A R DIGITECH
International Journal Of Engineering, Education And Technology (ARDIJEET)
www.ardigitech.in ISSN 2320-883X,VOLUME 09 ISSUE 04 31/11/2021

Agent-Based Blocking and Response, Intrusion Detection using Signature

[1]*Arvind Panwar*
[1]*Department of Computer Science & Engineering RDEC,Ghaziabad*
panwar.arvind01@gmail.com

Abstract: Developing a reliable system for detecting intrusions is a challenging task that does not have a simple or quick solution. We contend, however, that mobile agent technology significantly advances the cause of an IDS's optimal behaviour. This article explores several approaches to One possible solution to the issue of intrusion detection and response might be the use of mobile agents. The study examines the advantages of software agents in general as well as those that arise from mobility.Once we've covered these advantages, we'll go over some ways mobile agent technology can fix the problems with existing intrusion detection systems (IDS) and the security risks that come with them. After an intrusion has been identified, we also examine several innovative methods for automatic responses.

Keywords: Intrusion Detection, Mobile Agents, and Computer Security

Introduction

Originally designed as a kind of expert system, intrusion detection systems (IDSs) monitor user account activity patterns and alert the system administrator to any suspicious occurrences. Despite James Anderson's 1980 [1] proposal, the idea didn't take off until 1987 [2] when Dorothy Denning ut her groundbreaking intrusion detection methodology into publication [9]. A monolithic design was used in early intrusion detection system implementations [21, 27, 28]. This meant that data acquired from a single host was analysed centrally, either at or around the moment of collection. Designers of intrusion detection systems realised that keeping tabs on a single host's activities wouldn't catch assaults that included several hosts, so they came up with network-based IDSs. These use a traffic model to infer abuses or abnormalities from low-level packets that move across hosts [13]. One way to define network-based intrusion detection systems is as a shift from a detection focus on hosts to one on the network as a whole. Many issues with integrity and performance, as well as those related to audit trails, may be addressed by adopting a network-centric strategy [25].

The method utilised to detect an incursion is another way in which IDSs may be classified. Disruptions to a system's or user's usual pattern of operation might serve as indicators of an intrusion. Characteristics of inputted keystrokes, command profiles, and use time of day are all examples of possible behaviour. If the behaviour goes over a certain acceptable limit, a notice will be sent. It is also possible to identify an incursion if the observed behaviour closely matches a previously identified pattern. A rule-based method is usually used in this more direct kind of discrimination, whereby the rules codify patterns of intrusion called signatures. Notifications are triggered when an event or series of events matches a signature.

A two-component design was used by the initial generation of intrusion detection systems. The host's audit logs and internal interfaces or the connected networks' packet monitoring systems are the sources of data used in the gathering process. A centralised analysis method uses one or more detection

techniques to process such information. This design works well for smaller collections of monitored hosts, but it can't scale to accommodate bigger collections since all the analysis is done in one place. Generations that followedOne way in which intrusion detection systems (IDSs) tackle scalability is by include intermediary components that aggregate and preprocess data collected during collection for use in analysis [6].

A hierarchical design, like the one shown in Figure, is followed by almost all modern commercial IDSs.

1. Data is collected at nodes that are part of a network or at hosts themselves. In order to aggregate data from various leaf nodes, event information is sent to internal nodes. On the way to the root node, more aggregation, abstraction, and reduction of data might take place at higher internal nodes. An evaluation and reaction mechanism for attack scenarios is housed in the root, which is a command and control system.

It is common practice for the root to report to an operator console, allowing administrators to manually evaluate status and send commands.

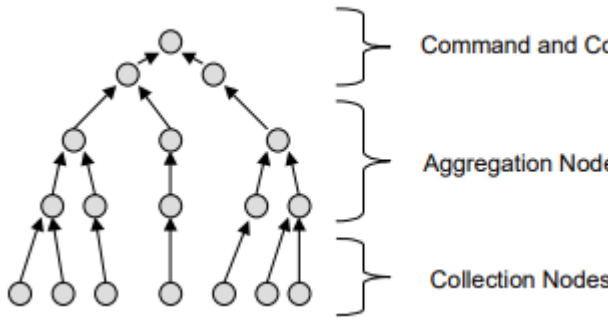

Figure 1: Hierarchical IDS Architecture

Hierarchical architectures often lead to effective communication because control trickles down from higher up the hierarchy and more sophisticated information filters up. Due to the tight binding, the design is fairly stiff, but it is ideal for building scalable distributed IDSs with central points of management in terms of both practicality and the ever-changing nature of communication channels. Although there is a loose hierarchy among IDS components, there is a general trend towards one. Any kind of component may communicate with any other kind of component; one-to-one or master-slave interactions are not necessary. An aggregation node and the command and control node may be immediately notified of a significant event by a collecting unit, for instance, to enhance responsiveness and notification. In addition, when several administrations are in charge of different parts of a business network or separate networks altogether, peer links between command and control nodes are essential [11].

Cooperating Security Managers [31] is one intrusion detection system (IDS) architecture that employs a network topology to centralise the gathering, aggregation, and command and control processes on each monitored system, allowing data to move freely between any two nodes. The security manager at the system where the incident happened will notify the system manager of the system where the connection originated if any major events happen at their system. The system manager is obligated to report to the next system manager in the chain when the connecting system is an intermediary node in the communication chain. At their most extreme, network structures—where every node communicates with every other node—tend to be communications inefficient due to the possibility for unrestricted flow of information. But their functional flexibility makes up for this inefficiency.

Current IDS Shortcomings

Modern intrusion detection systems aren't foolproof . Some problems are intrinsic to the design of IDSs, but developers are working to fix them by making current solutions better. Here are some of the most typical problems:

• **Inefficient:** intrusion detection systems are often needed to assess events as they happen. When dealing with the massive amounts of events that modern networks often experience, this criterion becomes more challenging to fulfil.

As a result, intrusion detection systems that are based on hosts might cause a system to run slower, while those that are based on networks can cause network traffic to drop packages that they are unable to handle due to a lack of time.

• **A Large Amount of False Positives:** The majority of intrusion detection systems (IDS) identify assaults throughout an organisation by examining data from a single host, application, or network interface in many places. Attack identification isn't foolproof, and the number of false warnings is large. Reduce false alerts by lowering thresholds, but increase the amount of assaults that pass through unnoticed as false negatives. The main challenge that intrusion detection system (IDS) manufacturers are now facing is enhancing the system's capacity to correctly identify threats.

• **Time-Consuming Upkeep:** Intruder detection system setup and maintenance often calls for expert-level understanding and a lot of muscle. The conventional wisdom is that expert system shells, which encode and match signatures according to rule sets, are the best way to identify abuse. The intricacies of the expert system and its language for expressing rule sets are involved in upgrading rule sets, and it may only be possible to indirectly specify the sequential interrelationships between occurrences. Adding a statistical measure, which is usually used to identify odd deviations in behaviour, may also need similar considerations.

Intrusion detection systems have historically been developed with a focus on a particular environment, making them inflexible when applied to different settings that share comparable regulations and concerns.

Another issue is that the detection technique could be hard to adjust to new use patterns. Another common issue with intrusion detection system implementations is the need to modify detection methods for each individual system and then update those mechanisms with newer, better detection approaches. If you want any changes or additions to take effect, you may have to restart the IDS.

• **Direct Attack Vulnerability:** Many intrusion detection systems are attackable due to their dependence on hierarchical arrangements for components. By compromising an internal node, an attacker may disable an IDS control branch or potentially disable the whole system by destroying the root command and control node.

Platforms housing such crucial components are usually designed to withstand direct assault.

However, existing implementations do not include additional survival approaches like mobility, redundancy, dynamic recovery, etc.

• **Being Easily Tricked:** By simulating the hosts' protocol stacks, network-based intrusion detection systems (IDS) can assess network traffic. Because the IDS and the target host have differing interpretations of specially tailored packets, attackers may take advantage of this disparity. In order to do this, one may modify fragmentation, sequence number, and

packet flags, among other things [26]. While the IDS is either unaware of the assault or is mislead to believe that the target fought back, the attacker gains access to the target.

Traditional intrusion detection systems have only been able to identify assaults, which limits their capacity to respond. Although detection is helpful, system administrators aren't always quick to assess IDS data and respond accordingly. Because of this, an attacker has a little window of time to do their thing before the administrator can stop them. In order to drastically shorten the amount of time that attackers have to further entrench themselves in a network, some intrusion detection systems are starting to include automatic reaction capabilities. Nevertheless, their capacity to respond dynamically to an assault is restricted.

• **Lack of a Standardised Approach to Construction:** Without a standard approach, constructing an IDS from existing components may be an expensive ordeal. On top of these problems, IDSs have to overcome new challenges all the time. The following problems have emerged as recent roadblocks:

• Complete Security enhancements to communication protocols have made encryption a viable option, allowing for encrypt data in transit from beginning to finish is becoming more popular. Not only does encrypted material prevent eavesdropping, but it also prevents a network-based intrusion detection system (IDS) from peering into packets and analysing their contents for intrusions.

• **Communications at High Speed:** Increases in the volume of data sent and received have an immediate impact on the processing power required to decipher packet contents, which in turn increases the likelihood of packet loss. The increasing use of switched communications over broadcast further complicates the task of a network-based intrusion detection system (IDS) to keep tabs on various communication channels.

• **Variety of assaults:** Intrusion Detection Systems (IDSs) need to be upgraded to locate newly developed assaults. Even though new attacks are introduced all the time, old ones are seldom removed. The detection system usually needs additional processing time for larger assault coverage.

• Technological Constraints: Building a programme that can reliably identify malicious code inside any given programme or protocol is currently not feasible. There is a risk of diminishing returns for intrusion detection systems when services change and new ones are added. This means that more resources will be required over time for smaller improvements in efficacy.

Conclusion

The previous sections have shown several potential applications of mobile agents to intrusion detection, which might lead to the development of new designs that are more efficient, scalable, and resilient. Even though it's far from ideal, mobile agent technology does a great job of helping an IDS achieve the desired behaviour. In addition to not only do parts of the detection equation become better, but the reaction side gets much better—maybe even more so. There will likely not be a dramatic shift to this model very soon as modern IDSs do not need mobile agent technology. On the other hand, the technology is well-suited for a more measured approach. Mobile agent technology may be able to establish a footing and then grow due to the benefits mentioned, especially when it comes to reacting to an incursion.

References

[1] James P Anderson, "Computer Security Threat Monitoring and Surveillance," Technical Report, James P. Anderson Co., Fort Washington, PA, April 1980.

[2] Midori Asaka, ShunjiOkazawa, Atsushi Taguchi, and Shigeki Goto, "A Method of Tracing Intruders by Use of Mobile Agents," INET'99 Conference, June 1999.

[3] Jai Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, E. H. Spafford, and Diego Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents," Department of Computer Sciences, Purdue University; Coast TR 98-05, 1998.

[4] KarimaBoudaoud, HoudaLabiod, "MA-NID: A Multi-Agent System for Network Intrusion Detection," Eighth International Conference on Intelligent Systems, June 1999.

[5] GiacomoCabri, LetiziaLeonardi, Franco Zambonelli, "The Impact of the Coordination Model in the Design of Mobile Agent Applications," Twenty-second Computer Software and Applications Conference (COMPSAC), August 1998.

[6] S. Staniford-Chen, et al., "GrIDS – A Graph Based Intrusion Detection System for Large Networks," Nineteenth National Computer Security Conference, pp.361-370, October 1996.

[7] David Chess, Benjamin Grosof, Colin Harrison, David Levine, Colin Parris, Gene Tsudik, "Itinerant Agents for Mobile Computing," IEEE Personal Communications, 2(5), pp.34-49, October 1995.

[8] Michael Conner, Chirag Patel, and Mike Little, "Genetic Algorithm/Artificial Life Evolution of Security Vulnerability Agents," Army Research Laboratory Federal Laboratory Third Annual Symposium on Advanced Telecommunications & Information Distribution Research Program (ATIRP), February 1999.

[9] Dorothy E. Denning, "An Intrusion Detection Model," IEEE Transactions on Software Engineering, 13(2), pp.222-232, February 1987.

[10] Serge Fenet and SalimaHassas, "A Distributed Intrusion Detection and Response System Based on Mobile Autonomous Agents Using Social Insects Communication Paradigm," First International Workshop on Security of Mobile Multiagent Systems, Autonomous Agents Conference, May 2001.

[11] Deborah Frincke, Don Tobin, Jesse McConnell, Jamie Marconi, and Dean Polla, "A Framework for Cooperative Intrusion Detection," Twenty-first National Information Systems Security Conference, pp.361- 373, October 1998.

[12] Guy Helmer, Johnny S. K. Wong, VasantHonavar, and Les Miller, "Intelligent Agents for Intrusion Detection," IEEE Information Technology Conference, pp.121-124, September 1998.

[13] L.ToddHeberlein, G.V. Dias, K. N. Levitt, B. Mukherjee, J. Wood, and D. Wolber, "A Network Security Monitor," Symposium on Research in Security and Privacy, pp.296-304, May 1990.

[14] Stuart Jacobs, Dave Dumas, William Booth, and Mike Little, "Security Architecture for Intelligent Agent Based Vulnerability Analysis," Third Annual Fedlab Symposium on Advanced Telecommunications/Information Distribution Research Program, pp.447-451, February 1999.

[15] Wayne Jansen and Tom Karygiannis, "Mobile Agents and Security," NIST Special Publication 800-19, September 1999.

[16] Wayne Jansen and Tom Karygiannis, "Privilege Management of Mobile Agents," Twenty-third National Information Systems Security Conference, pp.362-370, October 2000.

[17] Günter Karjoth, N. Asokan, and CekiGülcü, "Protecting the Computation Results of Free-Roaming Agents," Second International Workshop on Mobile Agents, Stuttgart, Germany, September 1998.

[18] Danny Lange and Mitsuru Oshima, Programming and Deploying Java Mobile Agents with Aglets, ISBN:0-201-32582-9, Addison-Wesley, 1998.

[19] Wenke Lee, Sal Stolfo, and KuiMok, "A Data Mining Framework for Building Intrusion Detection Models," IEEE Symposium on Security and Privacy, pp. 120-132, May 1999.

[20] Dharamveer, Samsher, Singh DB, Singh AK,Kumar N.Solar Distiller Unit Loaded with Nanofluid-A Short Review. 2019;241-247. Lecture Notes in Mechanical Engineering, Advances in Interdisciplinary Engineering Springer Singapore. https://doi.org/10.1007/978-981-13-6577-5_24.

[21] Dharamveer, Samsher**.** Comparative analyses energy matrices and enviro-economics for active and passive solar still. materialstoday:proceedings. 2020.https://doi.org/10.1016/j.matpr.2020.10.001.

[22] Dharamveer, SamsherKumar A. Analytical study of N^{th} identical photovoltaic thermal (PVT) compound parabolic concentrator (CPC) active double slope solar distiller with helical coiled heat exchanger using CuO Nanoparticles. Desalination and water treatment.2021;233:30-51.https://doi.org/10.5004/dwt.2021.27526

[23] Dharamveer,Samsher, Kumar A. Performance analysis of N-identical PVT-CPC collectors an active single slope solar distiller with a helically coiled heat exchanger using CuO nanoparticles. Water supply. 2021.https://doi.org/10.2166/ws.2021.348