Anomaly Intrusion Detection Based upon Anomalous Events and Soft Computing Technique

Yingbing Yu

***Abstract*—Intrusion detection systems (IDSs) attempt to identify attacks by comparing new data to predefined signatures known to be malicious (misuse IDSs) or to a model of normal behavior (anomaly-based IDSs). This paper investigates a new model to more effectively detect anomaly intrusions from masqueraders. Events with different weight values based on historical data generated by Windows operating system are collected to build the normal user profiles as a template. A fuzzy system is applied to evaluate and classify the potential threat level from user new activities in a system. Experimental results show the promising results with a high detection rate of masqueraders and a low false alarm rate.**

***Index Terms*—Anomaly intrusion detection, computer security, fuzzy logic, masquerader detection.**

I. Introduction

Intrusion detection has a significant role in the overall computer security architecture. R. Bace defines intrusion detection as the process of monitoring computer networks and systems for violations of security policy (a set of laws, rules, and practices that define the system boundaries) [1]. Computer security policy is the set of laws, rules, and practices that define the system boundaries (what is permitted and what is denied) and details exactly what operations are allowed. In 1980, Anderson published a paper in which computer security threat problem was examined for the first time [2]. Denning's paper "An Intrusion Detection Model" [3] in 1987 provided a methodological framework that later inspired many research projects and commercial products.

Intrusion Detection Systems (IDSs) attempt to perform the process of monitoring computer networks and systems for violations of security policy. IDSs can be categorized into two classes based on different detection approaches. Misuse (knowledge or signature-based) IDSs look for specific patterns that define a known attack. The information about known attacks and vulnerabilities of a system is encoded into "signatures". Any actions that trigger the matches will be reported as "attempts" of intrusions.

Anomaly (behavior-based) IDSs assume the deviation of normal activities under attacks and perform abnormal detection compared with predefined system or user behavior profiles. Anomaly intrusion detection approaches have the advantage of detecting previously unknown or new attacks, but suffer from the possible high false alarms due to the difficulty of building an adaptive model.

Manuscript received May 10, 2015; revised August 12, 2015.

Yingbing Yu is with the Department of Computer Science & Information Technology, Austin Peay State University, Clarksville, TN 37044 USA (e-mail: yuy@apsu.edu).

Anomaly IDSs can be used to detect inside attacks from masqueraders, defined as internal or external intruders who exploit legitimate users identification and password obtained illegally to perform malicious attacks. Inside abuse of computer system was reported as the second most cited forms of attacks which contributed to a large portion of financial loss [4].

To prevent a system from attacks due to identity theft, the effective approach is to deploy effective anomaly IDS to monitor user behavior and report any suspicious activities. Alarms are reported when an acclaimed user (masquerader) behaves out of characters and a large deviation with the genuine user's behavior profile is detected.

To distinguish a masquerader from genuine users is a challenging task due to the problem of concept drift, where the observed user behavior may change with different tasks, time, general knowledge level and such other uncertain elements [5]. In this paper, we introduce a model of user profiling to detect masqueraders based upon a fuzzy system to evaluate the potential threat.

The rest of this paper is organized as follows. Section II is the literature review that discusses masquerader detection related to user profiling based upon command sequences and typing biometrics. Section III presents the model of using events threat evaluation based upon a fuzzy system to determine the potential threat levels in order to detect masqueraders. Section IV presents the experimental results conducted. The paper concludes with Section V, which discusses the future research work.

II. Literature Review

Access control and authentication are not sufficient to prevent potential intrusions from masquerader which already got the authorization to access system resources by obtaining an authorized user identity illegally. User behavior profiling can be used for the purpose of classification, future behavior prediction and masquerader detection. Traditionally user behavior in a system is characterized by parameters such as login frequency, location frequency, last login, session elapsed time, password fails, location fails, amount of network traffic, resources used by user in a session and so on [3].

De Ru etc. developed a software methodology that improves security by using typing biometrics to reinforce password authentication mechanisms [6]. Typing biometrics is the analysis of a user's keystroke patterns. Each user has a unique way of using the keyboard to enter a password. For example, each user types the characters that constitute the password at different speeds. The methodology employs

doi: 10.18178/ijmlc.2015.5.6.550

fuzzy logic to measure the user's typing biometrics.

Machine learning and statistical methods have been widely used for the behavior profiling from the analysis of command sequences. Davison and Hirsh developed a model called IPAM (incremental probabilistic action modeling) to predict sequences of user actions [7]. Single-step command transition probability is estimated from training data. Balajinath introduced a Genetic Based Intrusion Detector (GBID) to model individual user behavior with a 3-tuple vector which is learnt later via a genetic algorithm [8]. Ryan used a back propagation neural network NNID (Neural Network Intrusion Detector) to identify users simply by what commands and how often they use, called the 'print' of a user [9].

Lane and Brodley [10] chose a machine learning algorithm IBL (instance based learning) to measure the similarity between the most recent 10 commands of a user and the profile extracted from the past. The similarity measure is the count of matches of a new sequence with the sequences from a user's command history, with a greater weight assigned to adjacent matches.

Schonlau selected several statistics-based methods to detect masqueraders, including uniqueness, Bayes one-step Markov, Compression, Multi-step Markov chain etc. [11]. Maxion and Townsend applied Naïve Bayer classification algorithm to user profiling with command-line data [12], which shows improvement over the best approach of Schonlau.

III. Anomaly Detection Based upon Anomalous Events Profiling and Fuzzy Logic

In a typical computer system of local area network (LAN) in many organizations, it runs the active directory domain service in Windows Server 2008 or 2012 as a domain controller and many client computers running Windows or Linux OS. The Windows operating system keeps the auditing information in three separate events logs files: application log, system log, and security log.

The application event log contains events generated by user applications. The system event log records the events that are generated by the system services, drivers, and other kernel mode events. The security event logs are events related to the system, such as illegal file/directory access, invalid password entries, and illegal access to certain privileged objects. We are mainly interested in those security related events to detect the anomalous behavior either from the user or from the system itself.

In this research, we use autonomous agent software running as a service under Windows OS to detect the potential threat masqueraders in which the genuine use accounts have been compromised to prevent the internal or external penetration. The agent software works in a client/server model in which each local agent will report the suspicious events to a central station where the threat will be evaluated based on the multiple inputs from one or more client computers. The potential anomaly behavior from a masquerader is determined by comparing the user's new activities to the user's historical profile from the past events.

For Anomaly IDS, the security events should be analyzed in a short period of time (a few minutes) to prevent further potential damages or further attaches. Each event from Windows logs files has the information of user account, date and time, event ID, event type and other information. In addition, the number of occurrence for each event will be identified. For each user, we also assign a weight value of importance for each events based on the ration of the occurrence of one events and the total number of events in the historical data. There is a threshold of each event for each user which is based on the average number of occurrence in the past plus a variation. These threshold values are indicators of evaluating the potential threat whenever it is beyond the normal limits.

The system will evaluate the threat for each user at each monitoring period based on the number of the anomalous events and their weights. In this research, the occurrence of each event from a user will be compared to the user profiles to determine the degree of deviation as a fuzzy membership. It compares each event current occurrence with the equivalent historical thresholds value, to determine if this measure is abnormal or not.

Three fuzzy sets "low"', "medium" and "high" are selected to measure the magnitude. The common triangular membership function is chosen in the model (Fig. 1). There is no overlap between fuzzy sets "low" and "high" and the point "b" splits the section "ac". The point at which two fuzzy sets intersect has the membership value of 0.5 with both fuzzy sets. An event can be assigned to one or two categories from the fuzzy membership definition.

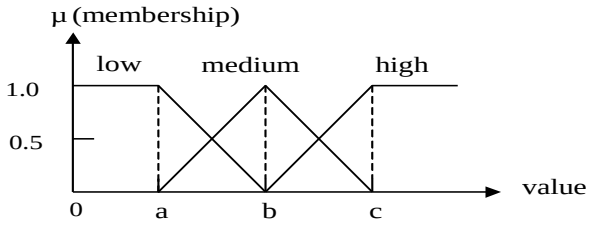

Fig. 1. Fuzzy triangular membership.

For each user, many events will be generated and events differ in their weights. The event's threat weight value is the same value as the event's weight in the user profiles built from the past historical activities. The Anomaly Detection evaluates the threat for each user at each monitoring period based on the number of the anomalous events and their weights. It calculates the threat level by adding the event weight for all the anomalous events.

Another three fuzzy sets "Normal", "Potential", and "Anomalous" are selected to measure the corresponding possibility of potential intrusion to the system. The same triangular membership function with different splitting points is selected. For a test case, if there is no obvious deviation compared with the behavior profile, it belongs to the fuzzy set "Normal". "Potential" means that the user or program actions are suspicious in some degree. "Anomalous" means that the behavior deviation is large enough to report alarms.

For a fuzzy inference system, the knowledge base consists of a collection of IF-THEN fuzzy rules. The inference engine, the backbone of the fuzzy system, conducts a mapping from input fuzzy sets to output fuzzy sets based on these fuzzy

rules. The following three rules are developed to evaluate the threat level for each event of a user.

Rule 1: If the deviation of an event is low, then the case is in the category of "Normal"

Rule 2: If the deviation of an event is medium, then the case is "Potential"

Rule 3: If the deviation of an event is low, then the case is "Anomalous".

After the membership values of facts with respect to each antecedent in a rule are determined, the MAX-MIN method is applied to measure the impact of fuzzy rules and the highest membership is selected. The inputs are combined logically with the MIN operator to produce a minimum firing strength for each rule. The consequent memberships of multiple rules are aggregated to get the overall output degree of membership using the MAX operator to produce a maximum value.

The defuzzification process is the mapping form a space of fuzzy actions defined over an output universe of discourse into a space of non-fuzzy (crisp) control actions. The number of occurrence's output linguistic value for event is converted to a numerical value (*Δthreat*), which is calculated using the center of area (COA) defuzzification method. The COA formula is:

$$\Delta\text{Threat} = \frac{\sum_{k=1}^{n} \mu_k * \text{center}(k)}{\sum_{k=1}^{n} \mu_k} \quad (1)$$

where n is the number of fired rules, μ_k is the degree of membership of rule k, and *center*(k) is the peak-value where the fuzzy set for the rule k has the maximum membership values. This final *ΔThreat* value can be compared with a predefined threshold value to determine the degree of the threat to the system for a certain test case.

The overall threat evaluation process is used to calculate the total threat evaluation level (T) that is corresponding to the total number of events. It reads all the events generated in the current monitoring period and calculated the Δ threat values *using the fuzzy system*. Then, it calculates the total threat evaluation level (T) by multiplying the event's weight values by their Δ threat values in the total threat array. The total threat evaluation value (T) is calculated using the formula:

$$T = \sum_{i=1}^{m} \Delta\,\text{threat}(E_i) * wgt(E_i) \quad (2)$$

where m is the number of events in the period of time, wgt (E_i) is the weight corresponding to the number of occurrence of event i in the user historical profiles.

To represent the threat level, five threat levels are defined according to the threat percentage value. The following heuristic linguistic variables appear to be effective. The "very Low" level indicates that the user activities are not associated with an attack. This level is defined for any threat evaluation value from 0 % to 15 %. Further investigation may be need to either classify as attacks or integrate into the user profiles if it is not a real attack but just the normal drifting behavior patterns.

The level "Low" indicates that there is a low possibility of attacks or the new activities are out of normal profiles from the past. This level represents any threat evaluation value greater than 15 % and up to 30%. The "Potential" level indicates that the user activities indicate a possible attack or existence of masqueraders. This level represents any threat evaluation value greater than 30 % and up to 60 %.

The "Suspicious" level indicates that the user activities indicate a level where it is likely that some potential attacks have taken place. This level represents any threat evaluation value from 60 % to 80 %. The "Anomalous" level indicates that the user attacks occur with a high degree of certainty. This level represents any threat evaluation value greater than 80% and up to 100 %.

IV. Experimental Results

We have tested the use profiling model in a real system to detect the potential masqueraders. The environment is a lab of 30 PCs which runs Windows Server 2012 OS as a local area network (LAN). The anomaly detection software is run on each computer to report the real time of events generated in every 10 minutes to a central station where the analysis is conducted to evaluate the potential threat. We have built the normal user behavior profiles of ten users over a period of six months in the lab. Then we disclosed the ten user account and password information to a large group of students to log in using one or more of the known accounts to either conduct regular tasks or try to perform computer or network attacks.

The software running in the lab will collect the new activities generated from these masqueraders to check if it can identify as potential threat at least in the threat level of "Alert" or "Anomalous". In addition, we also asked the original ten users to still log into the system as usual to generate some new events. These will be used to make sure that the system will not incorrectly label as abnormal and thus false alarms will be generated. The high false alarm rate is a major issue especially for anomaly IDSs since resources have to be allocated to investigate while it may trigger the system to be abandoned if it overwhelms the people to become unable to identify the real attacks instead.

Table I shows the results of successful masquerader detection rate and false alarm rate. For most users, the model can achieve a relatively high masquerader detection rate. The average detection rate for the 10 users is about 84.0% (840/1000) and the missing percentage is 10.6% (106/1000). If a normal case is classified incorrectly as abnormal, a false alarm is generated. In the experiment, only 16 of 200 normal cases are incorrectly identified as "abnormal" and the false alarm rate is 8.0%.

For a real intrusion detection system, it is the ultimate goal to detect masqueraders within a short time interval and alert the system earlier to prevent further loss. Based on the experimental results, the interval of 10 minutes of user activities achieves both a high detection rate and a low false alarm rate. We also analyze the data using a large period of time (20 minutes and 30 minutes) and the model achieves a

high detection rate while at still a very low false alarm rate.

We still think that an anomaly IDS in a real environment should be able to report the potential threat in a relative short period (5 minutes or 10 minutes) for the possible prevention of further damage to the system or possible deployment of anti-attack techniques. In general, it is fairly reasonable and effective for an anomaly IDS to detect potential masqueraders after about 10 minutes of user activities.

TABLE I: ANOMALY DETECTION RATE AND FALSE ALARM RATE

Users	Detection Rate	False Alarm Rate
User 1	81.0% (81/100)	5.0% (1/20)
User 2	84.0% (84/100)	5.0% (1/20)
User 3	85.0% (85/100)	5.0% (1/20)
User 4	79.0% (79/100)	15.0% (3/20)
User 5	92.0% (92/100)	10.0% (2/20)
User 6	80.0% (80/100)	0.0% (0/20)
User 7	88.0% (88/100)	5.0% (1/20)
User 8	83.0% (83/100)	10.0% (2/20)
User 9	77.0% (77/100)	20.0% (4/20)
User 10	91.0% (91/100)	5.0% (1/20)
Average	84.0% (840/1000)	8.0% (16/200)

V. CONCLUSIONS

In this paper, we introduce a model of anomaly detection based on user profiling from event logging and a fuzzy system to detect attacks from masquerades. Experiments conducted show promising results. In the future, we want to extend the current research of masquerader detection based on other user activities such as user commands execution in Unix/Linux system.

For example, as we have notices that in the last decade GUI and Internet-based applications have been deployed in both UNIX and Windows systems. A large of part of user activities associated with these applications may not involve individual commands directly entered into the system, but instead consist of mouse clicks on icons. The behavior modes from this kind of activity will differ significantly from those discussed in this paper. Future work would address these questions.

REFERENCES

[1] R. Race, *Intrusion Detection*, Macmillan Technical Publishing, 2nd Edition, Indiana, 2000.

[2] J. P. Anderson, "Computer security threat monitoring and surveillance," Technical Report, James P. Anderson Co., Fort Washington, PA., 1980.

[3] D. E. Denning, "An intrusion detection model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222–232, February 1987.

[4] Computer Security Institute and Federal Bureau of Investigation, *2013 CSI/FBI Computer Crime and Security Survey*, Computer Security Institute publication.

[5] T. Lane, "Machine learning techniques for the computer security domain of anomaly detection," PhD dissertation, Purdue University, W. Lafayette, IN, August 2000.

[6] D. Ru and W. G. Johannesburg, "Enhanced password authentication through fuzzy logic," *IEEE Expert*, vol. 12, no. 5, pp. 38-45, 1997.

[7] B. Davison and H. Hirsh, "Predicting sequences of user actions," in *Proc. AAAI Workshop on Predicting the Future: AI Approaches to Time-Series Problems*, July 1998, Madison, Wisconsin, pp. 5-12.

[8] B. Balajinath and S. V. Raghavan, "Intrusion detection through learning behavior model," *Computer Communication*, vol. 24, pp. 1202-1212, 2001.

[9] J. Ryan, M. Lin, and R. Miikkulainen, "Intrusion detection with neural networks," *Advances in Neural Information Processing Systems*, vol 10, pp. 943-949, 1998.

[10] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Transactions on Information and System Security*, vol. 2, no. 3, pp. 295-331, 1999.

[11] M. Schonlau, W. DuMouchel, W. Ju, A., Karr, M. Theus, and Y. Vardi, "Computer intrusion: Detecting masquerades," *Statistical Science*, vol. 16, no. 1, pp. 58-74, 2001.

[12] R. Maxion and T. Townsend, "Masquerade detection using truncated command lines," in *Proc. International Conf. on Dependable Systems & Networks*, Washington DC, June 23-26, 2002, pp. 219-228.

Yingbing Yu received his PhD degree in computer science and engineering from University of Louisville, Kentucky, USA in 2005. His research interests include computer security, intrusion detection, network security, soft computing, intelligent systems, fuzzy logic, wireless network and its security. He currently is an associate professor in the Department of Computer Science and Information Technology, Austin Peay State University, Tennessee, USA.