Applied Mathematics & Information Sciences
An International Journal

New Network Access Control Method Using Intelligence Agent Technology

Hu Ruo, Guo Jian hua

Guangdong Polytechnic Normal University, Guangzhou, 510665 P. R. China

Received: 22 Aug. 2012, Revised: 7 Oct. 2012, Accepted: 19 Nov. 2012
Published online: 1 Feb. 2013

Abstract: Today, new study has been made to the Abnormal Access Discovery (AAD). While label-based AAD have proven effective in discovering known abnormal access, label-based AAD can be able to automatically detect previously unrecorded risks. Traditional AAD cannot meet evolving network workflows in real time. To resolve this limitation, intelligence agent techniques can be used to create a new type of AAD that can be able to dynamically model a flow of network access. In this paper, we present two methods for abnormal network access. The two methods include diagram-based probe method and the classifying-based method. Overall, the diagram-based probe method achieved slightly superior results, but required more parameters than the classifying-based method. As a result of its fewer parameter requirements, the classifying method can be more easily generalized to different types of network traffic flows.

Keywords: Access control, abnormal access discovery, artificial intelligence and logic (ALC)

1 Introduction

Since the 1990's, internet usage has become an integral part of our daily lives. As a result, computer networks have experienced an increased number of sophisticated malware attacks. Whereas attackers previously attempted to gain access to restricted resources to demonstrate their skill, a new wave of internet-based attacks has shifted the focus primarily towards criminal motives. Due to the availability of software tools designed to exploit vulnerabilities, attackers can create viruses with greater structural complexity and damaging capability using less sophisticated skills. The security challenges resulting from an increasing number of devices connected to the inter-net has prompted a significant amount of research devoted to network security.

1.1 Abnormal Access Discovery

One notable topic of network security research is the development of Abnormal Access Discovery (AAD), which attempt to detect threats to a network or host through signature-based or anomaly-based methods. To detect intrusions, signature-based AAD generate "signatures" based on characteristics of previous known attacks. This allows the systems to focus on detecting attacks regardless of ordinary network traffic. Signature-based detection is the most common form of access discovery because it is simple to implement once a set of signatures has been created. Although this approach is effective in finding known threats to a network, it is unable to identify new threats until a new signature is made. To generate an accurate signature, a human expert is generally needed because this cannot easily be done automatically. Since the detection of new threats in a signature-based system is impossible without the aid of a new signature, an alternative method has been proposed.

In contrast to the signature-based approach, anomaly-based AAD adaptively detect new attacks by first generating a "normal" pattern of network traffic. These systems then find anomalies by comparing incoming data flows with the "normal" model. Anything that is considered statistically deviant is classified as abnormal. This allows for the systems to automatically detect new attacks though risking possible misclassification of normal behavior (false positive).In addition to the potential for false positives, anomaly-based systems also fall prey to "mimicry attacks", which attempt to evade the AAD by imitating

* Corresponding author e-mail: hu68@163.com

normal network traffic. One such attack is known as a Polymorphic Blending Attack (PBA), in which the attacker uses byte padding and substitution to avoid detection [1].

2 Materials and Methods

Two publicly available datasets were used to evaluate the Access Control methods proposed in this study.

The DARPA'99 dataset was used to provide a sampling of normal network traffic. This dataset simulates network communication from a fictitious United States Air Force base, and provides both attack-free and attack-containing network traces.

Following the same procedure used to process the DARPA'99 week one data, the numeric character values contained in all HTTP data flow weights from each of the three attack datasets with lengths of at least 1350 characters were extracted using New Star.

The weight information extracted from the DARPA and attack datasets was used to create training and testing datasets for our Access Control systems. For each of the five days in the DARPA dataset, 25% of the day's data flows were extracted to be used for training, and the remaining 75% of the day's data were set aside to be used for testing. To simulate the network traffic in real time, abnormal data flows were then sporadically inserted into both the training and testing data after an initial interval consisting of only normal traffic (60 data flows for training data and 180 data flows for testing data).In this way, different datasets were created with each attack type for all five days of DARPA'99 week one. The total number of abnormal data flows inserted into both the training and testing data was no more than 10% of all normal data for the given day with weight length 1350 characters or more. In some cases, the number of abnormal data flows inserted into data was less than 10% because there was not enough attack data available for that day. For the training data, 25% of the abnormal data was mixed with 25% of the normal data for each day. Likewise, 75% of the abnormal data was mixed with 75% of the normal data selected from each day for testing. For each data flow, 258 1-gram and 65,435 2-grams were extracted to pro-duce separate representations of the training and testing datasets. The datasets were stored in the ARFF file format used the by the open source machine learning software WEKA.

The weight information extracted from the DARPA and attack datasets was used to create training and testing datasets for our Access Control systems. For each of the five days in the DARPA dataset, 25% of the day's data flows were extracted to be used for training, and the remaining 75% of the day's data were set aside to be used for testing. To simulate the network traffic in real time, abnormal data flows were then sporadically inserted into both the training and testing data after an initial interval consisting of only normal traffic (60 data flows for training data and 180 data flows for testing data). In this way, different datasets were created with each attack type for all five days of DARPA'99 week one.

2.1 Classifying-Based Access Control

Classifying methods are commonly used for Access Control, and are generally created for the batch environment. However, some batch classifying methods, such as DBSCAN, can be modified to process flow data.

2.2 DBSCAN

DBSCAN is a density-based classifying method developed for the batch setting. The method takes two user-defined parameters, epsilon (ε) and minimum points, and relies on the concepts of ε-neighborhood and core-objects. A ε-neighborhood is defined by DBSCAN as being a set of points that have a distance to another point less than the user-defined parameter ε. More specifically, given point p and dataset D, the ε-neighborhood of $p(N_\varepsilon(p)$ is equal to:

$$N_\varepsilon(p) = q \in D | dist(p,q) \leq \varepsilon, \tag{1}$$

where $dist(p,q)$ is the Euclidean distance between points p and q.

A core-object is defined as a set of points within an ε-neighborhood that contain more points than the mini-mum points parameter. If p is part of a core-object, DBSCAN will expand the cluster around p.

The basic structure of the method is as follows:

1) DBSCAN takes the ε and minimum point parameters and then chooses a point p that has not been visited.

2) DBSCAN calculates $N_\varepsilon(p)$. If the size of $N_\varepsilon(p)$ is greater than minimum points, DBSCAN expands a cluster around p. Otherwise, the point is considered noise.

3) DBSCAN iterates to a new unvisited point and repeats the process.

Although DBSCAN was originally developed for a batch environment, it has provided an inspiration for flow classifying methods.

2.3 Wei-Flow

Wei-Flow is a flow classifying method based on DBSCAN with a damped window model. It expands the concept of a ε-neighborhood in DBSCAN with a fading function to maintain up-to-date information about the workflow. The fading function is defined as:

$$f(t) = 2^{-\lambda t}, \tag{2}$$

Where λ represents the decay factor and t represents the time.

Wei-Flow also modifies the core-object concept of DBSCAN, creating a core-micro-cluster with three additional attributes: radius, center and weight. The radius must be less than or equal to ε, and the weight of a cluster must be greater than the user-defined parameter μ. The weight w, center c and radius r of a core-micro- cluster are more formally defined at time t, for a set of close points, $p_1, p_2, \ldots, p_n$ with time-stamps $T_1, T_2, \ldots, T_n$ as:

$$w = \sum_{i=1}^{n} f(t - T_i), \tag{3}$$

Where $dist(pi, c)$ is the Euclidean distance between the point pi and the center c.

Although the p-micro-cluster permits the model to be updated dynamically, it generally will not provide a representative view of a workflow as new points appear. To handle this concept drift, Wei-Flow also introduces the outlier-micro-cluster (or o-micro-cluster) and an outlier-buffer that temporarily stores o-micro-clusters and allows them to become p-micro-clusters. The operation of Wei-Flow is as follows:

$$c = \frac{\sum_{i=1}^{n} f(t - T_i) p_i}{w} \tag{4}$$

Initial Step: run DBSCAN on a set of initial points to generate starting p-micro-clusters.

Online Steps, when a new point p arrives in the flow:

1) The method attempts to me to merge p with the closest p-micro-cluster. If the radius of the potential micro-cluster is less than or equal to the value of ε, the point is merged.

2) If the point is not merged to a p-micro-cluster, it tries to merge p with an existing o-micro-cluster. If the radius is less than ε, it is merged with the o-micro-cluster. Then if the o-micro-cluster now has a weight large enough to become its own p-micro-cluster, it is removed from the outlier-buffer and added to the model as a p- micro-cluster.

3) If the point cannot be merged to an existing o-micro-cluster, it creates a new o-micro-cluster and gets placed in the outlier-buffer.

4) If the weight of a particular cluster is less than the lower weight limit, the o-micro-cluster can be removed from the outlier buffer.

2.4 Creation of the Detection Model

The Access Control model was created in two steps. The first step used the training data to find a range for the parameters in Wei-Flow-Detection such as ε and minimum points. Using 50 initial points, multiple Wei-Flow-Detection models for each day were created to find a range of optimal parameters that could be used in the testing step. We found that ε had a larger impact on the predictions than the minimum points. During the first step, different parameter ranges were identified based on day and abnormal data flow type.

The second step used the testing data to make a prediction model, which was evaluated with the sensitivity and false positive rates. A false positive is a normal data flow classified as abnormal. The parameters used in this step were 180 initial data flows, 10 minimum points and a range of values specific to each day and attack type determined from the first step.

2.5 Method Description

The chart-based detection method provides a simple method for classification of network traffic. The method summarized in Method 2, creates a Chart compassing a ”normal” model of the network data flows expected to be encountered. This chart is generated by counting the frequency of n-gram features Found within data flow weights. To begin classification of flow of data flows, the method first requires x initial normal data flows to construct the normal model chart.

$$r = \frac{\sum_{i=1}^{n} f(t - T_i) dist(p_i, c}{w}. \tag{5}$$

This chart contains frequency counts from all initial data flows for each possible n-gram attribute. Since we are attempting to model normal traffic, it is imperative that no abnormal data flows are included when this model is created or else the model will be contaminated and detection rates will decrease. To effectively reflect the evolutionary nature of network traffic, the same fading function with decay factor used in Wei-Flow is applied to the chart after each new data flow is processed. This helps to reduce the impact of outdated flow data. After the initial chart has been built, the method can begin to classify the subsequent data flows.

In order to classify an incoming network data flow, the method builds a chart from the newly arrived data flow’s weight. The chart generated from the new data flow is then compared with the normal model chart (to which the fading function has been applied as each new data flow comes in) by computing the Pearson correlation value between the two charts. If the computed Pearson correlation value is above a user-defined threshold t, the data flow is classified as normal; otherwise, the data flow is classified as abnormal. Otherwise the data flow is classified as abnormal.

3 Results and discussion

3.1 Density-based Detection Results

After tuning the Wei-Flow-Detection-based system on data flows using 2-gram features, we discovered a range of ε values for each day that could be used to evaluate the

model. For every day except for Tuesday, the false positive rate was kept between 0% and 10% so that an appropriate detection rate could be found. Tuesday, however, needed the false positive limit to be heavily relaxed in order to achieve a moderate sensitivity. When testing the detection system, the false positive and sensitivity rates for the highest, middle and lowest εvalues were generated for both1-gram and 2-gram feature representations. The results with highest sensitivity for each virus type were then averaged to find best overall performance.

In general, the Wei-Flow-Detection-based system was able to correctly detect most Shell-code attacks, achieving on average 92% sensitivity with a 15% false positive rate. Similarly, Generic HTTP attacks produced 79% average sensitivity and a 14% false positive rate. CLET attacks, however, had a similar false positive rate of 14%, but a substantially lower average sensitivity of 66%. This disparity was likely due to the polymorphic nature of CLET attacks, which are designed to mimic normal network traffic.

Thursday exhibited the highest detection rates (up to 99%) whilst keeping the false positive rates below 6%. Also, Thursday experienced both the lowest ε value sand the largest ε range to achieve its results.

The models utilize both 1-gram and 2-gram features-produced similar results. Using the 1-gram representation for the same values, the system experienced slightly better detection rates at the expense of higher false positive rates. Also, because the 1-gram representation has a much smaller feature space than 2-gram, the total run-time of 1-gram was significantly less.

3.2 Chart-Based Detection Results

The chart-based method was applied to abnormal access control in two steps: training and testing. In the training step, favorable parameters for the method were approximated by performing several experiments on the training data. Since the training data included 60 initial normal data flows, this value was used for x during the training step. Most critically, appropriate values of t were ascertained for the different attack types on each day, as this parameter has the greatest effect on the performance of the method. Suitable values were also obtained for all other parameters during the training phase. The optimum value of q was found to be 180, as this allowed the method to maintain a fairly accurate model of normal traffic while minimizing the time needed to for this model to be rebuilt. Also, 32 were generally used for w, with r valued at 11 and h at 0.3. These parameters effectively limited the frequency of rebuilding the normal chart model while still allowing the method to handle concept drift in the data. A λ value of 0.02 was found to work sufficiently well, as this decay factor helped to better maintain an up-to-date normal model for normal traffic.

Because Wei-Flow operates in a flow environment, the core-micro-clusters need to change dynamically as time passes. To facilitate this, a potential core-micro-clusters or p-micro-cluster is introduced. P-micro-clusters are similar to core-micro-clusters, except they differ in that the center and radius values are based on the weighted sum and squared sum of the points ($\overline{CF^1}$ and $\overline{CF^2}$). Also, the weight must be greater than or equal to $\beta\mu$ where β defines the threshold between p-micro-clusters and outliers (described in the next paragraph) such that $0 < \beta \leq 1$. $\overline{CF^1}$ and $\overline{CF^2}$ are calculated using the formulas:

$$\overline{CF^1} = \sum_{i=1}^{n} f(t-T_i)p_i \tag{6}$$

$$\overline{CF^2} = \sum_{i=1}^{n} f(t-T_i)p_i^2. \tag{7}$$

Once appropriate parameters were identified, the testing phase began. In this step, the value 180 was assigned to x since the testing data contained 180 initial normal data flows. With the rest of the method's parameters remaining static, the method was tested using varying values of t in order to gauge sensitivity values at different false positive rates.

In order to estimate the formulas of Access Control model we use the Artificial Intelligence and Logic error method with considering modeling error in Equation (8):

$$y(t) = a_1 y(t-T) + a_2 y(t-2T) + \cdots + a_m y(t-mT) + e(t). \tag{8}$$

The error e(t) is generated because of not adopting the Access Control model to the real value. So to find the formulas, a_1, a_2, …, a_m in Equation (8) we use the sum Artificial Intelligence and Logic error and set of linear functions. Presented in Equation (9).

$$\begin{bmatrix} y(t) \\ y(t-T) \\ \vdots \\ y(t-mT) \end{bmatrix} = \begin{bmatrix} y(t)y(t-2T)\cdots y(t-mT) \\ y(t-2T)\cdots y(t-(m+1)T) \\ \vdots \\ y(t-(k+1)T)\cdots y(t-(m+k)T) \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} + \begin{bmatrix} e(t) \\ e(t-T) \\ e(t-kT) \end{bmatrix} \tag{9}$$

$$Y = \varphi \times A + E \tag{10}$$

Elements in the matrix A are the formulas which can be found by Artificial Intelligence and Logic error

method (11):

$$A = (\varphi^T \varphi)^{-1} \varphi^T Y \tag{11}$$

In Equation (11), φ^T is the transpose of the matrix φ, and $(\varphi^T \varphi)$ is the inverse of matrix. In practice:

If m is chosen larger than is required (i.e. overestimation of the model order), Equation (11), cannot be solved for any unique set of formulas, because of some columns in the matrix φ are not independent of each other. Hence $(\varphi^T \varphi)$ would be unique and will not have inverse. This means that the system of equations in Equation (8) will have an infinite number of answers for the formulas. Geometrically speaking, it is like fitting an infinite number of lines to a single point which is not the preferred case.

If m is chosen less than the required value, the number of independent equations would be more than the number of unknown variables $a_1 - a_m$. Such a system of equations has to be solved for the best approximation of formulas. The best approximation for formulas $a_1 - a_m$ is the use of the Artificial Intelligence and Logic error method.

Obviously, that is how much precision, the number of the data sent will increase and vice versa.

s previously mentioned, LANDING-C protocol divided to three phases: topology building phase, setup-state and steady-state and this method is proposed as following. These three phases is called LANDING-CEC.

a) Topology building phase

1) Start of network.

2) Base station receives position of all units that contain x and y.

3) Base station calculates distance of all units with each other's using the following Formula (12):

$$d_{ab} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \tag{12}$$

4) End of phase.

b) Setup-state phase

5) If Integrating has changed

5.1) BS calculates resource usage per unit using the Formulae (13) and (14):

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{first-amp} d^2 : d < d_{crossover} \\ lE_{elec} + l\varepsilon_{two-ray-amp} d^4 : d \geq d_{crossover} \end{cases} \tag{13}$$

$$E_{Rx}(l) = lE_{elec} \tag{14}$$

And after computing, place the sum of two variables in to the E_{ni}^t

5.2) Base station calculates remaining resource (E_R) of each unit using the Formula (15).

$$E_R = E_R - E_{ni}^t \tag{15}$$

4 Conclusions

In this paper, two intelligence agent technology techniques were applied to the problem of Access Control. First, a flow classifying method was used to detect abnormal data flows. Using 1-gram and 2-gram features, this approach achieved moderate success with Generic HTTP and Shell-code attacks but had a higher average false positive rate. Second, a flow adaptation of the relative frequency chart approach found in [6] was created using Pearson correlation to detect anomalies.

Though the chart-based approach achieved moderately better results, it required more fine-tuning because of the number of parameters used. In contrast, generalization of the classifying method was easier to achieve since it uses fewer parameters. This was evidenced by the ability of the classifying method to perform effectively on both 1-gram and 2-gram features with the same parameters, while the chart method required specific parameter tuning for each feature type.

Lastly, to better explain the performance differences between certain days, we analyzed the Pearson correlation between consecutive segments of 10 data flows. By plotting these values on a graph, concept drift and shift were visualized, and clear variations were observed between days. The location and frequency of concept shift and drift in the workflows, especially within the training phase, provided an account for the observed changes in performance.

Acknowledgement

This work has been supported by the produce-learn -research projects of the ministry of education of Guangdong province (2012B091100365) and Guangdong science and technology plan projects (2012B010500027).

References

[1] M. da S. Maximiano, M. A. Vega-Rodriguez, J. A. Gomez-Pulido and J. M. Sanchez-Perez, "A Hybrid Differential Evolution Method to Solve a Real-World Channel Access Problem," International Multi-conference on Computer Science and Information Technology, Wisia, 20-22 October 2008, pp. 201-205. doi:10.1109/IMCSIT.2008.4747240

[2] M. Yokoo and K. Hirayama, "Channel Access for Cellular Mobile Systems Using Constraint Satisfaction Techniques," Principles and Practice of Constraint Programming, Lecture Notes in Computer Science, Vol. **1713**, 1999, pp. 490-491.

[3] A. Gamst, "Some Lower Bounds for a Class of Channel Access Problems," IEEE Transactions on Vehicular Technology, Vol. **35**, No. 1, 1986, pp. 8-14. doi:10.1109/T-VT.1986.24063

[4] J. K. Hao, R. Dorne and P. Galinier, "Tabu Search for Channel Access in Mobile Radio Networks," Journal of Heuristics, Vol. **4**, No. 1, 1998, pp. 47-62.

[5] Ruo hu, Channel Access Controlling in Wireless Sensor Network using Smart Grid System, Applied Mathematics and Information Sciences, No. 6-3S (Nov. 2012), PP: 813-820.
[6] Ruo hu, Stability Analysis of Wireless Sensor Network Service via Data Stream Methods, Applied Mathematics and Information Sciences, No. 6-3S (Nov. 2012), PP: 793-798.

Hu Ruo, Man, 1968-11, Doctor, Associate professor. Major Research Areas: Information System Security, Concept Network. The main journal published: ≪Computer Science≫ Two Articles, ≪Computer Application Research≫, ≪Computer Engineering≫, ≪Journal of Tsinghua University≫, ≪Computer Engineering and Applications≫, ≪Applied Mechanics and Materials≫ (EI).