International Journal of Engineering Research
Volume No.6, Issue No.4, pp :220-223
ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

A Survey on Object-based Masquerade Detection System using Temporal and Spatial Locality Features

Priyanka K. Gaikwad
Department of IT, Walchand College of Engineering, Sangli, Maharashtra, India
priyanka.gaikwad@walchandsangli.ac.in

Abstract : ***Masquerades in computer intrusion detection is the person who unauthorized access somebody else computer account. Masquerade attacks are very difficult to detect because of they are mostly carried by insiders. The Masquerade Detection Systems (MDS) are used to prevent the user computer from the masquerade. In this paper, the problem of the masquerade detection based on different types of user behavior are considered. Previously most of the MDS are focus mainly on the user action and ignoring the object upon which that action is performed. Mostly the masquerades do not know the file system and layout of the user desktop so that they search more broadly in a manner that is different than the user. In this paper, the different masquerade detection approaches are reviewed and their results are compared.***

Keywords : ***MDS,***

I. Introduction

Information security is a challenge for organizations. Existing security technique's like firewalls, intrusion detection systems and antivirus are not enough to secure the information. Computers are used to store information so that users mostly depend on the computer and it increases computer dependency but does not increase the use of proper mechanisms for securing information. Still, users set a weak password or leave their computer sessions unattended. Masquerading is the process where a person spoofs someone else's identity and utilizes privileges which he is not entitled to. Therefore the Masquerade Detection Systems (MDS) are developed. The objective of such systems is to raise an alert when computer behavior changes to a certain extent from common computer behavior [1].

Masquerading can be performed by insiders or outsiders. In outsider masquerading the attacker can gain the root access to a victim's machines remotely and steal the information or cause the damage to the system. If an insider misuse the privileges of a user within an organization then it is difficult to detect the masquerader or who is the attacker? Because when insider tries to gain the access of another user inside the organization, most of his actions may be technically legal. Also, the insider has the enough knowledge about the user system as well as the behavior of the users.

Most research in MDS focuses on the user action and ignoring the object upon which that action is performed, for example, command execution (an action) usually ends up in the transformation of a file (the object). But the object upon which an action is carried out is important for distinguishing a user from a masquerade and this type of MDS is based on the locality features. There are two types of locality: temporal and spatial. The temporal locality is applicable to both action and objects while spatial locality is suitable to objects.

II. Overview of Masquerade Detection System

The MDS can detect both the known and unknown attacks if the attacker`s behavior is different from a normal user. Fig 1. Shows that the types of MDS. The command based masquerade detection systems are based on the command those are used by the user. The GUI based MDS are used to capture the user activities performed using either mouse movement and clicks or keyboard usage. The file-based MDS observing how user travel in his file system while the object based MDS are mainly focusing on the object that is files and folders.

III. Significance of Survey

Masquerade is one type of attack where an attacker behaves like an authorized user to gain the access to the user system. The fundamental assumption of masquerade detection is that each user has his unique characteristics when invoking commands. Hence, from this survey studied that the intrusion likely occurs when there is a significant difference from user's previous characteristics.

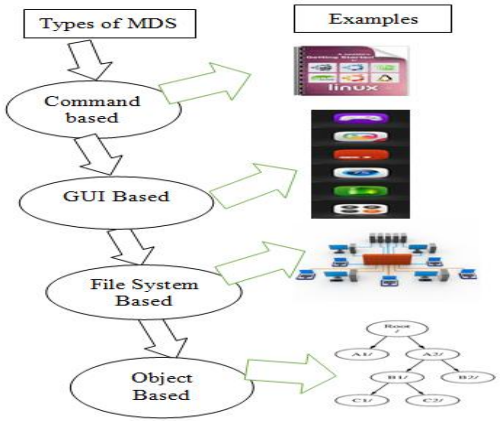

Fig 1. Types of Masquerade Detection System

ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

A. The Command-Based Masquerade Detection

The Command-Based Masquerade Detection Systems are mainly focusing on the user commands those are commonly executed. To implement this system the Schonlou *et al*. developed a SEA dataset [2] which is based on the Unix commands. SEA is a log of Unix commands that were recorded the 15000 commands for each of about 70 users. In this dataset, the author had randomly selected 50 users and the remaining 20 users as masquerades. The Naïve Bayes classifier is one of the most popular classifiers and it was first applied on the SEA dataset by Roy A. Maxion. In SEA dataset the argument of the commands was not collected because of privacy concerns. To solve this problem Maxion *et al*. developed Greenbergs dataset that includes information about both commands and arguments to prove the performance results obtained with SEA [3]. The Greenberg dataset contains logs of four different type of users; non-programmer, novice programmers, experienced programmers and computer scientists. After that Maxion *et al.* extended their previous work by applying the Naïve Bayes classifier on Greenberg's command line data. Similar to SEA dataset Greenberg dataset is also failed to attempt the attack.

B. Masquerade Detection based on GUI Data

GUI based systems are designed to be more visual and the users use the mouse clicks for activating commands rather than typing the commands on terminal. Command line data is not capable of capturing the user actions such as mouse movements, keyboard typing speed etc. Masquerading attack refers to the illegitimate activity on a computer system when one user impersonates another user. Mostly the masquerade attacks are carried by an insider and thus they are difficult to detect. In Graphical User Interface (GUI) based system the detection of these attacks is done by monitoring significant changes in user's behavior based on user profile. But in today's era, some profiles are based on the user command line data and so it does not represent user complete behavior. Hence, the author presents a new framework by creating a unique feature set for user behavior on GUI based system. Author has collected real user behavior data from live systems and extracted parameters to construct these feature vectors. These contain the user information such as mouse speed, distance, angles and amount of clicks during a user session [4]. The set of the actions which can be used to detect the masquerade attacks is known as the behavior profile.

Ashish Garg *et.al* collect the real user behavior data for multiple users and extract unique parameters to construct the feature vectors [5]. Hence, author developed an active system logger using Microsoft .NET Framework and C# language on windows XP system. The .NET Framework is chosen due to its ease of use and ability to interact with various windows components. This logger collects all the possible events those are performed by the user on the system in real time. The logger collects the event data such as keyboard activity, mouse movement co-ordinate and mouse clicks, system background processes and user run commands.

The Human keyboard interaction has been widely used for masquerade detection. The Killourthy and Maxion have developed a dataset of 51 users [6] [7]. The dataset contains the information about how the user typed the same password 400 times and extracting the 31-time features, including hold time, keydown - keydown time etc. The Killourthy and Maxion trained several classifications algorithms using different methodologies. But these methods are only applicable when both user and intruder type the same text.

In GUI based MDS the support vector machine is used for classifying the data. The SVMs are maximal-margin classifiers as compared to Naïve Bayes and it is probabilistic. Also, SVM algorithm provides better classification results with less training data.

C. Masquerade Detection System based on user search behavior

To detect the attack machine learning algorithms are apply that produce classifiers which can identify the suspicious behavior. The author Salem and Yingbo *et al.* use the user's search patterns for masquerade detection [9] [10]. For this eighteen users are monitored for four days to collect the data and develop the dataset RUU. After analyzing RUU dataset author shows that normal users display different search behavior and by using this behavior masquerade will be detected.

D. Masquerade Detection based on File System and File Similarity

The File is an important carrier of information. Usually, attackers are not familiar with the file system of the target computer, so the attacker may carry a large amount of file searching behaviors to find out the valuable files. While normal processes only need to access certain files to complete their functions. Therefore Wang *et.al* developed a method for detecting malicious process [8] [11]. The FPD-based method is used for detecting abnormal file access behaviors. File Path Diversity (FPD) is a quantized value which measures how far a set of file paths is spread out and it measures abnormal file access behaviors and normal ones, which is used to detect the malicious processes that controlled by attackers to search and steal valuable files. An author presenting an algorithm for calculating FPD and also developed a prototype system based

ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

on FPD for detecting malicious processes. The author defining the FPD function. *f* is the file and *p* are the most popular paths that file take. If *f* is a descendant of *p*, then fpd equals to 0. The main problem with this method is how to compute the most popular path which is used to calculate FPD function

Gates *et al.* claim that if files that one user currently access have been accessed in the recent past by the same user then this is the legitimate action and if the files are similar to the file previously accessed then this is less likely to be malicious theft. The author uses a function that assigns a score for each file when the access history of that file is given. The author proposed the five ways to evaluate score. The first method is NewUnique. It is a binary method: It scores 0 if the two files are same and 1 otherwise. The next 3 methods are Full Distance, Least Common Ancestor (LCA) and log LCA are based on the distance between two files. The fifth method is the Access Similarity, how similar two files are, depending on the set of users that access them [12]. The author's successfully applied the techniques to profile identification and anomaly detection.

E. Object-based Masquerade Detection

Most of the MDS focus mainly on the user action and ignoring the object upon which that action is performed. Here the file is considered as the object. J. Camina *et al.* prove that the object is more important to distinguish a user from the masquerade [15]. The author developed the file system navigation approach and tested it using the Windows-Users and Windows Intruder simulations log data set (WUIL) [14]. A dataset *contains* the file access logs from 20 users. There are two types of locality: temporal and spatial. The temporal locality is applicable to both action and objects while spatial locality is suitable to objects. Author compare the performance of two classifiers, one built with locality features and other with the WUIL features. There are four spatial locality and eight temporal locality features. To compute these features check how the user navigate in her file system. If the user is the legitimate user then he may travel the short distance within 30 sec. The TreeBagger classifier is used to compare the results of locality with WUIL dataset. TreeBagger classifier is the MATLAB R2014a based and it is a bootstrap aggregating algorithm generates then random trees using the training set. The classifier based on the locality features gives the better performance than the other classifiers.

IV. Comparative Study

Table 1 shows that the comparison between previously used masquerade detection system. In which most of the MDS are based on the user action so that it does not give the better performance results. But in the object based MDS extract the locality features and locality features are very effective to detect the masquerade.

Table 1: Comparision between different types of MDS

MDS types	Command based [2]	Command and argument based	GUI based [4]	File System based [5]	Object-based [8]
Dataset	SEA dataset	Green berg	GUI based dataset	-	WUIL dataset
No. of users	70 users	70 users	51 users	-	80 users
Classifier	Naive Bayes classifier	Naive Bayes classifier	Support vector machine	-	TreeBagger classifier
Advantages	Widely used in masquerade detection	Both the arguments as well as commands are collected which	It is used to uniquely identify users hence, provides better	To detect the masquerade and malicious processes in the file system file	The main focus is on the object (file). Locality features are better to separate a user
Disadvantages	The arguments of the commands are not collected	Failed to attempt the attacks. To simulate a	The methods [6] are only applicable when both user and	Difficult to compute the most popular path which is used to	Does not merge the locality and WUIL features. Treats them

V. Remark

Masquerade attack is a serious computer security problem. The aim of this paper is to study the various types of masquerade detection systems, including datasets, classifiers, and results. Most of the existing methods for user profiling have focused mainly on one type of activity like command usage, keyboard usage etc. After the overall survey, conclude that the only user behavior is not enough to detect the masquerade. The object is also important to classify the user as well as a masquerade. Previously, all MDS are based on the user action and ignoring the object. The results obtained from the classifiers on the user actions are not sufficient to classify the user and the masquerade,

ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

but the classifier which is based on the locality features gives a better performance than other classifiers.

REFERENCES

i. *M. Schonlau, W. Dumouchel, W. Ju, A. F. Karr, M. Theus, and Y. Vardi, "01Statscigalley", vol. 16, no. 1, 2001.*

ii. *R. A. Maxion, "Masquerade detection using enriched command lines", in Proc. 43rd Int. Conf. Dependable Syst. Netw. (DSN), pp. 5–14, Jun. 2003.*

iii. *M. Schonlau. (May 20, 2015). Masquerading User Data. [Online]. Available: http://www.schonlau.net/intrusion.html*

iv. *M. Pusara and C. E. Brodley, "User re-authentication via mouse movements", in Proc. ACM Workshop Vis. Data Mining Comput. Security (VizSEC/DMSEC), pp. 1–8, 2004.*

v. *Garg, R. Rahalkar, S. Upadhyaya, and K. Kwiat, "Profiling users in GUI based systems for masquerade detection", IEEE Work. Inf. Assur. pp. 48–54, 2006.*

vi. *K. Killourhy and R. Maxion, "Why did my detector do that?! Predicting keystroke- dynamics error rates", in Proc. 13th Int. Symp. Recent Adv. Intrusion Detection (RAID), pp. 256–276, Sep. 2010.*

vii. *K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in Proc. Int. Conf. Dependable Syst. Netw. (DSN), pp. 125–134, Jun./Jul. 2009.*

viii. *B. Camina, R. Monroy, L. A. Trejo, and E. Sanchez, "Towards building a masquerade detection method based on user file system navigation", Lect. Notes Comput. Sci. (including Subsea. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7094 LNAI, no. PART 1, pp. 174–186, 2011.*

ix. *M. B. Salem and S. J. Stolfo, "Modeling user search behavior for masquerade detection", in Proc. 14th Int. Symp. Recent Adv. Intrusion Detection (RAID), pp. 181– 200, Sep. 2011*

x. *Y. Song, M. B. Salem, S. Hershkop, and S. J.Stolfo, "System level user behavior biometrics using Fisher features and Gaussian mixture models", in Proc. Security Privacy Workshops (SPW), pp. 52–59, May 2013.*

xi. *X. Wang, Y. Sun, and Y. Wang, "an Abnormal File Access Behavior Detection Approach Based on File Path Diversity", in Proc. Int. Conf. Inf. Commun. Technol. (ICT), pp. 1–5, May 2014.*

xii. *C. Gates, N. Li, Z. Xu, S. N. Chari, I. Molloy, and Y. Park, "Detecting insider information theft sing features from file access logs", in Proc. Eur. Symp. Res. Comput. Security (ESORICS), vol.8719, pp. 383–400, Sep. 2014.*

xiii. *J. B. Camiña, J. Rodríguez, and R. Monroy, "Towards a masquerade detection system based on user's tasks", in Proc. 17th Int.Symp. Recent Adv. Intrusion Detection (RAID), vol. 8688, pp. 447–465, Sep 2014.*

xiv. *J. B. Camiña, C. Hernández-Gracidas, R. Monroy,and L. Trejo, "The windows-users and intruder simulations logs dataset (WUIL): An experimental framework for masquerade detection mechanisms", Expert Syst. With Appl., vol. 41, pp. 919–930, Feb. 2014.*

xv. *J. B. Camiña, R. Monroy, L. A. Trejo, and M. A. Medina-pérez, "Temporal and Spatial Locality : An Abstraction for Masquerade Detection", IEEE Trans. Inf. Forensics Secur.vol. 11, no. 9, pp. 2036–2051, 2016.*