International Journal of Security and Its Applications
Vol. 10, No. 8 (2016), pp.49-58
http://dx.doi.org/10.14257/ijsia.2016.10.8.06

Terminal Anomaly Detection System Based on Dynamic Taint Analysis Technology

Wang Yutong[1, a], Chang Chaowen[1, b] and Han Peisheng[1, c]

[1] *Henan Key Laboratory of Information Security, Zhengzhou Information and Science Technology Institute*
[a]wyt87380345@126.com, [b]ccw@xdja.com, [c]hps97430031978@163.com

Abstract

With the rapid development of computer systems, intrusion attack methods have become large-scale, distributed and complex. Traditional protection means such as vulnerability database, virus database and rule matching can't cope with the attacks hidden inside the terminals. This paper proposed a terminal anomaly detection system based on dynamic taint analysis technology from the data dimension of the terminals. Firstly we built a standard data path model based on HMM and evaluated the deviation degree of the current operating mode with it to find the abnormal working status of the terminals. The experimental results show that the structure is valid to discover the intrusion attacks with a high detection rate and low false alarm rate.

Keywords*: Anomaly detection system; location feature; standard data path model; dynamic taint analysis technology*

1. Introduction

With the rapid development of computer systems, Intrusion attack methods are being large-scale, distributed and complex with more serious information security issues. Various information security techniques have been proposed to solve these problems such as firewall, intrusion detection, vulnerability mining and anti-virus technology. However, the security techniques above are mostly focused on the external characteristics of the terminals by mode matching and characterization of the attacks. These passive techniques are too late to find the attacks which have caused serious damage so as to prevent the terminals fell to the same kinds of attacks. For the superior attacks which we called "Advanced Persistent threats (APT)" such as "Aurora", "Stuxnet", "Duqu" and "Flame", the techniques can't play a role. Eventually, the superior attacks caused incalculable economic losses to the enterprise and even posed a serious threat to the national security.

According to the essence of the attacks, we can see that the ultimate aim of each attack is to cause the entire network or terminals to enter an unintended working status with abnormal changes. For example, after "Stuxnet" is activated, it will attack SCADA systems, and modify the programmable logic controllers (PLC). Then it will manipulate the PLC to send control commands which will lead a disorder of the industrial control systems [1]. Therefore, the focus of the defense structure should be the internal characteristics of the terminals itself rather than the external characteristics.

In order to characterize the internal characteristics of the system, firstly we need to fully understand the business system workflow. Then we need to establish a standard operating model with continuous training and correction to reflect the normal working status of the system more fully and accurately. At last, we should continuously monitor the business system based on the standard model to become aware of attacks attempting to cause abnormal changes in terminals.

ISSN: 1738-9976 IJSIA

Data As a carrier of the information flow in terminals, its movement can directly reflect the current working status of the system. In this paper, we selected the logical address as the location feature of the data and establish a standard data path model based on Hidden Markov Model. Then we proposed a terminal anomaly detection method based on dynamic taint analysis technology. This method can detect the deviation degree from the standard path mode and gradually accumulate the degree to discover the attacks deeply hidden in terminals.

2. Terminal Anomaly Detection Method based on Dynamic Taint Analysis Technology

DTA (Dynamic Taint Analysis) is a dynamic analysis method of information flow. The main function of this technology is to record how the data runs and changes in the program. The ultimate aim is to identify the dependence between objective data and source data. DTA includes three aspects: marking, spreading detection of the tainted data. Tainted data marking means the marking module marks external untrusted data from the internet or other external file as “tainted data”. When the data is marked as “tainted data”, the results of various calculations are also not credible. So these results also need to be marked with a "tainted" property. This process is taint spreading. However, we also need to determine the spreading policy of the tainted data to give a precise definition of the operations which cause the taint spreading. The tainted data detection means to judge whether the data is contaminated and identify its source by checking the memory variables or registers.

At present, DTA is widely used in the field of software vulnerability analysis, but it is rarely applied in anomaly detection. In this section, DTA was used to record the spreading process of external data after it entered the system. Also, the status transition stored in the tainted data structures was regarded as the focus of analysis to model the standard data path. HMM (Hidden Markov Model) was used to model this path. The tainted data structure is a structure for recording related information of tainted data such as the number of system calls, current stack mapping and memory address. Finally, LIF trigger model was used to calculate the accumulation deviation degree to discover the abnormal working status of terminals.

2.1. Standard Data Path Model based on Hidden Markov Model

Hidden Markov model (HMM) [9] is a collection of mutually transferable finite states. It’s a double random process. One is used to describe the transfer of implicit state and the other one is used to describe the statistical relationship between the states and the observed values. HMM has three assumptions: the current status is only relevant to the previous status, the probabilities of states transfer is unrelated to time, the observed value is only relevant to current states. These three assumptions greatly reduce the complexity of the model. The following modeling process consists of three steps: determining model parameters, selecting location features and training model.

2.1.1. Determining Model Parameters: In order to establish a standard data path model based on HMM, the model will be described as a triad $\lambda=\{A,B,\pi\}$ where $A=(a_{ij})$ represents the state transition probability matrix, $B=\{b_j(k)\}$ represents the probability distribution of observed values and $\pi=\{\pi_i\}$ represents the initial probability distribution. The following parameters need to be determined through the training:

- Number of finite states N;
- State set $\phi=\{S_1,S_2,...S_N\}$;

- Transition probabilities a_{ij} in state transition probability matrix, the calculation method shows in formula (1):

$$a_{ij} = P(s_{t+1} = S_j \mid s_t = S_i) = \frac{P(s_{t+1} = S_j, s_t = S_i)}{P(s_t = S_i)} \tag{1}$$

- Observed values probability $b_j(k)$ the calculation method shows in formula (2) where V_t represents the observed values at state S_j:

$$b_j(k) = P(V_t \mid s_t = S_j) \tag{2}$$

2.1.2. Location feature quantification: In order to describe the flow paths of external data in terminals, some data features need to be selected and stored in the tainted data structure. These features should be able to describe the location of data in terminals at a certain moment. Then the features will be quantified as the major states in HMM.

In order to select a feature to describe the data location information in spreading process, we need to understand the spreading modes of data in a computer. The spreading of data in a computer includes three modes: Target mode, Initiator mode and DMA mode, target mode and Initiator mode are also called direct spreading mode. Target mode is the traditional program IO mode which the CPU communicates with PCI devices by program interruption; Initiator mode is the way of communication commands between PCI devices. That means a PCI device access other devices as a PCI master on the same bus; In DMA mode, PCI devices can efficiently transfer data directly to the memory without CPU.

For external data (tainted data), its spreading mode is target mode which means the program IO mode after it enters the computers and memory is the only place for programs running. The essence of data transfer and computing is the process to move data from a physical memory block (or register) to another one or spread to other ones after computation. So the addresses of physical memory can be characterized as the location feature of data at one time during the spreading process. However, the physical address is mapped by the logical address, so we select the logical address as the location feature of data.

In order to improve the efficiency of the model training, logical addresses should be quantized. Since in Win32 paging memory management, the ranges of decimal numbers which are corresponded to the three portions of logical address are 0-2^{10}, 0-2^{10} and 0-2^{12}, formula (3) was used to quantify the logical addresses. According to the spatial locality principle of programs, the quantitative results of similar logical addresses should be approximately equal.

$$\Omega = 100\log_2 d_i + 10\log_2 p_i + \log_2 o_i, \Omega \in [0,1112] \tag{3}$$

In formula (3), Ω represents the quantitative result of logical address in the range of [0-1112], d_i represents the decimal number which is corresponded to the index of page directory; p_i represents the decimal number which is corresponded to index of page table; o_i represents the decimal number which is corresponded to page offset. When the tainted data is transferred in the terminals until offset, storage or output, its location features will be quantified as a sequence of discrete values and this sequence can describe the transfer path of the tainted data in the terminals.

2.1.3. Model Training: When the terminal is under normal working status, we use dynamic taint analysis techniques to chronologically collect amounts of discrete values of location features generated in the terminals and the discrete values will compose the observation sequence collections. Then we will train the HMM model with these collections. In this paper, we use the Baum-Welch revaluation algorithm to train the model. We define λ as the initial status of the model and O as the sequence of observed values. λ will be iteratively computed by reconstruction model λ^* and the parameters will be continuously adjusted until they satisfy the convergence condition $P(O|\lambda) > P(O|\lambda^*)$. Formula (4) and (5) show the reconfigurable computing methods.

$$a_{ij} = \sum_{t=1}^{T-1} \xi_t(i, j) / \sum_{t=1}^{T-1} \xi_t(i) \tag{4}$$

$$b_{jk} = \sum_{t=1 \cap O_t = V_k}^{T} \xi_t(i) / \sum_{t=1}^{T-1} \xi_t(j) \tag{5}$$

In formula (4) and (5), $\xi_t(i,j)$ represents the probability when HMM in state S_i; $\sum_{t=1}^{T-1} \xi_t(i)$ represents number expectations that the model transfers from state S_i to others; $\sum_{t=1}^{T-1} \xi_t(i, j)$ represents number expectations that the model transfers from state S_i to S_j. the training process is shown in Figure 1

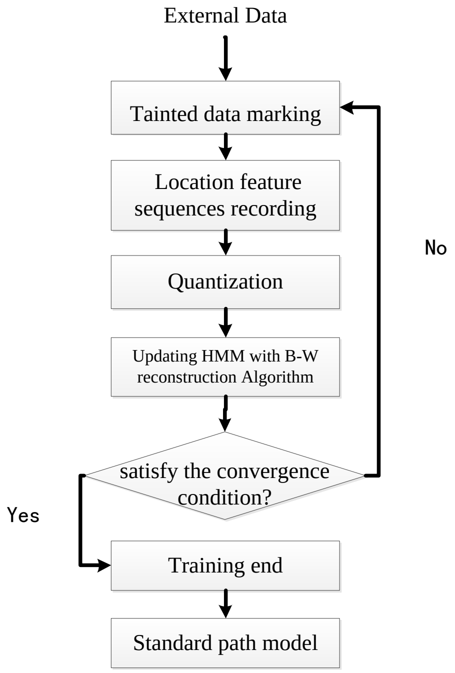
External Data
Tainted data marking
Location feature sequences recording
Quantization
Updating HMM with B-W reconstruction Algorithm
satisfy the convergence condition?
No
Yes
Training end
Standard path model

Figure 1. The Training Process of Standard Path Model

2.2. Terminal Anomaly Detection Method

The terminal anomaly detection process can be understood as the matching problem between current data spreading path and standard spreading path.

At time t, we define the distance between observed location feature value V_t and the location feature value V_t^* in standard path model as $dist(V_t^*, V_t)$. In order to describe the gradual deviation process of data from the standard model, we use the leaky integrate-and-fire model (LIF) to calculate the accumulation of the deviation degree and make a judgment about whether the system deviates from the standard path model according to the calculation results. Here, we select Hellinger distance (HD) as the main indicator. We use the following formula (6) to calculate the Hellinger distance:

$$dist(C_t^*, C_t) = \sum_{i=0}^{n-t} (\sqrt{C_t^*[i]} - \sqrt{C_t[i]})^2 \tag{6}$$

In the formula, $V_t^*[i]$ represents one feature value in the location feature sequence of standard path model; $V_t[i]$ represents one observed value of the current data. W represents the size of the sliding window.

Figure 2 shows the schematics of the leaky integrate-and-fire model (LIF). The model can integrate the system input in a period of time, then it will trigger some response based on the integration results. *I(t)* is the drive current which is the input of the model. When the capacitor *C* is charged by the drive current, resistor *R* is constantly consuming the current. Therefore we get the formula (7):

$$I(t) = I_R(t) + I_C(t) = \frac{u}{R} + C\frac{du}{dt} \tag{7}$$

The integration result at time *t* can be calculated by the following formula (8):

$$u(t) = u_r \exp(-\frac{t}{RC}) + \frac{1}{C}\int_0^t \exp(-\frac{s}{RC}) I(t-s)ds \tag{8}$$

Since under normal working status, the deviation distance $dist(V_t^*, V_t)$ of the system does not always equal to 0, In order to reflect the accumulation of deviation more accurately and tolerate the deviation distance under normal work conditions, we calculated the average deviation distance $dist(V_t^*, V_t)_{\mathrm{ave}}$ according to the historical records. Then the D-value between $dist(V_t^*, V_t)$ and $dist(V_t^*, V_t)_{\mathrm{ave}}$ will be used as the drive current *I(t)* which is the input of the LIF. We defined ε as the alarm threshold. If the integration results *u(t)* satisfied the alarm condition $u(t)\geq\varepsilon$, that means the current data spreading deviates from the standard path model.

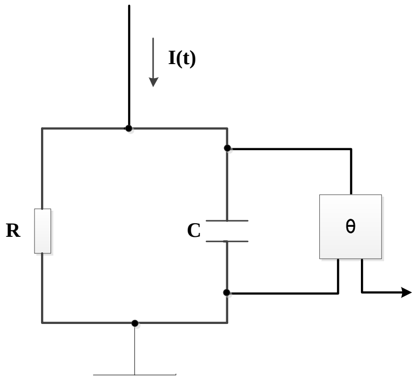

Figure 2. The Schematics of the Leaky Integrate-and-fire Model (LIF)

2.3. The Architecture of Terminal Anomaly Detection System

Figure 3 shows the architecture of the terminal anomaly detection system based on dynamic taint analysis technology. The system consists of three components: dynamic taint analysis system, terminal anomaly detection system and decision & display system.

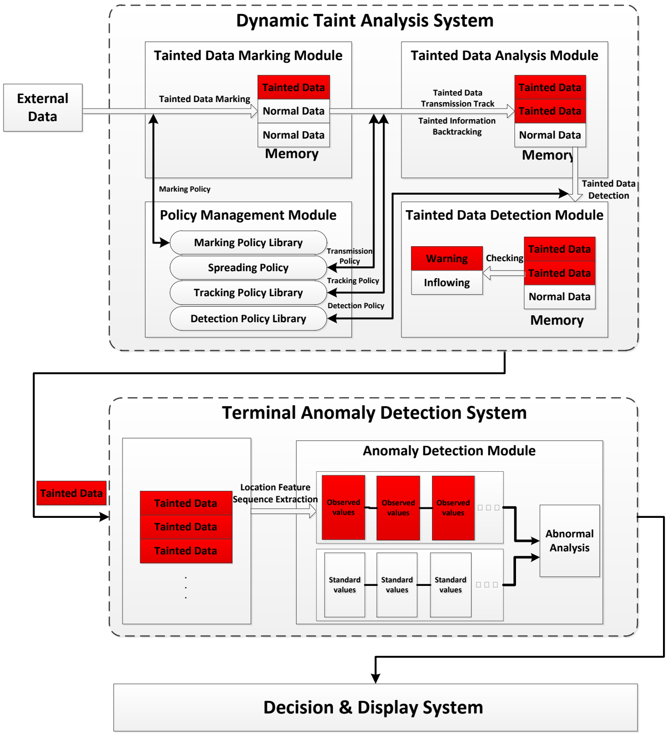

Figure 3. The Architecture of Terminal Anomaly Detection System

2.3.1. Dynamic Taint Analysis System: Dynamic taint analysis system consists of four modules: policy management module, tainted data marking module, tainted data analysis module and tainted data detection module.

- Policy Management Module

Policy management module consists of marking policy library, spreading policy library, tracking policy library and detection policy library. They respectively provide specific policies to related modules.

- Tainted Data Marking Module

According to the marking policy, tainted data marking module establishes a one to one mapping relationship between external untrusted data and marked with Tainted marks and allocate tainted data structures to the tainted data to get the spreading of tainted marks and backtracking of tainted information.

- Tainted Data Analysis Module

According to the workflow of business software, tainted data analysis module models the business commands based on the semantics of assembly instructions to realize fine-grained taint tracking of instruction-level and tainted mark spreading. Its function is to track the use trail of the tainted data in memory and spread the tainted marks based on the spreading policy.

- Tainted Data Detection Module

The function of this module is to detect the critical data such as return address, function pointer, parameters and return values. Then it will generate a warning to the tainted data, other normal data will flow into the next work aspect.

2.3.2. Terminal Anomaly Detection System: Terminal anomaly detection system consists of two modules: tainted data cache and abnormal analysis module.

- Tainted Data Cache

When the upper layer system has detected the tainted data, tainted data will be backed up and stored in the tainted data cache. The function of tainted data cache is to exchange tainted data with abnormal analysis module.

- Abnormal Analysis Module

The function of this module is to analyze the working status of the terminals with the comparison method between current data path and standard path. The abnormal detection method is given in Section 3.2. If the current path deviates from the standard path, the terminal is proved to have security risks.

2.3.3. Decision and Display System: The function of D&D system is to give a judgment of the current security status of the terminal and visualizes the results for artificial analysis and monitor the working status of the terminal.

3. Experiment and Analysis

We implemented this architecture on a Win32 system and used the public dataset KDD 99 as the experimental data set to validate the detecting effect of the anomaly detection system. The data set was from an intrusion detection evaluation project which was sponsored by DARPA at MIT Lincoln Laboratory. Each data in KDD99 had the last item which description indicated whether the data was normal or abnormal. Each sample was described by 41 the other features such as duration, protocol_type and service etc. The original data consisted of two parts: seven weeks of training data and two weeks of test data. Among them, the data sets on Tuesday in the third week, Thursday in the fifth week and first three days in seventh week didn’t have any attack occurred. These sets can be used as the training set and normal test set.

In order to ensure the efficiency and effectiveness of the experiment, redundant features or unimportant features were removed in data preprocessing. We selected the data set of first four days which had no attacks occurred as the training set. The test set consisted of two parts: normal test set with the data set of Wednesday in seventh week and Abnormal test set with all abnormal data. The hardware environment of this experiment is Core I7 processor, 1TB hard disk and 8GB memory. The experiment consisted of two parts: simulated training experiment and anomaly detection experiment based on MATLAB7.0.

3.1. Simulated Training Experiment

The main purpose of the training experiment is to validate the model updating formula and determine the relevant parameters of HMM $\lambda = \{N, \phi, A, B, \pi\}$. Parameter calculation is the core issue of model training. From the training method we can see that the value of states number *N* directly determines the length of discrete sequence, the scale of the matrix *A* and *B*, so *N* will greatly influence the training efficiency and accuracy of the model. However, if the value of *N* is too large or too small, the model can’t accurately reflect the data transfer path. Here we introduced the parameter μ to evaluate to model accuracy. Formula (9) shows the calculation method of μ:

$$\mu = \frac{k_n}{K} \times 100\%, n \le N \tag{9}$$

In the formula, k_n represents the number of samples which have *n* states in their lifecycle, *K* represents the total number of samples.

In the experiment, we selected the values of sliding window size *W* are 5-10 and analyzed the number of states *N* based on the experiment results. Table 1 shows the correspondence of the training time *t*, accuracy evaluation parameter μ and the sliding window size *W*:

Table 1. Results of Training Time and Accuracy

W	*t*	μ
5	*1.34s*	*68.6%*
6	*1.86s*	*75.6%*
7	*2.35s*	*78.8%*
8	*3.67s*	*85.6%*
9	*7.88s*	*88.3%*
10	*13.46s*	*91.3%*

From the experimental results, we selected eight as the sliding window size because the model had a high fitting degree of the valid information in the training set and training efficiency was ensured. So the model was trained to get other parameters with sliding window size *W*=8.

3.2. Anomaly Detection Experiment

The main purpose of the anomaly detection experiment is to validate the detection capability of the anomaly detection method proposed in this paper and determine the value of the alert threshold ε. The value of the threshold ε can directly affect the experiment results. The main test indicators include:

- Detection rate (DR) = total number of detected abnormal samples / total number of abnormal samples.
- False alarm rate (FAR)= the number of normal samples detected as abnormal ones / total number of normal samples.

In the experiment, the data set of Wednesday in seventh week C1 was used as the test set of the false alarm rate experiment; All the abnormal data set C_2 was used as the test set of the detection rate experiment; Figure 4 shows the results of these two experiments. We can see, with alarm threshold ε increasing, FAR is gradually reducing, but DR is also reducing. When $\varepsilon \geqslant 30$, with alarm threshold ε increasing, DR is rapidly reducing, but FAR has little change. In order to take into account both DR and FAR, we selected ε=30 to finish the experiment. At this time, DR was 91.47% and FR was 6.23%. DR was relatively high and FAR was acceptable.

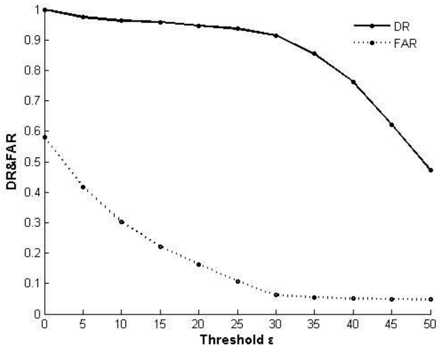

Figure 4. Changes of DR and FAR with Alert Threshold ε

In kdd99, abnormal data can be divided into four categories: PROBE, DoS, R2L and U2R, we gave four experiments for these four abnormalities to test the detection performance of the architecture. We selected the alert threshold ε = 30. The experiments results were shown in the ROC plot in Figure 5:

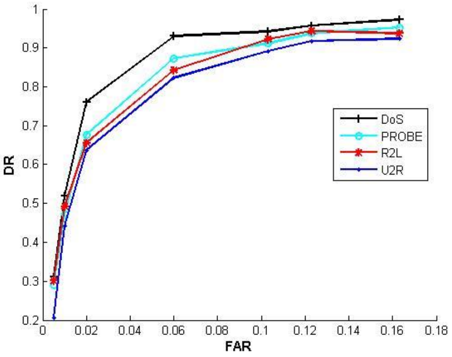

Figure 5. Detection Performance of PROBE, DoS, R2L and U2R

3.3. The Experimental Results

As it can be seen from the above results, the number of limited states *N* can greatly affect the training of standard path model. When *N*=8, the training can take into account both efficiency and accuracy. The alarm threshold *ε* can adjust DR and FAR. When *ε*=30, DR and FAR can meet the requirements; This architecture has a good detection performance of these four categories of abnormalities and the detection performance of DoS is the best.

4. Summary

This paper proposed a terminal anomaly detection system based on dynamic taint analysis technology. Firstly we built a standard path model of data flow based on HMM and proposed a terminal anomaly detection method based on LIF model. This method can reflect the gradual accumulation process of the deviation from the standard path model. Then we proposed the architecture of the anomaly detection system and the working methods of the modules. The experiments show that the system had a good detection effect on abnormal states. In the future, we will have a further research on the portability of the system and apply the system to the actual business terminals to test its detection performance. According to the actual application, we will improve the detection performance of the system.

References

[1] Langner R., “Stuxnet: Dissecting a cyberwarfare weapon[J]”, Security & Privacy, IEEE, vol. 9, no. 3, **(2011)**, pp. 49-51.

[2] L. Zheng, P. Zhou, Y. Jia and W. Han, “How to Extract and Train the Classifier in Traffic Anomaly Detection System [J]”, Chinese Journal of Computers, vol. 4, **(2012)**, pp. 719-729+827.

[3] Y. Zhu, J. Yang and J. Zhang, “Anomaly Detection Based on Traffic Information Structure[J]”, Journal of Software, vol. 10, **(2010)**, pp. 2573-2583.

[4] J. Zhao, H. Huang and S. Tian, “Protocol Anomaly Detection Based on Hidden Markov Model[J]”, Journal of Computer Research and Development, vol. 4, **(2010)**, pp. 621-627.

[5] C. Li, X. Tian, X. Xiao and M. Duan, “Anomaly Detection of User Behavior Based on Shel Commands and Co-Occurrence Matriux[J]”, Journal of Computer Research and Development, vol. 9, **(2010)**, pp. 1982-1990,

[6] Y. Li, D. Li, T. Cui and H. Li, “Real-time detection framework for network intrusion based on data strearn[J]”, Journal of Computer Application, vol. 2, **(2015)**, pp. 416-419+429.

[7] N. Sun, Y. Guo and Y. Yao, "Network data stream abnormal detection model based on SVM incremental learning method[J]", Computer Engineering and Application, vol. 29, **(2012)**, pp. 78-81+205,

[8] Kang M. G., McCamant S. and Poosankam P., "DTA++: Dynamic Taint Analysis with Targeted Control-Flow Propagation[C]", //NDSS, **(2011)**.

[9] Blunsom P., "Hidden markov models[J]", Lecture notes, vol. 15, **(2004)**, pp. 18-19.

[10] Bhuyan M. H., Bhattacharyya D. K. and Kalita J. K.., "Network anomaly detection: methods, systems and tools[J]", Communications Surveys & Tutorials, IEEE, vol. 16, no. 1, **(2014)**, pp. 303-336.

[11] Anandapriya M. and Lakshmanan B., "Anomaly Based Host Intrusion Detection System using semantic based system call patterns[C]", //Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on. IEEE, **(2015)**, pp. 1-4.

[12] Murtaza S. S., Khreich W. and Hamou-Lhadj A., "A trace abstraction approach for host-based anomaly detection[C]",//Computational Intelligence for Security and Defense Applications (CISDA), 2015 IEEE Symposium on. IEEE, **(2015)**, pp. 1-8.

[13] Hong B., Peng F. and Deng B., "DAC-Hmm: detecting anomaly in cloud systems with hidden Markov models[J]", Concurrency and Computation: Practice and Experience, **(2015)**.

Authors

Wang Yutong, male, he was born in Jilin, China, in 1990. He received the B.S. degree, in Beijing Institute of Technology in 2013 He is doing his M.E degree in Zhengzhou Information and Science Technology Institute, China. His research interests focus on network and information security, network security situation awareness. (wyt87380345@126.com)

Chang Chaowen, male, he was born in Henan, China, in 1966. He is currently a professor and PhD Tutor in Zhengzhou Information and Science Technology Institute. He is a senior member of CCF, ad hoc expert in the field of mobile information security of National Ministry of public security. His research interests focus on network and information security, cloud computing. (ccw@xdja.com)

Han Peisheng, male, he was born in Hebei, China, in 1978. He received the Engineering PhD in Beijing University of Technology in 2015. He is a lecturer in Zhengzhou Information and Science Technology Institute, China. His research interests focus on network and information security, Access control.
(hps97430031978@163.com)