DATA DRIVEN DETECTION STRATEGY ENGINE FOR BETTER INTRUSION DETECTION ON CLOUD COMPUTING

Daniel W.K Tse  
*City University of Hong Kong, iswktse@cityu.edu.hk*

Lynna L.H. Zhang  
*City University of Hong Kong, lzhang265-c@my.cityu.edu.hk*

Oscar Z.H. Cui  
*City University of Hong Kong, zhcui3-c@my.cityu.edu.hk*

Sherry H.Z. Li  
*City University of Hong Kong, hongzhuli2-c@my.cityu.edu.hk*

Follow this and additional works at: [http://aisel.aisnet.org/pacis2014](http://aisel.aisnet.org/pacis2014)

**Recommended Citation**

[http://aisel.aisnet.org/pacis2014/360](http://aisel.aisnet.org/pacis2014/360)

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2014 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
DATA DRIVEN DETECTION STRATEGY ENGINE FOR BETTER INTRUSION DETECTION ON CLOUD COMPUTING

Daniel W.K. Tse, Department of Information Systems, City University of Hong Kong, Hong Kong, iswktse@cityu.edu.hk
Lynna L.H. Zhang, Department of Information Systems, City University of Hong Kong, Hong Kong, lzhang265-c@my.cityu.edu.hk
Oscar Z.H. Cui, Department of Information Systems, City University of Hong Kong, Hong Kong, zhcui3-c@my.cityu.edu.hk
Sherry H.Z. Li, Department of Information Systems, City University of Hong Kong, Hong Kong, hongzhuli2-c@my.cityu.edu.hk

Abstract

In this paper, we attempt to base on CIDS framework and initiate a Data Driven Detection Strategy Engine (3DSE), a new thinking on identifying suspected users by adopting Decision Tree and Logistic Regression techniques to mine the usage patterns (from audit log and alert log) of different cloud member. Moreover, according to the analytical mining results, we also propose a danger-coefficient ranking model, which allows system to adopt different security strategies to monitoring users of different security levels. Deploying this engine, cloud system can be automatically trained up and become more efficient and effective on intrusion detection.

Keywords: intrusion detection; cloud computing; decision tree; logistic regression; coefficient ranking
1 INTRODUCTION

Cloud computing provides business a flexible and powerful platform to get timely computing capabilities without bearing the high cost and complex infrastructures. Many organizations quickly respond to this advanced technology to gain the benefits of elastic IT solutions and dramatic cost reduction. With successful and rapid spread, cloud servers worldwide accumulate lots of data and handle plenty of business activities every second.

However, although it brings much convenience to organizations, a serious concern on security raises as well. From a survey conducted by IDC (2008), the number one concern about cloud services is security. Another survey conducted by Deloitte Consulting (2011) continued to give a similar outcome that insufficient data security/availability is the top reason against the use of cloud computing solutions. Hackers probably make use of the capability much more readily when invading into other virtual machines in the trusted cloud network. It results in a severe loss on data, privacy and business secrets which must be a disaster for those individual users. In order to raise more public attention on this new security threat from abuse of cloud computing, Cloud Security Alliance (CSA) (2010) reported this as the top #1 threats of cloud computing security. It claims that the chief reason for nefarious use of cloud is that the cloud service providers apply weak detection strategies and low-level of security control. On the other hand, Jaeger et al. (2008) indicated that, as cloud computing has created a new environment varied from conventional internet world, by analogy to information policy, we should have corresponding cloud policy to specify the activities over cloud platform. Users need to obey the basic restrictions of access and usage in the cloud environment.

From a technical perspective, as Kholidy & Baiardi (2012) points out, the conventional methods, i.e. Intrusion detection system (IDS) (Scarfone & Mell 2007) and firewall are no longer efficient in cloud environment. The HIDS (Host-based IDS) and NIDS (Network-based IDS), which are commonly used in internet framework, have several drawbacks if they are applied in cloud environment. Therefore, Kholidy & Baiardi (2012) have introduced an innovative cloud IDS (CIDS) framework that can leverage the detection effectiveness and efficiency comparing to the conventional one. However, due to the flexibility and multi-users accessibility in cloud environment, applying traditional IDS rule-based detection may lead to a relatively high failure rate of reporting. For example, different legitimate cloud members may have various access times, which should be alerted by conventional rule-based IDS.

2 LITERATURE REVIEW

2.1 Security in Cloud Computing

Cloud computing consumers use the services provided by three parties. In the process of the cloud computing, the most important and necessary parts are the front and back end (Dhage & Meshram 2012). The front end represents the consumers’ side, where they can use the cloud resources through an interface which may be a computer or a web site. The back end represents the providers’ side, where contains all diversified consumers’ data and other information. The consumers can utilize what they want without any actual infrastructures. Therefore, it appears that the roles of providers and users in cloud computing are very important. How the providers distinguish the potential risky users and how to handle with the risks are central to the security assurance in cloud environment. To clarify the cloud security issues, we review the following factors that have a big influence and play a main part in cloud security.

- **Access**
  
  Only the authorized users can use the data in the cloud computing process. Even though a user is authorized, they may have a different level of the priority to access the data. The low-priority users sometimes may attach some special or sensitive data which probably brings potential risks to the cloud that are harmful to the whole system. Thus, access is the critical and basic aspect to defend the first level of risks.

- **Availability and Download**
Availability is that users can be satisfied by what they want no matter when they want it. It stands in the users’ side. Users may not get what they want if they suddenly have a huge download traffic which is beyond the capacity of cloud computing. It usually results in a server system breakdown in cloud system and threatens the security and stability.

- Data Security and Location

Data security is definitely important for the users and providers. For users, the data should be protected and exclusive, only for their internal use. If data is used in an undesired way, it may lead to a lot of vulnerabilities which may result in serious data leakage. For providers, data security is concerning their credibility and profit. In addition, cloud computing is famous for its liquidity, which means that data does not locate in a fixed place or a settled infrastructure. It should raise much concern by users on where the data is located.

2.2 Intrusion Detection System (IDS)

2.2.1 Introduction on IDS

IDS (Intrusion Detection Systems) which is also known as IDPS (Intrusion Detection Prevention Systems) is a series of procedure or software installed inside the organization system or network to monitor malicious and abnormal activities occurred in the system or network. The major functions of IDS are to monitor and identify suspicious activities, to log the activities information, to block the abnormal activities and to report them to corresponding system administrator. (Depren, et al. 2005) Normally, the IDS are comprised of four components: event generators, event analyzers, response units and event databases. Event generators serve to capture visit information from the whole computing environment or system and to publish them to other components. Event analyzers aim to make analysis based on the information from generators and produce analysis results. Response units will make corresponding reaction according to the analysis such as cutting off the connection, changing file attributes or just giving out an alert. Event databases are used to store various data of the middle process or the final outcome, which serve to direct the execution for event analysis and response. Since not all network traffic pass through the firewall, and only intrusions outside from the network or system are likely protected by the firewall, hence IDS is created beyond the prevention capability of firewall to detect those suspicious and unauthorized attacks originated from within the network or system.

2.2.2 Advantages and Disadvantages in Three Types of Conventional IDS

There are basically three types of traditional IDS according to the detection data source, including HIDS (Host-based IDS), NIDS (Network-based IDS) and distributed IDS (DIDS) (Chandola, et al 2009). HIDS monitors specific host machines, NIDS identifies intrusions on key network points and DIDS operates both on host as well as network. HIDS is installed on a particular computer to monitor activities which only happen on that system. HIDS excels at insensitive to large network traffic and capable of running in an encrypted environment. Because of its centralization, it increases the likelihood of being the target of attack by hackers and vulnerable to attacks against the host system. Besides, HIDS might cause performance issue over the host system as well as taking up more space for relevant audit trail and logs.

NIDS resides on computer connected to the segment of the network of organization to oversee activities occurred on the network. It could be easily deployed in the network without much disruption and affection over the existing network running operations. However, it does not work for encrypted network packet and susceptible to large network volume. DIDS adopts the measure of distributed detection but centralized control which consists of multiple IDS over the large network. They are able to communicate with each other which can be regarded as co-operative agents distributed throughout the network. It excels at the ability to detect attack across an entire cooperative network and this improves the efficiency of the detection. However, because of its broad data protection range, the performance of the network might be affected to some extent.

2.2.3 IDS Techniques
There are basically two types of intrusion detection techniques which are anomaly based and signature/misuse based intrusion detection. (Vieira, et al 2010) Anomaly based intrusion detection is to detect uncertain behaviors deviated from normal and accepted behaviors which are predefined in advance as the standard. If those uncertain behaviors are far more deviated from standard and normal ones, they will be regarded as intrusion. Anomalies also known as outliers, exceptions or peculiarities are patterns in data that do not conform to a well-defined notion of normal behavior of a system (Karthikeyan & Indra 2010). As there is no need to make a definition on each type of intrusion of uncertain and abnormal behaviors, hence it is effective to detect any possible unknown intrusion. Therefore, the false report rate is low whereas the mistaken report rate is high.

In contrast, misuse based intrusion detection is to detect the matching degree of a behavior with those abnormal behaviors which are yet pre-defined and known in advance. Misuse detection also referred to as signature-based detection because alarms are generated based on specific attack signatures. (Vieira, et al 2010) If each unacceptable behavior could be defined, then any system activity or behavior matched will be considered as intrusion. This type of detection is able to accurately pinpoint the details of the intrusion but is not effective for detecting those unknown intrusions. In addition, it needs to keep a persistent upgrade for known intrusion behaviors. Therefore, the missing report rate is high whereas the false report rate is low.

2.2.4 IDS in Cloud Environment

Since cloud computing is still not mature, lots of vulnerabilities make it a fragile environment leaving backdoors for attackers to break in and mess up the whole system. Although some intrusion detection systems (IDS) exist for internal malicious activities detection, most of these IDS are for non-cloud system. IDS for cloud environment are in demand. As introduced by Kholidy & Baiardi (2012), IDS that can be applied to cloud system should have following six characteristics in order to adapt the main features of cloud. They are 1) distributed and scalable, 2) separate the host machine and IDS, 3) having a flexible structure to adapt different cloud structure, 4) having a strong behavior detection system which combines behavior and knowledge based techniques, 5) taking different service models and user requirements into account, 6) having alarm system to alert the suspect behaviors.

2.2.5 Data Mining Adopted in IDS

Caulkins, et al. (2005) also proposed an application of classification tree in its intrusion detection. As they mentioned, research on IDS has been mostly concentrating on misuse detection. Thus their effort was emphasized more on anomaly detection. In their work, they created a DT using Chi-Square splitting criteria and trained a model from the MIT Lincoln Lab data sets (1999). The modeling aims to classify the network session into two categories - probable attacks and non-probable attacks. It also constructs an additional filtering of signature detection.

However, their discussion was only limited to TCP packets only. In order to test the model, they collected hours of traffic from the UCF network and ran the data. The result shows the model did label approximately eighty-five percent of the probable attacks properly. Marinova-Boncheva (2007) described an overview of Intrusion Detection and data mining method in detection system. Distinguished from data fusion, data mining is used to detect unknown or hidden patterns, and a typical technology is classification tree, or decision tree that is used. The author also uses it to exemplify the process by testing a sample of Windows NT attack data set. However, other than misuse detection, it is pointed out that the existing anomaly detection remains a deficiency that it reports high false alarm rate as recognizing previously unseen but yet legitimate system behaviors.

2.2.6 Decision Tree

Dhage & Meshram (2012), Kholidy & Baiardi (2012) both proposed a respectively IDS architecture which is not limited to a central IDS in the entire network. The former is relatively more conceptual by introducing a mini-IDS instance mechanism attached to each cloud user and a cardinality system of node controllers, which is of more flexibility and releasing the load of one single IDS in a traditional system. However, this framework is demonstrated as a relatively theoretical model with an insufficiency of rule control mechanism and other supportive components. Chirag, et al. (2012) proposed a meliorated feature function in its integrated NIDS in cloud system. It extends the mode of
2.2.7 Logistic Regression

Wang (2005) proposed a multinomial logistic regression modeling for anomaly detection. Most of the researches use the decision tree algorithm to output a binary detection, “abnormal” or “normal”. However, there are actually multi-type attacks. Therefore, this modeling aims to provide detail classification of the attacks type. In the logistic regression process, the attack types are converted as dummy variables and categorized by independent and dependent ones. This algorithm is novel and the result is respectively positive, but the regression algorithm needs rather accurate and suitable value setting to the variables, and is very sensitive to the inappropriate numerical value.

2.3 User Behavior Pattern in Cloud

In order to successfully apply Intrusion detection system (IDS) to the cloud, the most important thing that we have to do is to acquaint with the user behavior patterns. As cloud computing is still a developing technology, its novel fancy features, i.e. shared underlying infrastructure, distributed nature, allocation of resources on demand, make cloud system a desirable environment for hackers to take malicious actions. Some hackers have come up with new ways to attack cloud system, so that the previous IDS are not secure enough to protect cloud computing system. In order to equip cloud system with the ability to detect possible attacks, we have reviewed some researches on identification to the user behavior patterns. Vieira, et al. (2010) point out user behaviors can be detected by two methods, behavior-based method and knowledge-based method. In other words, user behaviors can be interpreted by two user patterns, i.e. Knowledge-based User Pattern and Behavior-Based User Pattern.

2.3.1 Knowledge-based User Pattern (KBUP)

In general, knowledge-based user pattern represents malicious use behaviors on computing system that have been recorded in public database as an attack pattern. Comparing the current user actions to KBUP, system can identify that the current user is an attacker, if the user’s action is very similar to one of the attack pattern in KBUP. For cloud, Singh & Shrivastava (2012) offer a breakdown structure of cloud system in order to identify cloud user pattern by comparing to the existing non-cloud system. They divide cloud structure into three parts, namely, service users, service instances (or just services), the cloud provider. They split the above three parts into two sides, resulted in six possible combinations listed as following.

(a) Service-to-User  (b) User-to-Service  (c) Cloud-to-Service
(d) Service-to-Cloud  (e) Cloud-to-User  (f) User-to-Cloud

After the whole structure is broken into detailed types, some of the cloud structures can be compared to the structure of existing non-cloud system, which indicates that both are similar in many aspects, such as infrastructure, user’s behavior pattern, even the way of being attacked. For instance, 1) a service instance towards a user is very similar to server-to-client interface, which can be attacked by privilege escalation, SQL injection or buffer overflow attack, 2) a user to service model has many similarities with HTML-based service. Therefore, this structure is likely to be attacked by SSL certificate spoofing and phishing.

2.3.2 Behavior-Based User Pattern (BBUP)

Besides the known user pattern illustrated above, unknown user behaviors also compromise cloud system. Unknown behaviors include intentional and unintended behaviors.

Some scholars propose taxonomy of end user behaviors (Stanton, et al. 2005)

- Intentional destruction
- Detrimental misuse
- Dangerous tinkering
- Naive mistakes
Any one of these behaviors would probably bring destructive damages to the cloud system. Here, any action that may damage system is considered as attack. To better protect our system, studies on these known behaviors are required.

Generally, users have certain behavior when they use cloud system. Behavior-based user pattern (BBUP) depicts the normal use pattern which usually contains a sequence of user’s operation habits. For instance (Tian, et al 2010), in using a cloud system at any time, a user may conduct several behaviors, i.e. open applications, download files, or input system commands. These normal behavior patterns can be described by the average number of applications being opened, average resources downloaded and certain commands inputted. By comparing BBUP to the current user behaviors, if result indicates a huge difference, the system can treat the current user as an attacker. Focusing on classifying user behavioral patterns and deviations from such patterns, we are able to identify a large range of unknown attack behaviors. All these studies of behaviors are for the later use in behavior detection. The more information we know how hackers attack, the more possibilities our system can detect attacks.

3 PROPOSED METHOD

In this session, we will introduce our proposed model, Data Driven Detection Strategy Engine (3DSE), in details based on the researches done in the previous sections. As shown in Figure 1, firstly data from IDS and user-basis knowledge dataset is captured and normalized by 3DSE. After formatted, such user pattern data sets are input into the core of 3DSE, where mainly adopts Decision Tree and Logistic Regression techniques to identify suspected user behaviors and generate corresponding dangerous-coefficient score of different cloud users. According to the score, system categorizes users by applying a Danger-Coefficient Ranking Model so as to determine the users’ risk level. In the next step, regarding the five levels of ranking, system suggests different security strategies (i.e. detection, protect or prevention) towards the relevant users. Finally, after finishing a working cycle, 3DSE will get back to the data capturing for next cycle to realize continuous improvement. The following sections will introduce respective components inside the 3DSE.

Figure 1. Proposed Model – Data Driven Detection Strategies Engine

3.1 CIDS Framework Environment

Initially, before getting in detail of four processes in our proposed model, we need to briefly recap the structure of CIDS, which is the basic environment for 3DSE. Figure 2 depicts the relationship between CIDS framework and 3DSE.
As distribution is one of the inherent features of cloud, in order to adapt this feature, every cloud node has intrusion detection system inside. Within the intrusion detection system, Virtual Machine (VM) Monitor provides Auditor System with the data collected from Logs and Audit collectors. Auditor System stores and prepares the necessary behavior data for both Correlator and Detectors. Collectors and detectors work together to detect the malicious behaviors in cloud system based on behavior and knowledge intrusion detection methods and then send the processed result to alert system. After alert system receives the result, it will send warning messages to administrators. 3DSE comes between VM monitor and Alert system as an improvement in Collectors and Detectors component.

3.2 Data Capturing and Normalization

With an understanding of the CIDS framework environment, it comes to the first step of the proposed 3DSE - Data Capturing and Normalization. It aims to classify various kinds of attacking patterns into different categories as the input parameters of subsequent decision tree training process. Hackers will use different hacking techniques for different kinds of attack and purpose, and different attack might have different user behaviors. Furthermore, hackers are likely to divide one single attack event into different sub attacks as different stage, and one sub attack of certain stage might be the prior step of the next one. In this way, the consequent damage of the single attack event might not emerge until its final step finished. In this case, it will be more efficient to reduce the risk probability if such attack is detected in the early time slot. Therefore, based on different hackers’ attacking behavior and strategy, this data capture and normalization model should be designed to collaborate together with IDS to enhance the capability of the cloud computing system.

3.2.1 Parameter Definition Pool (PDP)

In order to set up the parameter definition pool, the parameters should be firstly present.

Below list the possible behavior related parameters:

- IP Address (IP) - It is the IP address of the user who is visiting the system.
- Port Number (PN) - It is the port number by which the user is visiting the system.
- Visiting Frequency (VF) - It describes the visiting frequency of certain behavior on user level. If one abnormal behavior of certain user happens frequently for example every two hours compared with his usual behavior, and then it is likely to be a potential attack and should be alerted by the detection system.
• User Credibility (UC) - The UC is measured based on the user’s history behavior and action records to see whether the user is a threat to the system or not and the severity of the threat.
• Visiting Time Slot (VTS) - VTS indicates the time when the user visits the system on behavior based. Suppose one user is used to visit the system in the morning at 8:00 am to check his personal emails every time. If one day he changes the time to at night 11:00pm, then this action is possible to a potential attack and needs to be alerted to system. The visiting time slot could be classified to different ranges and system could then calculate which range the current visiting time fall into.
• Visit Count (VC) - VC indicates the count of visiting on daily basis for certain user. If it is calculated as quite a different value at any time of the user compared with the usual value, then it is likely to be a potential attack.
• Visiting Time (VT) - VT indicates the duration of the users’ single action on behavior based. For example, normally it just takes one minute on average for a user to login to the system. If one time of login action takes five minutes, then it is most likely this login is an attack.

The parameter definition pool is suffused based on IDS rule set predefined (Gul & Hussain 2011). For the IDS technique of anomaly detection, there will be an IDS rule set which is comprised of predefined rules for matching user behaviors to detect malicious or abnormal actions to the system. All of the factors above need to be normalized as the input parameters for later decision tree training.

3.2.2 Knowledge Dataset on User Basis (KDU)

Generally IDS will include the event database which stores various kinds of data for later analysis or data with other purposes as well as logs. For anomaly detection, there is centric knowledge dataset as the benchmark to detect the abnormal behaviors by matching with predefined rules. Apart from that, the KDU mainly aims to store and maintain the information related to the behavior pattern benchmark on user basis. After each training on decision tree, the dataset will be updated and the benchmark might be updated.

3.2.3 Data Normalization Calculator (DNC)

When the event generators of IDS monitor and detect the attacks, the data normalization calculator will be activated to make the calculation on each parameter for this event according to certain format. The calculation is conducted based on the existing data stored in event database of IDS and data in KDU, which will be introduced later. In addition, DNC will handle the normalization of each parameter as well.

3.3 Data Mining Techniques

After the strategy definition structure is erected, we also propose a methodology that develops a robust Intrusion Detection System by using the data mining technologies and anomaly detection. In cloud Intrusion detection, end user behaviors include intentional and unintentional behaviors, and in most of the researches, anomaly detection is used to discriminate the pattern as an attack or not. Classification-tree techniques are used to accurately predict probable attack sessions. In this research, decision-tree method is also used as a key proportion in prediction in central control of attack behavior detection. In order to make the whole process more robust and reduce the false alarm probabilities, we propose a pre assessment system which reduces the possibility that the user’s naïve or unintentional behavior is mistakenly understood.

These extra consolidated methodologies are absorbed to establish a more trusty worth modeling. Unlike the existing dynamic data mining model that only distinguishes between normal and abnormal or suspect attack, our system pre-define the classification of user behavior trust group to distribute a different weighted value of treatment to user accounts. Based on this seriousness-degree class, we then distribute different strategies to cope with the specific users.

Our user behavior trust model consists of two parts. First is trust evidence pool, which is obtained by the existing invasion detection system and network flow detection tool. The EIDS, such as Tcpdump, contains the trust evidence includes the times of access, the times operation failure, transfer delay etc.
The network flow contains HTTP, TCP, and UDP etc. Second is the user behavior evaluation. The data can be obtained from all kinds of log and audit trails, the evaluation mechanism is defined in the former section of weighting model.

The trust evidence pool has existing knowledge set and the training data is based on this knowledge based dataset. During this process, a regression method is used to extract effective evidence attributes. Classification tree is used to output the anomaly judgment, and rules set are key influence to the DT’s performance, thus we propose logistic regression as a selection of effective rules set.

3.3.1 Candidate Rules Preparation

There are several steps to first generate the rule set.

- Generate the candidate rules from the random data
- Evaluate and then select a minimal rule set
- Train the selected rules on the training data set
- Prune the rules that are not correctly predicted

However, rules are used as input evaluation of features in decision tree, a more reliable and comprehensive rules set plays key influence on the efficiency and accuracy of the prediction output. Therefore, a trusty algorithm is needed to generate the original candidate rules. In this research, we propose a respectively greater number of original but potentially significant candidates and use regression to select the representatives of the dominant rules. Afterwards, we implement the rules training process to promise valid rules set. We use this amendment to reduce omissions of effective candidate rules, with an application of big data and data mining.

The input parameters should be normalized and the dependent variables should be converted as dummy variables if needed. Wang’s (2005) method is directly used to detect the categorized attack type but is rather sensitive and we deploy it as a method of candidate rules selection to perform better accuracy with an extra application of decision tree.

We assume that the dependent variable is categorized as user behavior hierarchy. Using the training data to run out a set of responding coefficients \( w_i \), and a threshold is set when \( w_i > \tau \) means an effective feature rule. A sequence of feature rules is obtained and the decision tree’s nodes are up-bottom with the feature rules of coefficient from highest to lowest, the training process is a splitting threshold training at each node. This modeling is more efficient for dynamic detection.

3.3.2 Decision Tree

We create this decision tree using the Chi-Square splitting as the feature and a corresponding threshold at each node.

- Training rules set, \( F_T = \{ f_{ij} = 1,2, \ldots N \} \), \( N \) is the number of input features
- Training data set \( D_T \), a package of random behavior recording data according to rules set.
- Each node’s attribution \( g = \{(f,\tau), f = f_{ij} = 1,2, \ldots N\} \)

We set an original threshold at each splitting branch and calculate each node’s probability function using chi-square method which is multivariate normal distribution.

Afterwards, the ending output is a package of nodes with a prototype model of

\[
P = \{ (f_{m,i}, h_i), i = 1,2, \ldots N, m = 1,2, \ldots M \}, M \text{ represents the model number.}
\]

3.3.3 Ranking Score

The previous section describes the training procedure for the classifier, In this section, we detail the strategy class ranking method to deploy. Through this method, we only obtain the result histogram distribution to decide the suspect or un-suspect sessions. However, as we mention in the previous section that a score ranking is deployed to the following alarm procedure.

In this research, we calculate the deviation of the total feature threshold as the judgment to distribute the score to determine the final ranking system. If we define a binary output from the Decision Tree
algorithm, then the outcome is only judged by a binary category, “abnormal” or “non-abnormal”. However, in order to define the user behavior into a more categorized discrepancy, we remain the threshold at each node on the effective path. Then we assume that the user who matches with the higher probability of a total features set is more likely to initiate a serious attack. Thus, we compute the deviation by calculating the coefficient,

$$\sigma_{r, m}^2 = \sum_{i}^{N} (x_i - \tau_i, m)^2$$

Therefore, we augment the model

$$P = \left\{ \left( f_{n2m}, i \right), i = 1, 2, ..., N, m = 1, 2, ..., M \right\}$$

with the coefficient matrix. We then obtain each feature’s weighted score by averaging each model’s.

Afterwards, we determine the final strategy ranking system by classifying the obtained weighted values, and we define the ranking level by the matching of features with higher weights. The ranking package will be discussed in more detail in next section. This model is also suitable for attack detection (packet analysis) and the final outcome can be the classification of attacks.

### 3.4 Dangerous-Coefficient Ranking Model

According to the dangerous-coefficient score of different cloud users, the following step illustrates the Dangerous Coefficient Ranking Model (DCRM) on defining different categories of users from high risk to low risk under the unified score standard. It serves as the basis to provide reference to system when conducting various security strategies.

The users can be divided into five levels from the low risk to high risk that the cloud system confronts. Corresponding measures will be conducted according to the user behavior trust levels.

Different risk-level user categories are illustrated as below:

- **Normal User (NU)** -- Cloud users who obey the system policy and procedures are classified as the normal users, the action of whom will not lead to any risks to the system which may result in data leakage, paralysis and so on.
- **Noticeable User (NOU)** -- The cloud users’ action is classified into this category may make system vulnerable and sensitive but producing little risks to the system on both hardware and software aspects. The actions might be considered abnormal compared with the usual ones, such as a large amount of download, numerous authorization defaults.
- **Suspect User (SU)** -- The user will take some attacks to the system. The attacks may occur in a relatively high possibility no matter which degree the attacks are in. In other words, if they take any risks to the system in high probability, they belong to the Suspect User. The CIDS must be sensitive to these attacks originally so that they cannot intrude the system easily. In case they make it successful, the remedial measures are playing an important role.
- **Significant Suspect User (SSU)** -- Significant Suspect User is a kind of cloud user whose vulnerabilities are apparent and enormous to the cloud system within the cloud providers and other cloud users in high possibility. Under this case, the IDS may detect the attacks frequently. If these attacks intrude the system successfully, they will cause the system disable and shut down. As a result, the cloud provider and the clients will have a huge economic loss even the provider losses reputation.
- **Proved Hacker (HP)** -- This kind of user is named and proved hacker because their purposes are apparent to stealing the other users’ data or the doing some attacks which will lead to a loss. This kind of user is in the highest level about vulnerabilities. The IDS should take immediate measures to defend these attacks.
### Division of users, Corresponding scores, Degree of risks

<table>
<thead>
<tr>
<th>Division of users</th>
<th>Corresponding scores</th>
<th>Degree of risks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal User (NU)</td>
<td>0-10</td>
<td>No influence</td>
</tr>
<tr>
<td>Noticeable User (NOU)</td>
<td>10-30</td>
<td>Little risky</td>
</tr>
<tr>
<td>Suspect User (SU)</td>
<td>30-50</td>
<td>Lowly risky</td>
</tr>
<tr>
<td>Significant Suspect User (SSU)</td>
<td>50-70</td>
<td>Highly risky</td>
</tr>
<tr>
<td>Proved Hacker (HP)</td>
<td>70-100</td>
<td>Certainly risky</td>
</tr>
</tbody>
</table>

*Table 1. Summary of different risk-level users and its corresponding score.*

#### 3.5 Strategy-based Users Behaviors Monitoring

In order to protect the cloud computing system efficiently, necessary and effective measures should be taken from the aspects of both cloud providers and cloud users aspects. The common strategies are the encryption mechanism, security authentication mechanism and access control policy (Xin 2010). Current actual strategies are the transformation of the three big directions.

- **Normal users (NU)** -- Their actions will not influence the stability of the cloud system, so the general security policies and guidelines is enough for the system if the users and providers access to the data properly.

- **Noticeable User (NOU)** -- This kind of users potentially incurs some risks to some extent to the system. Under this suspected condition, system needs to take more measures to validate the users’ suspicion possibility. One feasible measure is that system can make full use of the proposed engine by extending the range of data captured by the component of data capturing and normalization in order to analyze the information of the users’ behavior as much as possible.

- **Suspect User (SU)** -- Users labelled in this level usually performance abnormally comparatively. Within this level, the system should think them as dangerous symbols that can make the system work ineffectively. It is urgent for both the detection and remedial work to do enough protection measures to defend potential attacks. Corresponding Measure to be adopted is authority restrictions, i.e. firewall, limited file access permission, network visit restriction. If the actions are proved to be innocent, alarm is released.

- **Significant Suspect User (SSU)** -- As this kind of users has significant high level of risks similar to hackers, system level restriction should be conducted to protect the system from immediate damage, i.e. terminal connection immediately and isolate users’ account from others. Furthermore, system should remind administrator to analyze users’ account in details. If any evidences found to prove as hacker, the next level of strategies should be employed.

- **Proved Hacker (HP)** -- Proved Hacker is the most threatened user level in our ranking system. If a user is considered as Proved Hacker, the system will back up all information related to this user in case of necessary future recovery, then delete these information in the system in order to reject any possible request and entry of this highly dangerous user.

#### 4 DISCUSSION AND LIMITATIONS

It is very important for us to remind readers that 3DSE, which is the main content discussed in this research, is just a conceptual idea used to improve Cloud Intrusion Detection System. There is no practical testing based on real data being conducted. Let us back to the mainstream. As cloud is a complex and huge computing structure, there will be numerous invisible security threats inside. On one side, our proposed method improves intrusion detection system in cloud and provides the system administrator a hierarchical list that helps the administrators to assign authority to ranked users. On the other side, cloud is too huge to solve all problems at once, so our model still has many rooms to be improved in future. There are several problems needed to be taken into account.

First remained problem is that, when a new user enters system, the system becomes fragile. Because new users do not have any record in the system, if they are impersonated by hacker, the system can only detect the abnormal behaviors by analyzing their behaviors based on knowledge-based method.
rather than using both knowledge-and-behavior-based methods. Therefore, once a new comer arrives, the system would be probably in danger for a while, which also can cause unrecoverable damage. In terms of security in cloud system, this is a bad news, so this problem has to be solved in our future study.

Secondly, besides the problem is caused by new users, if we want our ranking model to be more accurate, the factors related to user behavior need to be extracted as complete as possible and we have to address the possible performance issue caused by feeding and consumption amongst the big volume of KDU. In addition, the trust model is based on large behavior accesses. The accuracy and authenticity of learning process also depends on the scale of training data. To support this relatively slow rise model, subsequent rules are required to enrich and consolidate the knowledge set.

Last but not the least, as we propose a conceptual model in this article, the strategies that are designed for telling system how to manage users with different ranking level in our model are just a primary concept which still need to be improved in order to fully take advantages of the results provided by our proposed model.

5 CONCLUSION

Cloud Computing is undoubtedly the most popular technique in IT industry while meanwhile with plenty of controversy and concerns due to its immaturity and potential security issues. In this research, we propose an intelligent engine (3DSE) by incorporating the knowledge and technique related to IDS and data mining based on one CIDS framework. This proposed engine aims to improve the accuracy of IDS reporting and to enhance the efficiency and effectiveness of the detection on those malicious intrusion and attack. The main idea of this engine is to compute a relative precise and reliable ranking score for cloud users in terms of their behaviors in order the cloud system to take corresponding strategic action. This engine comprises of three parts - data capture and normalization procedure, data mining process and dangerous-coefficient ranking model. In order to obtain the ranking score, the engine will establish and maintain the behavior knowledge benchmark and relevant rules on user level. The data capture and normalization procedure will collaborate along with the data mining process by adopting advanced techniques including decision tree training and logistic regression analysis. After mining process, the user level behavior benchmark dataset will be suffused and maintained for future detection cycle. Besides, the mining process will also generate the concrete ranking score as the input variables for the dangerous-coefficient ranking model, and then the ranking model will adopt corresponding strategy to handle various ranking outcome.

In conclusion, although there are still a lot of works to be done to defend cloud system, we believe that we have walked through the most difficult first step and we are heading to the right direction. In near future, we will come up with an effective and efficient model in protecting cloud environment.
6 REFERENCES


