Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

A Survey on Intrusion Detection Methods

Rajalekshmi V[1], Archana G[2]
[1]*Computer Science and Engineering, Sree Buddha College of Engineering For Women*
[2]*Computer Science and Engineering, Sree Buddha College of Engineering For Women*

Abstract—— In high speed networks intrusions are emerging day by day using different techniques. Intrusion detection methods are also available in various ways. Intrusions can be detected using graphs, using programs etc. This paper explains some of the intrusion detection methods
Keywords— Spam zombies; Intrusion detection; attack graphs; cloud computing; DDoS

I. INTRODUCTION

Intrusions are those activities that violate the security policies of a system and intrusion detection systems are used to identify those intrusions. As the networks and new technologies are growing fast new intrusions are also introduced day by day. Therefore supervision of communication systems get more complicated. Intrusions are detected using alerts. Correlation of alerts is by using their similar attributes, prior knowledge and corresponding vulnerabilities. Vulnerabilities are weakness in which an attacker can exploit that weakness and thus cause intrusions.

Compromised machines are those machines which are affected by vulnerabilities and these are one of the key security threats on the internet. Several distributed Denial of Service attacks have shown the necessity of better protecting computers and networks connected to the internet. These networks consist of vulnerable computers and these are the reason for an attacker to attack a network. Interesting information for the attacker in these networks cause denial of services.

Distributed Denial of Services has shown the necessity of better protecting computers and networks connected to the network. Nowadays many attack tools are available in the world, so any person without any depth knowledge of attack can be easily attack a computer or a network.

Cloud computing is another important concept. Cloud computing refers to both the application delivered as services over the internet and the hardware and the systems software in the datacenters they provide services [2]. It provides three layers of services: PaaS (Platform as a Service), IaaS (Infrastructure as a Service) and SaaS (Software as a Service). Attacks are mainly affected on IaaS. The datacenter hardware and software is what we called as cloud. When a cloud is made available in a pay-as-you go manner to the general public then it is called public cloud; the services being sold is called utility computing. Internal datacenters of a business or organizations that are not available to public is called private cloud. Cloud users face security from both inside and outside the cloud. The cloud user is responsible for application level security, denial of service attacks by users, protecting cloud from providers etc. Protecting from providers because as by definition provider controls the bottom layer of the software stack. Primary security mechanism towards cloud is virtualization. The methods that are explained in the later sections can also be used in virtual networks and thus protect clouds.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes some of the detection methods, section 4 represents acknowledgment and last section, section 5 represents the conclusion of this paper by analyzing the explained methods in section 3.

II. RELATED WORKS

This section provides some of the papers that are referred for detection methods.

The area of detecting malicious behavior has been explored. The work by Duan [3] explains a new method called SPOT that is derived from a powerful mathematical statistical tool called SPRT. It

provides an effective spam zombie detection system by examining outgoing messages from a network.

BotHunter [4] and BotSniffer [5] are other two method for detecting botnets in a network. Botnets are malwares that are established using a command and control through channels that they are updated and directed.

Ou et al. proposed MuLVAL [6], an attack graph tool, which adopts a logic programming approach and uses datalog language to model and analyze the network. Using this it is possible to get all the multihost, multistage attack paths.

Roy et al. proposed ACT [8] consider attacks and countermeasures on a single tree structure. They devised several objective functions based on greedy and branch and bound techniques to minimize the number of countermeasure, reduce investment cost, and maximize the benefit from implementing a certain countermeasure set.

Wang et al. [7] devised an in memory structure, called queue graph (QG), to trace alerts matching each exploit in the attack graph. However, the implicit correlations in this design make it difficult to use the correlated alerts in the graph for analysis of similar attack scenarios.

NICE proposed by Chung et al. [1] is a multiphase distributed vulnerability detection, measurement and countermeasure selection mechanism in virtual networks which is built based on attack graph based analytical models and reconfigurable virtual network based countermeasures.

III. SOME INTRUSION DETECTION METHODS

This section explains some of the intrusion detection methods.

3.1. SPOT

SPOT is an effective zombie detection system by monitoring the outgoing messages from a network. It is designed based on a powerful statistical tool called Sequential Probability Ratio Test (SPRT) which has bounded false positive and false negative error rates. This method can be used to test between two hypotheses, as the events occur sequentially. SPOT detects spam zombies which mean machines those send spam machines. Since SPRT has many features it can be effectively used for detecting spam zombies. This technique uses three algorithms, first is the SPOT detection algorithm, second is the spam count and percentage based detection algorithm and third one is impact of dynamic IP addresses.

In the first algorithm when an outgoing message is arrived at the SPOT detection system it recorded its IP address so that it can classified it into either spam or non spam using spam filter. In this SPOT want to detect machines of two cases. One is the identification of normal machines in a network and second is to continuously monitoring the machines that are determined to be normal by SPOT.

Second algorithm calculates the number of spam messages and percentage of spam messages sent from an internal machine. Third algorithm formally evaluates the impact of dynamic IP addresses on detecting spam zombies.

3.2. BotHunter and BotSniffer

A bot is a, malware, self propagating application that infects vulnerable hosts through direct exploitation or Trojan insertion. They differ from other malwares by their ability to establish a command and control (C&C) channel. If a machine is under the control of C&C server then it forms Botnets.

BotHunter is an evidence trail approach through communication sequences that occur during the infection process. This approach is called dialog correlation strategy. Here bot infections are considered as a set of loosely ordered communication flows that are exchanged between an internal host and one or more external entities. BotHunter correlator is driven by Snort and it consists of two additional bot specific anomaly detection plug-ins for malware analysis: SLADE and SCADE. SLADE analyze incoming traffic flow and targeting selected protocols that are indicative of common malware infections. SCADE performs port analysis to both incoming and outgoing network traffic.

BotSniffer is also based on anomaly detection algorithms and is implemented as several plug-ins for the open source Snort. This results in the high accuracy of detecting botnets C&Cs with a very low false positive rate.

3.3. MuLVAL

MuLVAL is an end to end framework and a reasoning system that conducts multihost, multistage vulnerability analysis on a network. Datalog is the modeling language that is adopted by MuLVAL for the elements in the analysis. Datalog captures the operating system behavior and the interaction of various components in the system. It can also be represented as a framework for the modeling interaction between software bugs with system and network configurations.

The inputs to MuLVAL are:

- Advisories: Includes reported vulnerabilities and check whether they are already in the machines or not.
- Host Configuration: Includes the configuration of software and their services running in the host.
- Network configuration: Includes configuration of routers and firewalls.
- Principals: Consists of the users in the network.
- Interaction: Consists of how the components in the network interact.
- Policy: Consists of what accesses are to be permitted.

MuLVAL is an attack graph tool which adopts a logic programming approach. This attack graph is constructed by accumulating true factors of the monitored networking system and it terminates efficiently because number of the facts is polynomial in the system.

3.4. Queue Graph

A Queue graph approach is used to remove obstacles towards alert correlation. Queue graph only keeps in memory those alerts which matches with each of known exploit. This correlation is explicitly recorded and the correlation with other alerts is implicitly represented using temporal order between alerts. Linear time complexity and quadratic memory requirement for this approach is independent for the number of alerts received and thus efficiency does not decrease with time.

As a continuation for queue graph, a unified method for correlation, hypothesis and prediction of new alerts is introduced. The knowledge from the queue graph and the facts represented by the correlated alerts are compared and the inconsistency between them implies the attacks missed by IDS. Finally this queue graph is modified to a result graph by removing all its transitive edges and aggregates those alerts in terms of correlation. Thus the resultant graph contains no redundant information and becomes more efficient.

3.5. Attack Countermeasure Tree

Attack countermeasure tree (ACT) is an attack tree paradigm whose structure takes attacks as well as countermeasures into account. Here use greedy and branch bound techniques are used to achieve some goals. In ACT:

- Defense mechanisms can be places at any node of the tree. Not just as the leaf nodes.
- Generation and analysis of attack scenarios and countermeasure scenarios are automated using mincuts.
- Security analyses are performed in an integrated manner.

ACT consists of three different classes: atomic attack events, detection events and mitigation events. ACTs are used to perform fast and efficient computation of optimal set of countermeasures for system based on a non-state space model thus avoids the state space explosion problem. It also shows how some analysis and optimization can be performed without making probability assignments.

3.6. NICE

NICE, Network Intrusion Detection and Countermeasure Selection in Virtual Networks is used to establish a defense in depth intrusion detection framework by incorporating attack graph procedures. NICE does not needed to improve any of the existing intrusion detection methods, instead it employs a reconfigurable virtual networking approach to detect and counter the attempts to compromise virtual machines.

It mainly consists of two steps.

- NICE continuously verify the network using a software agent called NICE-A to capture and analyze cloud traffic.
- If any of the vulnerabilities found, put them in a deep packet inspection mode.

NICE uses an OpenFlow network programming APIs to build a monitor. NICE uses two graphs. First one is called Scenario Attack Graph (SAG) which represents and results of such actions by extending the concept of MuLVAL. Second graph is called Alert Correlation Graph (ACG) used to select appropriate countermeasure from a collection of known countermeasures. Based on the vulnerabilities or attacks machines are placed in different states.

The design of NICE consists of:

- NICE-A: Light weighted software agent used to scan the network for detecting attacks.
- VM profiling: It is used to get precise information about the states, services running, pen ports etc of virtual machines in a cloud.
- Attack analyzer: This is used to construct and update attack graphs, alert correlation and countermeasure selection.
- Network controller: This is a key component to support the programmable networking capability to realize the virtual network configuration feature based on OpenFlow protocol.

IV. CONCLUSION

By analyzing all these detection methods we can understand that there are different types of attacks and for those attacks there are different types of detection and countermeasures are also available. From the above explained six methods we can conclude that NICE is better among all other methods. Because it uses a consolidated approach of all other detection methods. It uses attack graphs in a nice way such that attacks can be predicted and detected.

To improve the performance and efficiency of NICE host based IDS are also needed to incorporate.

V. ACKNOWLEDGEMENT

This work is supported by our guide and my colleagues.

REFERENCES

[1] Chun-Jen Chung,Pankaj Khatkar, Tianyi Xing, Jeongkeun Lee,Dijiang Huang, “NICE:Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems”, IEEE Trans.Dependable and secure computing,vol.10, no.4, pp.198-211.July/August 2013.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A View of Cloud Computing”, ACM Comm., vol. 53, no. 4, pp. 50-58, Apr. 2010

[3] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker, “Detecting Spam Zombies by Monitoring Outgoing Messages,” IEEE Trans. Dependable and Secure Computing, vol. 9, no. 2, pp. 198-210, Apr. 2012.

[4] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, “BotHunter: Detecting Malware Infection through IDS-driven Dialog Correlation,” Proc. 16th USENIX Security Symp. (SS ’07), pp. 12:1-12:16, Aug. 2007.

[5] G. Gu, J. Zhang, and W. Lee, “BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic,” Proc. 15th Ann. Network and Distributed Sytem Security Symp. (NDSS ’08), Feb. 2008.

[6] X. Ou, S. Govindavajhala, and A.W. Appel, “MulVAL: A Logic- Based Network Security Analyzer,” Proc. 14th USENIX Security Symp., pp. 113-128, 2005.

[7] L. Wang, A. Liu, and S. Jajodia, “Using Attack Graphs for Correlating, Hypothesizing, and Predicting Intrusion Alerts,” Computer Comm., vol. 29, no. 15, pp. 2917-2933, Sept. 2006.

[8] A. Roy, D.S. Kim, and K. Trivedi, “Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Countermeasure Trees,” Proc. IEEE Int’l Conf. Dependable Systems Networks (DSN ’12), June 2012..

