International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation And Computing (ICIAC-12th &13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

Review of Intrusion Detection Systems (IDS) and Agents Based IDS

Abhilasha D. Kulkarni*, Prof. R. B. Joshi**
*(Department of Computer Engineering, MMMCOE, Pune University, Pune-52
Email: abhilasha1101@gmail.com)
** (Department of Computer Engineering, MMMCOE, Pune University, Pune-52
Email: ramjoshi@mmcoe.edu.in)

ABSTRACT
Since last decade in network security research area, security of mobile ad hoc networks and computer network is becoming important in every individual's day-to-day life. There are many tools and methods presented by various authors to protect wired and wireless networks from different kinds of security threats. These tools are working for defending the networks from such intrusions and attacks. In this paper our main aim is to present the survey over the concepts of intrusion detection, anomaly detection, and detailed history over the same. In addition to this we are taking the review of different intrusion/anomalies detection methods taxonomy. The goal of this survey paper is heading toward the future direction into agent based security methods for both wireless as well wired networks. Finally the agents in intrusion detection systems are presented with their advantages and disadvantages.

Keywords– Intrusion detection, Security, SNORT, wireless networks, wired networks

I. INTRODUCTION

The approach used is the distributed or the agent based computing approach in which not only the workload will be divided between the individual processors, but also the IDS will be able to obtain an overall knowledge of the network's working condition. Having an overall view of the network will help the IDS to detect the intrusion more accurately and at the same time it can respond to the threats more effectively. In this approach, servers communicate with one another and generate alarm. In order to respond to an attack, sometimes it can be sufficient enough to disconnect a subnet. In this type of system in order to contain a threat, the distributed IDS can order servers, routers or network switches to disconnect a host or a subnet. One of the concerns with this type of system is the extra workload that the IDS will enforce on the network infrastructure. The communication between the different hosts and servers in the network can produce a significant traffic in the network. The distributed approach can increase the workload of the network layers within the hosts or servers and consequently it may slow them down. There are two approaches in implementing an agent-based technology. In the first approach, autonomous distributed agents are used to monitor the system and communicate with the agents in the network. Zhang et al.[46] report implementing a multi-agent based IDS where they have considered four types of agents: Basic agent, Coordination agent, Global Coordination agent, and Interface agents. Each one of these agents performs a different task and has its own subcategories. For example, the basic agent includes: Workstation agents, Network segment agents and Public server agents. These subcategoryagents respectively work on the workstations of the network, as well as, the subnet level and public server level (Mail agent or FTP agent). In this way, the complex system breakdown into much simpler systems and will become easier to manage.

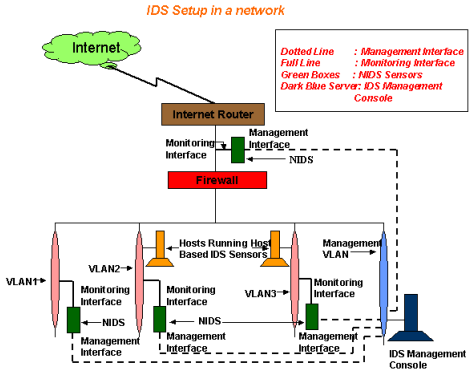
In the second approach, mobile agents are used to travel through the network and collect information or to perform some tasks. Foo et al.[16] report an IDS development work [17] using mobile agents. They use the Mitsubishi's Concordia platform in their work to develop a mobile agent based IDS. Using the mobile agent, the IDS perform both the port scanning and it checks the integrity on the critical files of the system. The proposed agent based IDS will raise the alarm if it detects any alteration on the critical files of the system. Mobile agents can be sent to other systems to monitor health of the target system and to collect information. Luo et al.[18] introduce a new Mobile Agent Distributed IDS (MADIDS). Authors address number of deficiencies that exist in distributed IDSs: "The overload of data transmission", "The computation bottleneck of the central processing module" and "The delay of network transmission". Paper reports that one of the main goals of the system is to improve the performance of the IDS in regard to speed and network traffic. In a work reported by Ramachandran et al. [19] the idea of neighborhood-watch is implemented for the network security. There are

three different types of agents in three different layers. All the agents are defined in PERL (Practical Extraction and Report Language). In the front line (bottom layer) there is a Cop agent that is a mobile agent. There are different types of Cop agents dependent on their assignments. A Cop agent is responsible for collecting data from various sites and reporting them to its respective detective agent. In this system, each site will store all the important security information about its neighbors. This information includes checksum of critical data files and system binaries, etc. It will also store a list of its neighbors in the neighborhood. There are neighbors (hosts) within each neighborhood (subnet) who can be inspected by the mobile agents called Cops. By voting among themselves, neighbors will decide on the course of action they intend to follow.

II. REVIEW OF INTRUSION DETECTION SYSTEM (IDS)

The Intrusion detection system complements the firewall security in a similar way. The firewall protects an organization from malicious attacks from the Internet and the Intrusion detection system detects if someone tries to break in through the firewall or manages to break in the firewall security and tries to have access on any system in the trusted side and alerts the system administrator in case there is a breach in security [8].

2.1 Architecture of General IDS



2.2 Stateful vs. Stateless

A Stateful server loses all its volatilestate in a crash. It restores the state using a recovery protocol that is based on a dialog with clients, or abort operations that were underway when the crash occurred. Server needs to be aware of client failures in order to reclaim space allocated to record the state of crashed client processes (orphan detection and elimination). With stateless server, the effects of server failure and recovery are almost unnoticeable. A newly reincarnated server can respond to a self-contained request without any difficulty.

2.3 Cost vs. Benefit

Costs related to computer securityare often difficult to assess, in part because accurate metrics have been inherently unrealistic. Of those costs that can be measured, the largest in terms of monetary value typically involve theft of proprietary information or financial fraud. Others that are more difficult to quantify but have resulted in severe loss of use or productivity include viruses and malware, Web server denial-of-service attacks, abuse of access privileges, and equipment vandalism or outright theft. We see the results of surveys of organizations providing estimates as to breach incidents (supposedly affecting 90% of large corporations and government agencies in 2002, according to the Computer Security Institute), security expenditures (projected at more than $3 billion in 2004 by International Data Corp.), and malicious code (worldwide loss estimates by Computer Economics exceeded $13 billion in 2001 alone), and so on, with numbers continuing to reflect dramatic growth each year. However, lacking any way to translate such statistics into expenditures and losses per organization, per computer, or per user, the true impact of these figures remains uncertain [7].

2.4 False Positives and False Negatives:

A false positive occurs when the scanning reports finding a virus when there is in fact no virus present. The chances of this occurring depend on the type of virus checking being done, and also on the general quality of the software. Scanners that use virus definition files don't report false positives very often; software that looks for "virus-like behavior" will report false positives constantly, because they are only guessing at what "might be" viruses (such as updates to program files, etc., which can be quite legitimate in some cases.)

2.5 Detection vs. Prevention

Intrusion Prevention Systems (IPS),also known as Intrusion Detection and Prevention Systems (IDPS), are network security appliances that monitor network and/or system activities for malicious activity. The main functions of intrusion prevention systems are to identify malicious activity, log information about said activity, attempt to block/stop activity, and report activity. Intrusion prevention systems are considered extensions of intrusion detection systems because they both monitor network traffic and/or system activities for malicious activity. The main differences are, unlike intrusion detection systems, intrusion prevention systems are placed in-line and are able to actively prevent/block intrusions that are detected. More specifically, IPS can take such actions as sending an alarm, dropping the malicious packets; resetting the connection and/or blocking the traffic from the

offending IP address [9].

2.6 Signature-Based Detection

This method of detection utilizes signatures, which are attack patterns that are preconfigured and predetermined. A signature-based intrusion prevention system monitors the network traffic for matches to these signatures. Once a match is found the intrusion prevention system takes the appropriate action. Signatures can be exploit-based or vulnerability-based. Exploit-based signatures analyze patterns appearing in exploits being protected against, while vulnerability based signatures analyze vulnerabilities in a program, its execution, and conditions needed to exploit the said vulnerability.

2.7 Statistical Anomaly-based Detection

This method of detection baselines performance of average network traffic conditions and once a baseline is created, the system intermittently samples network traffic, using statistical analysis to compare the sample to the set baseline. If the activity is outside the baseline parameters, the intrusion prevention system takes the appropriate action.

2.8 Stateful Protocol Analysis Detection

This method identifies deviations of protocol states by comparing observed events with "predetermined profiles of generally accepted definitions of benign activity."

III. REVIEW OF DETECTION METHODOLOGIES

The3.1 Misuse-based Intrusion Detectionsystems

Misuse detection IDS modelsfunction is very much the same sense as high-end computer anti-virus applications. That is, misuse detection IDS models analyze the system or network environment and compare the activity against signatures (or patterns) of known intrusive computer and network behavior [10]. These signatures must be updated over time to include the latest attack patterns, much like computer anti-virus applications. Misuse detection has its share of advantages as well:

-If the target deployment is only a few computer systems, then a misuse-based IDS is easy to implement, update and deploy. However, if the scope of deployment is large (many computer systems), the implementation, updating and deployment could be quite a task, which would be a disadvantage. -Misuse-based IDS can be used very quickly. There isn't a need for the IDS to "learn" the network behavior before it can be of use.

-The signature matching also provides fewer false alarms (false positives) than other IDS methods. -If the signatures of attacks used by the misuse detection system are reliable, then attacks that match those signatures are very quickly identified, which makes the determination of corrective measures easier.

-Computer administrators can write their own signatures in accordance with the organizations security policy.

Like anti-virus software, the signatures containing the attack patterns are constantly changing. Good computer and network hackers are well aware of the patterns of known exploits. These patterns can be modified to decrease the chances of raising any red flags. Intrusion detection systems that follow the misuse detection model need to be constant updated to stay a step ahead of the hackers.

a. Advantages of misuse intrusion detection system -Can name attacks -System administrators can write their own signatures

-Easy to implement -Properly implemented, it does not give many false alarms.

b. Disadvantages of misuse intrusion detection system -The signature database tends to get big and clustered after a while. This can slow down the system.

-Cannot completely detect novel attacks -Needs to be updated with new signatures to catch newly discovered attacks

-Unprotected against new attacks during the time it takes to write new signatures.

3.2 Anomaly-Based Intrusion Detection Systems

Anomaly detection uses models of the intended behavior of users and applications, interpreting deviations from the "normal" behavior as problem. Maxion and Tan have expanded this definition: "An anomaly is an event (or object) that differs from some standard or reference event, in excess of some threshold, in accordance with some similarity or distance metric on the event" [12].

a. Advantages of Anomaly intrusion detection -Can easily detect attacks from the inside

-Hard for an intruder to know how he should behave to not raise an alarm since profiles can be on individual users

-Can detect previously unknown attacks-Can use more sophisticated rules

b. Disadvantages of Anomaly intrusion detection -Complex to implement -High rate of false alarms

-Still not satisfying enough in a dynamicenvironment

-Cannot name attacks.

3.3 Specification Based Intrusion Detection System

Specification-based monitoring compares the behavior of objects with their associated security specifications that capture the correct behavior of the objects. The specifications are usually manually crafted based on the security policy, functionalities of the objects, and expected usage. Specification- based detection does not detect intrusions directly –

it detects the effect of the intrusions as run-time violation of the specifications instead. As the specifications are concerned with the correct behavior of objects, specification-based detection does not limit itself to detecting just known attacks. The specification- based detection approach has been successfully applied to monitor security critical programs, applications, and protocols [13]. In particular, specifications for the Address Resolution Protocol (ARP) and the Dynamic Host Configuration Protocol (DHCP) have been used to detect attacks that exploit vulnerabilities in these protocols.

IV. REVIEW OF AGENT BASED IDS

T4.1 Agents In Intrusion Detection System The need for a clear understanding of agents is necessitated due to the fact that the intrusion detection system that we have developed and extended, is layered on top of an agent platform, called Grasshopper , based on the first mobile agent standard MASIF2 (Mobile Agent System Interoperability Facility) of OMG (Object Management Group). The term agent or software agent is usually deciphered well in the artificial intelligence community, where it stands for a program that can behave autonomously to perform a multitude of dynamic tasks based on the logistics that have been programmed into it by a user.

4.2 Agent Classification

Based on the mobility of agents, they can be classified into three main types

• Static Agents: The first is the concept of static agents. Static agents area fragment of code that do not move to different locations, and stay at a constant position throughout its life cycle i.e. they remain at the same logical and physical location from the point of creation to the point when they are destroyed, or the program terminates. MASIF is an interoperability standard that allows agents from different mobile agent platforms to interact with each other.

• Semi-Mobile Agents: Semi-mobile agents, as the name suggests, have some mobility. They are in fact an inherent type of mobile agents, which are created at one logical or physical location, but are moved to another location for its functional life cycle.

• Mobile Agents: Mobile agents are a fragment of code, which can move around, hopping from machine to machine during its life cycle depending on the runtime task allocated to it. Mobile agents are based on a terminology, well known in literature as mobile code. The term mobile code can be defined as the capability to change the binding between the pieces of code, and the location where they are executed. The scope of the advantages or disadvantages of using any of the above mentioned agent types can vary based on the functionality of the agent based system that is being deployed. If latency is a big issue in the system, one should opt for static and/or semi-mobile agents. This is because the greater the mobility of an agent, the higher the latency introduced into the system caused by the time required to create it at a new location and to transfer the runtime state of the agent. If the host where the agent runs is very fragile or more prone to destruction or tampering, it would be best to use a mobile agent rather than a static agent, as it is easier for mobile agents to find a new location to run at than static agents.

4.3 Agent Design Paradigm

It is important to understand thedesign paradigm to be used while using mobile agents. Some of the cost benefits that have to be measured include latency, memory access, partial failures and concurrency. The following are some of the designparadigms that can be used with mobile agents:

• Client-Server model: In a client- server model, a client at location A, asks a server at location B to perform a certain task. A does not have the means or methods of processing a given task, but can only make request for work to be done. An example of this is a Java RMI (Remote Method Invocation) call.

• Mobile agent (MA) model In a Mobile agent model, user-x may have some means and methods to process data at location A. There are also data or methods located at another location B that user-x requires for completing the process. In that case, user-x can migrate from location A to location B, carrying with it some of the processed data from A and the know-how to process the data available at B.

In APHIDS, the two design paradigms used are the client-server and the mobile agent model. Hence the readers are referred for a further explanation of the remaining two models namely, Remote Evaluation (REV) and Code on Demand (COD).

4.4 Advantages and Disadvantages of Agents

The use of mobile agents offers wide advantages especially in distributed systems that cannot be overlooked. The major categories of these are summarized as follows:

• Reduction in Network Traffic: As we know, in case of mobile agents, the agents themselves move to data. I.e. we move the agent code to the data rather than moving the data to the agent code. This allows for a dramatic reduction in the amount of bandwidth consumed in the log correlation process (explained in later sections) as data is almost always larger than the few Kb size of agents in general.

•Asynchronous autonomous interaction: An advantage of mobile agents is its ability to asynchronously process information. This is vital in a network where network connections are volatile, such as wireless networks. In such cases, the agent could

migrate to a mobile device to gather data. Even if the connection breaks, the agent could continue processing data on the mobile device and report back whenever the connection is reestablished. This adds to the agent's capability to work in a fault tolerant mode.

• Software Upgrades: Usually in order to update software on multiple hosts, an administrator has to first stop the server functionality, then uninstall the old version of the software, and then reinstall the new version. The entire software system has to be stopped for upgrades. The advantage of mobile agents or agents in general in this situation is that if each component of the upgraded software is managed by an agent, then it is as easy as disabling the old agent and deploying a new agent which has the required functionality. In this way one could avoid bringing down the entire system and instead stop just a single agent- based component.

• Functionality in heterogeneous environments: Most agents today can work in heterogeneous environments. This is due to the fact that these agents are usually written in a language which is portable to multiple platforms, such as java or Perl. Since agents sit on top of an agent framework, they can easily function regardless of if the host runs a version of Linux or Windows operating system. The significant reduction in costs of placing agent frameworks in hosts over the past few years have added to the benefits of running agents. Just like there are advantages to using agents, there are also drawbacks to using agents [14]. The applicability of advantages or disadvantages to using agents is based immensely on the specific user needs or goals that have been put forward. Some of the major drawbacks mentioned by authors include:

• Agent Security The one and only reason that has hindered the wide usage of mobile agents in the real world has been its security constraints. One of the key problems associated with mobile agent security is the malicious host problem i.e. how much trust can be placed on a host where the agent travels to, given that the agent may have valuable highly secured data. This data could be as vital as a person's credit card information in an unencrypted format, or the password to ones bank account. Many have claimed that if the agent is placed in a closed environment then this problem does not exist. But the fact is that this problem still persists in situations when an intruder has overtaken a system in a closed environment without the knowledge of the administrator.

• Lack of Shared Language Even though many tasks have been overtaken by FIPA (The Foundation for Intelligent Physical Agents) to create a standard ACL (Agent communication language) 3, most agent platforms do not adhere to this language. Hence it is hard for agents to communicate with each other when they are based on different platforms.

• Required Agent Platform Any piece of agent code available today needs to run on an agent platform that contributes to the control and deployment of agents. For example, our APHIDS system has to use the Grasshopper agent platform to execute its tasks. Similarly, to run java applets, the system has to have a java runtime environment available. The dependence of mobile agents on an agent platform is an extra requirement that has to be made, without which they cannot function. The problem is further compounded by the fact that not all agent platforms follow a given set of rules and procedures thus hindering interoperability issues even with the existence of standards such as MASIF (explained previously).

• Denial of Service Any piece of code that is written by a programmer can have flaws. For example a user could perform a logical error in his code by making the *fork ()* system call in a while loop. The presence of such snippets of code in a mobile agent code that travels to a location and executes it, could allow them to launch denial of service attack against the host where they reside by hogging all the available system resources [15].

V. CONCLUSION

Considering the surveyed literature, it is clear that in order to be able to secure a network against the novel at- tacks, the anomaly based intrusion detection is the best way out. However, due to its immaturity there are still problems with respect to its reliability. These problems will lead to high false positives in any anomaly- based IDS. In order to solve this problem, usually a hybrid approach is used. In the hybrid approach, the signature-based approach is used together with the anomaly-based approach. In this way, the second approach is mostly used for the novel tactics while the accuracy of the first approach (signature based approach) will provide a reliable detection for the known attacks. Specification-based approach is only good when system specifications and de-tails are known and applying limitations on the user is acceptable. The generic definition of the normal behavior and the anomaly behavior in the system are presented in this paper. The intension for introducing these generic definitions was to help researchers to converge on the definition of the normal behavior of the network. In network-based IDS, agent based systems play an essential role. In such systems a distributed processing architecture is a must and system has to collect information from different components within the network. Implementing such architecture, one should avoid increasing the network traffic. Large volume of data and non-deterministic normal behavior of the network are two major challenges in IDS de-sign. As

the volume of data using the header of the packets is already very large, using information in the payload will make the process even slower. However, there are works reported by some researchers in this area that show good progress in using packets payload for the analysis. The intrusion detection products were analyzed with respect to the software or appliance based production and the benefits of either of the designs were discussed. Building hardware appliances can be more difficult for companies with lower development budget. However, appliance based IDSs are more appreciated in the market. From the consumer point of view, appliance based IDS is easier to install and to maintain. In manufacturer's view, appliance based IDS is a more secure design to manufacture but as the same time more expensive to produce. Another aspect of the IDS design is the issue of the missed attacks. If some attacks are not detected by the IDS, there are no means to notice them. This is especially the case with the novel attacks.resources [15].

VI. FUTURE WORK

As for the future work, intension is to produce IDS capable of anomaly and signature based intrusion detection. There are two options in front of us, i.e. host based or network based IDS. The host based IDS can be easier to implement, though the network based IDS needs more time and effort for its implementation and design. In return, the network based IDS will provide a more reliable and more accurate IDS. The network IDS needs to have environment awareness. Thus, the network based IDS need special sensors for its work. Agent based technology is one of the essential blocks in this distributed architecture design methodology.

The selected approach for our future work is the network based software product. However, the host based approach will be considered as well. The project timeframe and the budget are main issues with regard to this decision. Nevertheless, accepting the expenses, it is always possible to convert a software based IDS to the appliance version of it. From the theoretical point of view, it is intended to improve the accuracy of the anomaly based intrusion detection. One way to do so is to use the payload of the packets. Therefore, it is necessary to envisage a method either to reduce the size of the data or to process the data more quickly. The main idea is to find a method to handle high volume of data with less information loss. For the same reason, features should be evaluated with respect to their information value. In this way, every feature will be associated with a coefficient of importance that determines its overall effectiveness in comparison to the other features. Efficient algorithms and programs can provide a great help for this purpose.

REFERENCES

[1] Defeng Wang, Yeung, D.S., and Tsang, E.C., "Weighted Mahalanobis Distance Kernels for Support V ector Machines", IEEE Transactions on Neural Networks, Vol. 18, No. 5, Pp. 1453-1462, 2007.

[2] Glenn M. Fung and O. L. Mangasarian, "Multicategory Proximal Support V ector Machine Classifiers", Springer Science and Business Media, Machine Learning, 59, 77–97, 2005.

[3] Guang-Bin Huang, Dian Hui Wang and Y uan Lan, "Extreme learning machines: a survey", Published: 25 May 2011_ Springer-Verlag, 2011.

[4] Hyeran Byun and Seong-Whan Lee, "Applications of Support Vector Machines for Pattern Recognition: A Survey", Springer-Verlag Berlin Heidelberg,2002

[5] G. Jacob Victor, Dr. M Sreenivasa Rao and Dr. V. CH. Venkaiah, "Intrusion Detection Systems Analysis and Containment of False Positives Alerts", International Journal of Computer Applications (0975 – 8887), V olume 5– No.8, August 2010.

[6] Muhammad Awais Shibli, Sead Muftic. Intrusion Detection and Prevention System using Secure Mobile Agents, IEEE International Conference on Security & Cryptography (2008)

[7] David Wagner, Paolo Soto. Mimicry Attacks on Host Based Intrusion Detection Systems, 9th ACM Conference on Computer and Communications Security (2002).

[8] Harley Kozushko. Intrusion Detection: Host-Based and Network- Based Intrusion Detection Systems, (2003).

[9] Lin Tan, Timothy Sherwood. A High Throughput String Matching Architecture for Intrusion Detection and Prevention, Proceedings of the 32nd Annual International Symposium on Computer Architecture (ISCA 2005).

[10] S. Mrdovic, E. Zajko. Secured Intrusion Detection System Infrastructure, University of Sarajevo/Faculty of Electrical Engineering, Sarajevo, Bosnia and Herzegovina (ICAT 2005).

[11] Yeubin Bai, Hidetsune Kobayashi. Intrusion Detection Systems: technology and Development, 17th International Conference of Advanced Information Networking and Applications, (AINA 2003).

[12] Sang-Jun Han and Sung-Bae Cho. Combining Multiple Host-Based Detectors Using Decision Tree, Australian Joint Artificial Intelligence Conference, (AUSAI 2003).

[13] Host Intrusion Prevention Systems and Beyond, SANS Institute (2008).

[14] Intrusion Detection and Prevention In-sourced or Out-sourced, SANS Institute (2008).

[15] Mario Guimaraes, Meg Murray. Overview of Intrusion Detection and Intrusion Prevention, Information security curriculum development Conference by ACM (2008).

[16] Simon Y. Foo and M. Arradondo, "Mobile agents for computer intrusion detection," in Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory, pp. 517–521. IEEE, IEEE, 2004

[17] Mitsubushi Corporation. "Concordia mobile agent development kit,". Software

[18] G.Luo,X.L.Lu,J.Li,andJ. Zhang, "Madids: A novel distributed ids based on mobile agent,"ACM SIGOPS Operating Systems Review, vol. 37, pp. 46–53, Jan. 2003

[19] Website of the Honeynet Project. "Honeynet project," http://www.honeynet.org/