point o and the direct

distance between data point *p* and point *o*, shown as below:

$$reach\_dist_k(p,o) = \max\{k\text{-}distance(o), d(p,o)\} \tag{4.4}$$

The definition of local reachability density is based on the reachability distance. The data points whose distance from Point *p* is less than or equal to *k-distance(p)* are called its k-nearest-neighbor, denoted as $|N_k(p)|$. The local reachability density of data point p is the reciprocal of the average reachability distance between it and the adjacent data points.

$$lrd_k(p) = \frac{1}{\frac{\sum_{o \in N_k(p)} reach\_dist_k(p,o)}{|N_k(p)|}} \tag{4.5}$$

The local outlier factor measures the relative density of the neighboring data points. The local outlier factor is the ratio of the average local reachability density of the neighboring data points of point *p* to the local reachability density of point *p*:

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd(o)}{lrd(p)}}{|N_k(p)|} = \frac{\sum_{o \in N_k(p)} lrd(o)}{|N_k(p)|} / lrd(p) \tag{4.6}$$

If the LOF of data point *p* is much larger than 1, it might indicate that data point *p* is alienated from other points and may be an outlier, as shown in Fig. 7.

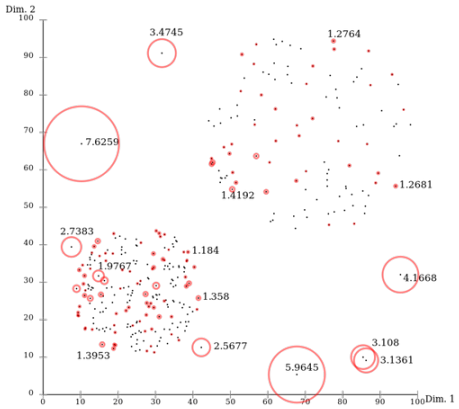


Figure 7: Detection Example

In addition, the current algorithm used for anomaly detection includes the Holt Winters [6], time series decomposition [7, 8], P2P behavior detection using the moving average [9] and etc. Holt Winters uses the stability and regularity of time series to predict the value by seasonality and trend; it distinguishes the abnormal value by comparing the real value with the predicted value. Time series decomposition predicts value by the combination of factors such as long-term trend seasonal variation, cyclic variable and irregular variable to seek the abnormality. The other major algorithms for abnormality detection shall be referenced in literature [10].

The above anomaly detection methods are applied in a specific scenario to detect the scan behavior of websites. In order to hide their behaviors, hackers usually perform centralized scan on a few virtual paths or individual pages of the site. Therefore, it is necessary to identify the local scan behavior, and the detection algorithm needs to be built for fine-grained detection.

5.Model Design

5.1Data analysis

(1) The URL domain that can be enumerated

The web application system requires 6 domains (i.e., the strings split by "/" in the URL path) to cover 99% of its requests, and the URL with a small number of domains is mostly a

simple index page. In addition, the request is more likely to hit a URL with the number of domains of less than 10 (see Fig. 8). This result shows that the same public domains of URL are few and enumerable, regardless of the wide range of each domain of URL and the large total number of URLs.

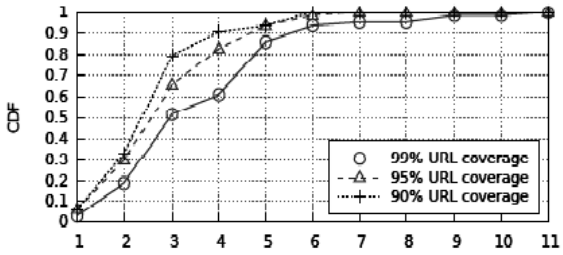


Figure 8: Cumulative distribution of URL in the Same Public Domains

(2) Home access of web application system

Take the access data of the homepage of a web application system for example. An access to the root domain URL contains 65 HTTP GET requests, including the page file, script and image; and each request corresponds to a file; the submission of data once corresponds only to one HTTP POST request; and each HTTP request will generate a log record on the server. It can be concluded that the number of requests in the GET mode is much larger than that of requests in the POST mode in Web log record (Fig. 9).

Method	Type	URL
GET	application/x-javascript	http:/ ndexcount.js?Version=201612121551
GET	application/x-javascript	http:/ ndexLogin.js?Version=201612121551
GET	image/gif	http:/ jif
GET	image/gif	http:/ jif
GET	image/gif	http:/ jif
GET	image/gif	http:/ jif
GET	image/png	http:/ x/bl.png
GET	image/png	http:/ x/tl.png
GET	image/png	http:/ x/b.png
GET	image/png	http:/ x/tr.png
GET	image/png	http:/ x/br.png
GET	image/png	http:/ og_highlight-hard_100_f2f5f7_1x100.png
65 requests		
POST	*	http:/ lexlg.aspx
1 request		

Figure 9: Home Page Access of A Web Application System

(3) Difference on access data between subdomains

The subdomain of the web application system is generally classified according to the service function, and the page distribution is quite different. Through the tracking of user access, there is a difference on the access to subdomains, and the difference is significant between some subdomains with concentrated function distribution and other subdomains (Fig. 10).

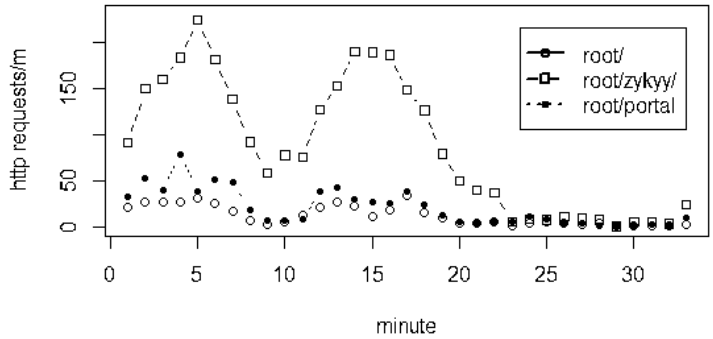


Figure 10: Difference on the Access to Subdomains

5.2Design Approach

Those who have access to a web application system are system users, business-related

users, browsing users, and potential attackers. All user behaviors will leave traces in the web log. The access behavior of system users is a manual operation, in contrast with the behavior characteristics of scan. Through the establishment of characteristic indexes, the access behavior of system users can be regarded as the baseline to detect the scan behavior and judge whether it is the abnormal based on various indexes. In addition, the subdomain access of the web application system is different, and the scan is not necessarily initiated in the root domain of the site. Therefore, it is necessary to compare the characteristics of subdomains to detect all the scan behaviors. The design approach is as shown in Fig. 11.

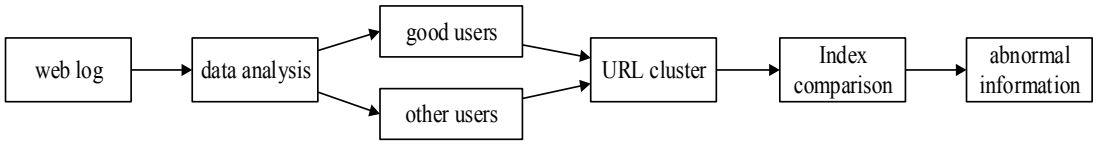


Figure 11: Design Idea

5.3User Screening

Within an organization, a good-intention user refers to the others that use the application system to complete their own work. Their behavior is regular and normative, and the behavior characteristics are of reference value. Based on the service features of the enterprise, it is found that a good user has the following two characteristics:

(1) The access frequency per month

An enterprise-level application system is built to meet the needs of business development. The establishment of the application system is geared to the needs of a business area and specific system users. The daily work of the system users is inseparable from the application system. Thus, the access to and the operation of the application system is continuous. Since a month is taken as the time unit in the enterprise's production plans, business activities, economic settlement, and tax returns and so on, it is advised to screen system users based on the monthly access frequency.

(2) Data submission action contained in the operation behavior .

System users in an enterprise, especially the ones in the key business location, are bound to submit relevant business data to the system while operating the application system, rather than just to browse. The data-mining algorithm APRIORI is used to mine the IP and GET frequent item sets in the user behavior, to verify the user access behavior. Based on the data analysis 1, the HTTP request for an access to the home page of the system accounts for $1/10^6$ of the number of log entries generated by the system. Thus, the minimum support of this algorithm is set to $1/10^6$, in order to get the maximum number of system user IP sets. And the confidence (i.e., the proportion of GET in the frequent item sets) is set to 0.4. With the one-year log as input, it can be seen from the output of the algorithm that as the user's access to the system increases, the confidence gradually decreases at first and then tends to be steady, indicating that the user's POST action is gradually increasing. Therefore, the system user with the confidence of 1, with their behaviors in GET mode, is removed to obtain the good user's IP set (Fig. 12).

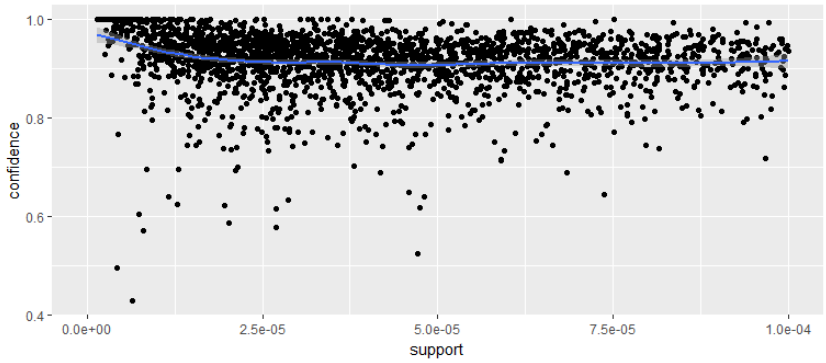


Figure 12: Fitting of the Support and Confidence of IP and GET Frequent Item Sets

5.4Clustering Algorithm

According to data analysis, the subdomains in the URL are enumerable and limited, and each URL is a set of domains. If there is a question mark in the domain at the end, the contents of the parameters after the question mark will be filtered out. Thus, a URL *u* can be expressed as: *F(U) = {f1, f2, ..., fi}*, where the subscript of each domain *f* is its position in *u*. The distance between two URLs (*u* and *u'*) is denoted by *Dist (u, u')*. The number of public domains is the maximum value of public domains obtained through a comparison of domains in the order from the root domain to the subdomain. The distance between URLs can be represented by the number of public domains, so that the element distance function *Dist* is expressed as:

$$Dist(u,u') = \begin{cases} \infty & , if\ F(u) \cap F(u') = \varnothing \\ \dfrac{1}{\max(index)} & , otherwise\ \sum_{index=1}^{n} (f_{index} - f'_{index}) = 0, n \in \left[1, \min(|F(u)|, |F(u')|)\right] \end{cases} \quad (5.1)$$

The distance function Dist indicates that when two URLs have more public domains, the distance between the two URLs is shorter. The clustering algorithm uses distance function Dist and hierarchy of the URL structure to finish URL clustering. The algorithm uses temporary cluster to identify cluster processing node and cluster completed by user to identify the final cluster node. L(T) denotes the level of URL; meanwhile, the distance between any two nodes is less than or equal to L/L(T). The root node will be created as a parent node of any cluster node of the L(T) = 1.

DC(T1,T2) is used to denote the distance between any two cluster processing node(T1 and T2).The data analysis shows that the level L is a limited value, when a URL is inputted, the algorithm will initialize it by F(URL) and create temporary cluster in every level as F(T), and |F(T)|≥L(T). All the temporary cluster nodes that have the same parent node will be merged. The condition of clustering is:

$$\begin{aligned} &DC(T_1,T_2) = Dist(T_1,T_2) \\ &Dist(T_1,T_2) \le \frac{1}{l}, where\ l(T_1) = l(T_2) = l \end{aligned} \quad (5.2)$$

The url x/2 and x/y/z will be merged at the first level in cluster algorithm, but the url m/n only has the same parent node which is the root node with other nodes, as shown in Fig. 13

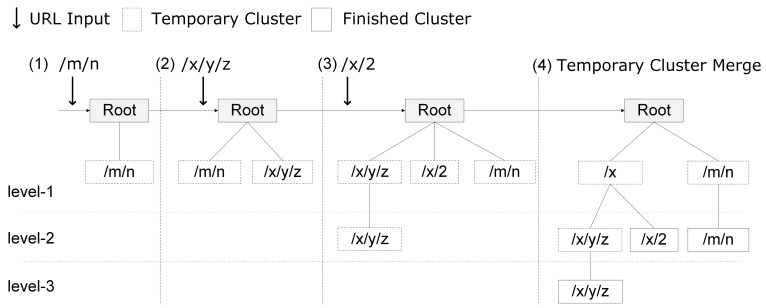


Figure 13: URL Clustering Process

The clustering algorithm is expressed as follows:

```
Algorithm uCluster(URL u, L, N)
current = root
for d = 1 to min(L, |F(u)|) do
    merge_flag = false
    if current is not finished_CLUSTER then
        for each CLUSTER e in current DESC by his
        do
            if |F(e) ∩ F(u)| ≥ d then
                current = c
                merge_flag = true
                F(c) = F(e) ∩ F(u), break
            end if
        end for
        if merge_flag == FALES then
            if |F(current)| == N then
                set current FINISHED_CLUSTER, break
            else
                TEMPORARY_CLUSTER t = F(u)
                current.add_temporary_cluster(t)
                current = t
                if d == min(L,|F(u)|) then
                    set current FINISHED_CLUSTER
                end if
            end if
        end if
    end if
end for
```

5.5Anomaly Detection Method

The time series is composed of time slots, with a minute as the unit time, and the comparison on the operating indexes of good-intention users and other users is monitored. In the same period, when the behavior index of other users is higher than that of good-intention users and has a large overlap, the probability of the abnormal scan behavior is higher (Fig. 14).

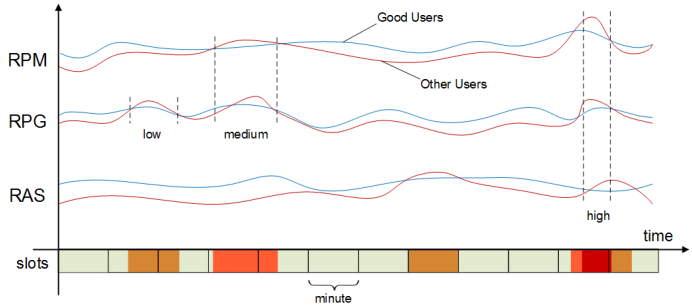


Figure 14: Anomaly Detection Method

(1) Detection time slot: 1 minute by default;

(2) RPM (requests per minute): average number of requests per IP per minute;

(3) RPG (ratio of POST and GET): the ratio of the average number of POST to GET per IP HTTP request per minute;

(4) RAS (ratio of abnormal and normal status): the ratio of the average of abnormal to

normal state per IP HTTP per minute.

5.6Overall Model

(5) The model uses the middleware application log as input, and parses it into structured data that contains client IP, URL, HTTP status, HTTP method and visit time. The IP Addresses of good-intention users will be selected by the two principles of user screening. The model uses the algorithm to build URL clusters, the time slot sequence will be established in each cluster, three indexes of RPM, RPG and RAS are used to detect the running status of website, the logs that don't show the good-intention user record will be outputted as abnormal information.

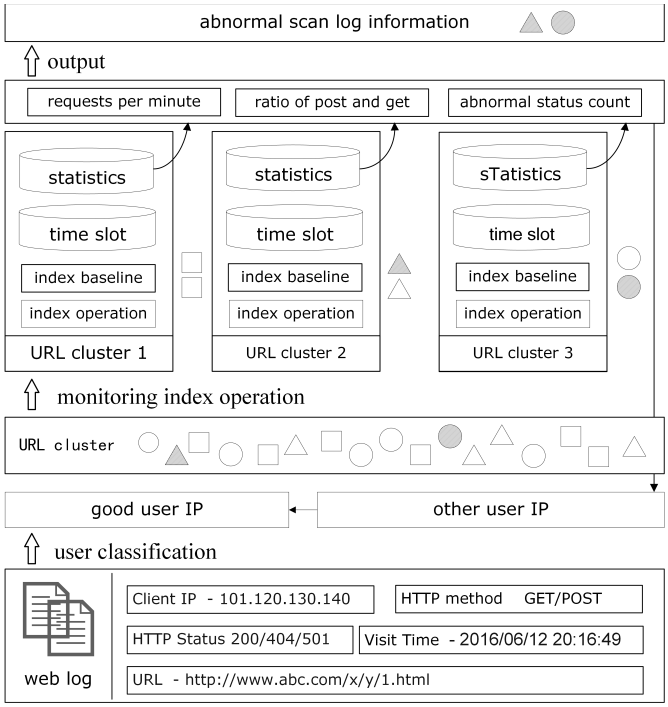


Figure 15: Functional Architecture of the Model

6.Inadequacy of Overall Particle Size

The detection process of the statistics on the total number of HTTP requests on the site is shown below. During the monitoring, the number of requests from other users exceeds that from the good users on many occasions, rendering it impossible to timely identify and accurately locate abnormal scan behaviors. If only local scan and lightweight scan are performed, the number of requests from other users will not exceed that from the good users. In this way, no abnormal scan behavior will be accurately and timely identified through overall detection of the site. Therefore, the post-clustering subdivision feature detection of the URLs maintains the sensitivity as well as fast and accurate positioning of the scan behaviors.

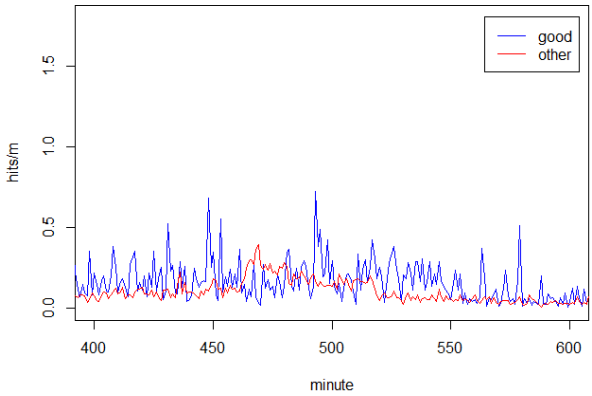


Figure 16: Situation of Overall Site Monitoring

7.Application Effect

The model can effectively avoid false positives and accurately identify the web anomaly scan behavior by comparing the operation status of the three indexes and the correlation of time periods, during which the web log data of the same subdomain of the application system in different time slots are input in the model.

Fig. 17 shows a comparison on the index operation of a subdomain in an application system using the model on a business day in February 2014. The operating index of other users exceeds that of good users in a time interval in the RPG index, but the operation of other indexes does not show such a situation in the same time interval. No abnormal situation was found in the detection of the log in this time interval.

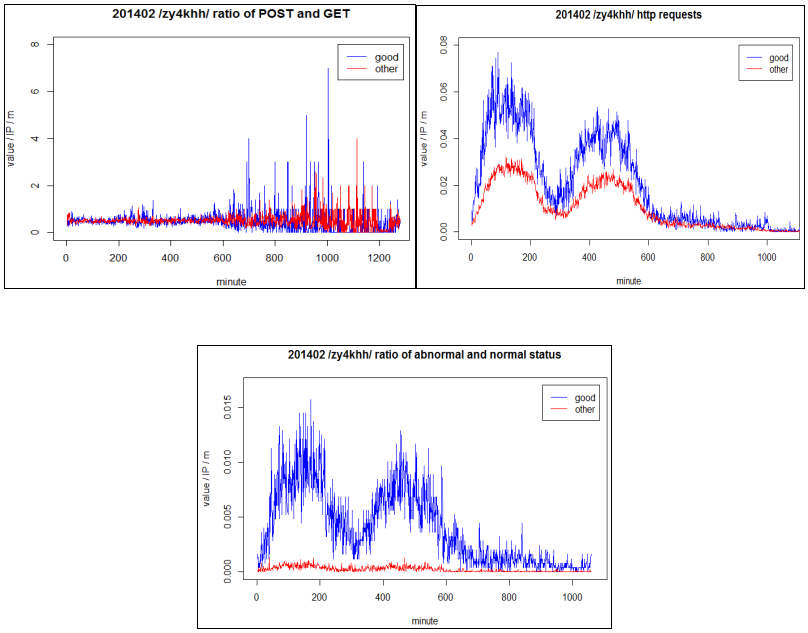
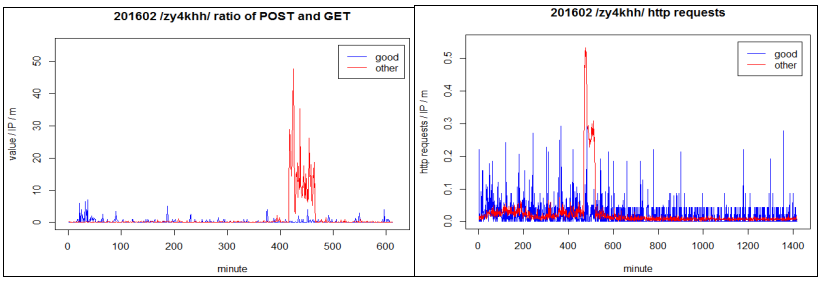


Figure 17: Comparison of RPG RPM RAS Index Operation in A Subdomain (February 2014)

Fig. 18 shows a comparison on the index operation of a subdomain in an application system using the model on a business day in February 2016. It is found that the three detection indexes of other users exceeded those of good users in the same time interval.



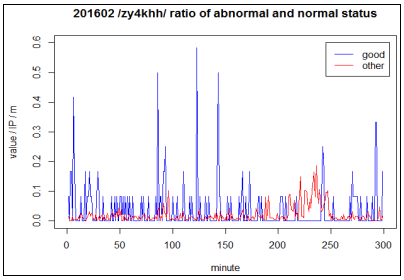


Figure 18: Comparison of RPG RPM RAS Index Operation in A Subdomain (February 2016)

By viewing the web log of the index overlap intervals, it is found that in a short period, an IP sends a large number of HTTP requests to some URLs of the application system. The POST request is sent to the same URL multiple times, and in a request, the parameter domain of URL takes the server operating system file of the target website as the request content rather than a web page. In conclusion, this is obviously an abnormal behavior (Fig. 19).

	date	time	method	uri	query	username	ip
18985	2016-02-21	07:59:21	POST	/zy4khh/chemicalserch.aspx	-	-	10.
18986	2016-02-21	07:59:21	GET	/zy4khh/topiclist.aspx	PortalCall=..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f.....	-	10.
18987	2016-02-21	07:59:21	POST	/zy4khh/chemicalserch.aspx	-	-	10.
18988	2016-02-21	07:59:21	POST	/zyportal/externalaccidinforserch.aspx	-	-	10.
18989	2016-02-21	07:59:21	POST	/zy4khh/chemicalserch.aspx	-	-	10.
18990	2016-02-21	07:59:21	POST	/zy4khh/chemicalserch.aspx	-	-	10.
18991	2016-02-21	07:59:21	GET	/zy4khh/topiclist.aspx	PortalCall=../../../../../../../../../../windows/win.ini%00...	-	10.
18992	2016-02-21	07:59:21	POST	/zy4khh/chemicalserch.aspx	-		
18993	2016-02-21	07:59:21	POST	/zy4khh/chemicalserch.aspx	-		
18994	2016-02-21	07:59:21	POST	/zy4khh/chemicalserch.aspx	-		

PortalCall=../../../../../../../../../../windows/win.ini%00.jpg&title=%E8%AE%BA%E6%96%87&type=100

Figure 19: Abnormal Web Log

8.Conclusion

The security management of the application system is the key to identifying and confirming the abnormal behavior, but it is impossible for the system security manager to monitor the safe operation of multiple application systems all the time. As a part of system security management, the web abnormal behavior detection model based on clustering can locate the abnormal scan behavior intended for site penetration attack and further avoid error alarming largely. Through a comparison on index operation of users after screening and classification, this model can effectively detect the abnormal scan behavior, which can provide a powerful reference for improving the security protection of the application system and repairing the security vulnerabilities caused by the logic errors and functional deficiencies of the application system.

References

[1]. Lee S B, Pei D, Hajiaghayi M, et al. *Threshold compression for 3G scalable monitoring*[C]//INFOCOM, 2012 Proceedings IEEE. IEEE, 2012: 1350-1357.

[2]. Münz G, Li S, Carle G. *Traffic anomaly detection using k-means clustering*[C]//GI/ITG Workshop MMBnet. 2007.

[3]. F. T. Liu, K. M. Ting and Z. H. Zhou，*Isolation-based Anomaly Detection*，TKDD，2011

[4]. M. M. Breunig, H. P. Kriegel, R. T. Ng, J. Sander. *LOF: Identifying Density-based Local Outliers*. SIGMOD, 2000.

[5]. M. Goldstein. *FastLOF: An Expectation-Maximization based Local Outlier detection algorithm*. ICPR, 2012

[6]. M Szmit,A Szmit,S Adamus,S Bugała. *Usage of Holt-Winters Model and Multilayer Perceptron in Network Traffic Modelling and Anomaly Detection, International Journal of Bio-Medical Computing* , 2012 , 5 (4) :313-314

[7]. H. Liu, and M.S. Kim, *Real-Time Detection of Stealthy DdoS Attacks Using Time-Series Decomposition,* Communications (ICC), 2010 IEEE International Conference, 2010:1-4.

[8]. Leszek Borzemski and Maciej Drwal. *Time series forecasting of web performance data monitored by mwing multiagent distributed system*. In ICCCI (1), 20–29, 2010.

[9]. Choffnes D R, Bustamante F E, Ge Z. *Crowdsourcing service-level network event monitoring*[J]. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 387-398.

[10]. Chandola V, Banerjee A, Kumar V. *Anomaly detection: A survey*[J], 2009, 41(3): 15