ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 6, June 2017

Automated Discovery Of Internal Attacks

[1]Saiteja, [2]Abdul Azeez
[1]msaiteja933@gmail.com, [2]md.ab.azeez.aa@gmail.com.

Abstract: ***Cyber security is that the body of technologies, processes and practices designed to safeguard networks, computers, programs and knowledge from attack, harm or unauthorized access. During a computing context, the term security implies cyber security. This survey paper describes a targeted literature survey of machine learning (ML) and data processing (DM) strategies for cyber analytics in support of intrusion detection. This paper focuses totally on cyber intrusion detection as it applies to wired networks. With a wired network, associate oppose must experience many layers of defense at firewalls and operative systems, or gain physical access to the network. The quality of ML/DM algorithms is addressed, discussion of challenges for victimization ML/DM for cyber security is conferred, and some recommendations on once to use a given methodology area unit provided.***

I. INTRODUCTION

Generally, among all well-known attacks such as pharming attack, distributed denial-of-service (DDoS), eavesdropping attack, and spear-phishing attack insider attack is one of the most difficult ones to be detected because firewalls and intrusion detection systems (IDSs) usually defend against outside attacks. To authenticate users, currently, most systems check user ID and password as a login pattern. However, attackers may install Trojans to pilfer victims' login patterns or issue a large scale of trials with the assistance of a dictionary to acquire users' passwords. When successful, they may then log in to the system, access users' private files, or modify or destroy system settings. Fortunately, most current host-based security systems and network-based IDSs can discover a known intrusion in a real-time manner.

II. PROPOSED WORK

In this paper, we have proposed an approach that employs data mining and forensic techniques to identify the representative SC-patterns for a user. The time that a habitual SC-pattern appears in the user's log file is counted, the most commonly used SC-patterns are filtered out, and then a user's profile is established. By identifying a user's SC-patterns as his/her computer usage habits from the user's current input SCs, the IIDPS resists suspected attackers.

MODULES:

USER INTERFACE DESIGN:

In this module, we design the windows for the project. These windows are used to send a message from one peer to another. We use the Swing package available in Java to design the User Interface. Swing is a widget toolkit for Java. It is part of Sun Microsystems' Java Foundation Classes an API for providing a graphical user interface for Java programs. In this module, mainly we are focusing the login design page with the Partial knowledge information. Application Users need to view the application they need to login through the User Interface GUI is the media to connect User and Media Database and login screen where user can input his/her user name, password and password will check in database, if that will be a valid username and password then he/she can access the database.

CONTROL CENTER INITIALIZATION MODEL:

This is the second and important module in our application Manager Interface Design plays an important role for the Manager to move login window to Manager welcome window. Manager Will enter the salary details of users and allocating the project to the Team leader.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 6, June 2017

III. MINING USER AND ATTACKER HABITS:

An insider attacker may log in to a system by using another user's login ID and password and do something maliciously. However, attackers may install Trojans to pilfer victims' login patterns or issue a large scale of trials with the assistance of a dictionary to acquire users' passwords. When successful, they may then log in to the system, access users' private files, or modify or destroy system settings. Fortunately, most current host-based security systems and network-based IDSs can discover a known intrusion in a real-time manner. However, it is very difficult to identify who the attacker is because attack packets are often issued with forged IPs or attackers may enter a system with valid login patterns The IIDPS processes SCs collected in u's log file with a sliding window, named a log-sliding window (L-window for short), which is used to identify consecutive SCs of size along their submitted sequence and partition the SCs in the window into k-grams where k is the number of consecutive SCs, k =2, 3, 4, . . .

IV. IMPLEMENTING ATTACK ANALYZER:

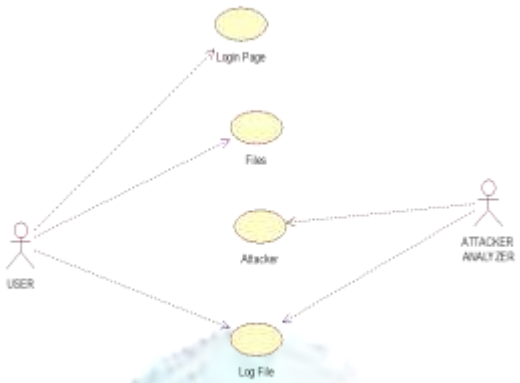
Internal Intrusion Detection and Protection System (IIDPS), which detects malicious behaviors launched toward a system at SC level. The IIDPS uses data mining and forensic profiling techniques to mine system call patterns (SC-patterns) defined as the longest system call sequence (SC-sequence) that has repeatedly appeared several times in a user's log file for the user. The user's forensic features, defined as an SC-pattern frequently appearing in a user's submitted SC-sequences but rarely being used by other users, are retrieved from the user's computer usage history.

V. ATTACK GRAPH MODEL:

This is the second module of our project in this with the advent of web applications, an attack graph is a modeling tool to illustrate all possible multi-stage, multi-host attack paths that are crucial to understand threats and then to decide appropriate countermeasures. In an attack graph, each node represents either precondition or consequence of an exploit. The actions are not necessarily an active attack since normal protocol interactions can also be used for attacks. Attack graph is helpful in identifying potential threats, possible attacks and known vulnerabilities in a cloud system. Since the attack graph provides details of all known vulnerabilities in the system and the connectivity information, we get a whole picture of current security situation of the system where we can predict the possible threats and attacks by correlating detected events or activities.

VI. SYSTEM DESIGN

Use Case Diagram:



FUTURE ENCHANCEMENT:

For future work, further study will be done by improving IIDPS's performance and investigating Third-party shell command. This paper presents intelligent lightweight IDS, which used the forensics technique to profile the user behavior to automate the maintenance of user profile, data mining technique to find out the cooperative attack, and watermark technique to trace back the hackers or intruders. The goal of the system is to detect the intrusion real-time, effectively and efficiently.

CONCLUSION:

In this paper, we have proposed an approach that employs data mining and forensic techniques to identify the representative SC-patterns for a user. The time that a habitual SC pattern appears in the user's log file is counted, the most commonly used SC-patterns are filtered out, and then a user's profile is established. By identifying a user's SC-patterns as his/her computer usage habits from the user's current input SCs, the IIDPS resists suspected attackers.

REFERENCES:

1.S. Gajek, A. Sadeghi, C. Stuble, and M. Winandy, "Compartmented security for browsers—Or how to thwart a phisher with trusted computing," in *Proc. IEEE Int. Conf. Avail., Rel. Security*, Vienna, Austria, Apr. 2007, pp. 120–127.

2. C. Yue and H. Wang, "BogusBiter: A transparent protection against phishing attacks," *ACM Trans. Int. Technol.*, vol. 10, no. 2, pp. 1–31, May 2010.

3.Q. Chen, S. Abdelwahed, and A. Erradi, "A model-based approach to self-protection in computing system," in *Proc. ACM Cloud Autonomic Comput. Conf.*, Miami, FL, USA, 2013, pp. 1–10. 2010.

4. F. Y. Leu, M. C. Li, J. C. Lin, and C. T. Yang, "Detection workload in a dynamic grid-based intrusion detection environment," *J. Parallel Distrib. Comput.*, vol. 68, no. 4, pp. 427–442, Apr. 2008.

5. H. Lu, B. Zhao, X. Wang, and J. Su, "Diff Sig: Resource differentiation based malware behavioral concise signature generation," *Inf. Commun. Technol.*, vol. 7804, pp. 271–284, 2013.

6. Z. Shan, X. Wang, T. Chiueh, and X. Meng, "Safe side effects commitment for OS-level virtualization," in *Proc. ACM Int. Conf. Autonomic Comput.*, Karlsruhe, Germany, 2011, pp. 111–120.

7. M. K. Rogers and K. Seig fried, "The future of computer forensics: A needs analysis survey," *Comput. Security*, vol. 23, no. 1, pp.12–16, Feb. 2004.