Distributed Intelligent Systems for Network Security Control

Mohamed Shili[1], Hamza Gharsellaoui[1,2,3] and Dalel Kanzari[4]
[1]*LISI Laboratory, National Institute of Applied Sciences and Technology (INSAT), Carthage University, Tunis, Tunisia*
[2]*National Engineering School of Carthage (ENIC), Carthage University, Tunis, Tunisia*
[3]*Al Jouf College of Technology, TVTC, Al Jouf, K.S.A.*
[4]*Institute of Applied Sciences and Technology of Sousse (ISSATso), Sousse University, Sousse, Tunisia*

Keywords: Network Security, Intelligent Agent, Misuse Systems, Attack Propagation, Intrusion Detection.

Abstract: The great number of heterogeneous interconnected operating systems gives greater access to intruders and makes it easier for malicious users to break systems security policy. Also, a single security control agent is insufficient to monitor multiple interconnected hosts and to protect distributed operating systems from hostile uses. This paper shows the ability of distributed security controller's agents to correlate data stream from heterogeneous hosts and to trace abnormal behavior in order to protect network security. An experimental study is done to improve our proposed approach.

1 INTRODUCTION

The point of network security is to oversee the normal actives of its components and to prevent any attempt to break in and disturb their functionality. In order to do that, we need to compare different types of security systems as a base for our approach. When we talk about security we need first to identify the security parameters, intrusion.com gave us the following formula:
Security = visibility + control, but this formula is incomplete, our approach is based on this idea: Security = visibility + control + "self awareness", Speaking of network security self awareness is more like testing the limits of artificial intelligence and giving the security a "Soul". In this sense, we need to identify the potential parameters in which way we give the security system a developing character to enforce high protection levels against external threats a number of intelligent tools which were based on machine learning; like intrusion detection systems based on autonomous agent are developed to survey the network activity and to signal all abnormal deviation the aim of these agents is to detect data signatures that threaten the system's behavior. In this paper, we propose two kinds of intrusion detection agents to protect network components the local agent controller that detects the malicious data signature hidden in directly connected systems based on a local base of known attacks and a central agent controller that collects data from distributed local agent's controller to detect the presence of a propagated network threat.

Section 2 presents an overview on security systems. Section 3 presents our original proposed approach and Section 4 presents the experimentation and comparison. Finally, Section 5 concluded this paper.

2 SECURITY SYSTEMS REVIEW

Operating systems are compromised from the malicious exploit of their services and the intent to violate the security policy (Zimmermann., 2003), (Prigent., 2003), (Ludovic., 2003) and (Michel., 2001). Therefore, it is required to use mechanisms that prevent the security administrator from improper use of the system like the intrusion detection system (Spafford., 2000). The aim of intrusion detection system is the find malicious behaviors inside the network and to automate the decision making process by keeping the continuity of the normal network components tasks. In this context, several methods are used to detect intrusion and to correlate alerts in order to find propagated network threat.

2.1 Mobile Autonomous Agents

Selker (Selker., 1994) and Morreale (Moreale, 1998)

Shili, M., Gharsellaoui, H. and Kanzari, D.
Distributed Intelligent Systems for Network Security Control.
DOI: 10.5220/0006009203480352
In *Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016) - Volume 1: ICSOFT-EA*, pages 348-352
ISBN: 978-989-758-194-6

define an agent as a software component that performs one or more communication tasks by acting in a pre-set manner to reach predefined fixed goals. Besides, Green (Green, 1997) defines mobile agent as a computational entity which acts on behalf of other entities in an autonomous fashion, performs its activities with some level of pro-activity and reactivity, exhibits some degree of the key attributes of learning, co-operation and mobility. Jaisankar et al., (Jaisankar., 2009) showed that an autonomous agent has the capacity to detect unknown attacks without any prerequisite knowledge about specific attacks by surveying the anomalous user activity. Mobile Agents require code source development and resource deployment for each agent. Moreover, the existence of noisy and data disruption can influence the agent to transfer wrong information.

2.2 Genetic Algorithms

Hoque et al. (Sazzadul., 2012) discuss the capacity of genetic algorithms to detect various types of network intrusions. In this context, genetic algorithms are used to find optimal solutions for network security. Potential solutions detected are encoded as sequences of bits, characters or numbers. The unit of encoding called a gene, and the encoded sequence is called a chromosome. The genetic algorithm begins with a set (population) of these chromosomes and an evaluation function that measures the fitness of each chromosome, the goodness of the problem solution represented by the chromosome. It uses reproduction and mutation to create new solutions, which are then evaluated. The selection of chromosomes for survival and recombination/evaluation sequence is iterated many times and if the problem is constructed, the strong solution gradually emerges. The issue in using genetic algorithms is that they require a fine definition of the problem with predefined complete variables.

2.3 Data Mining

Nguyen and Choi, (Nguyen., 2008) proved the utility of data mining to detect real-time network attacks by identifying suspicious patterns in row data. In this context, a set of classified algorithms using knowledge dataset allow to extract knowledge from voluminous quantity of heterogeneous data. This technique permits to treat a several system audit-files issued from distant systems, to extract information about current traffics and to check the presence of known correlated attack signatures. Data mining requires an enormous updated signature base to detect developed attacks.

2.4 Expert System

Anderson et al. (Anderson, 1995) use expert rules to characterize a known intrusive activity represented in activity logs. The expert system is based on a set of rules that encode the knowledge of a human expert. These rules reflect the partially ordered sequence of actions that comprise the intrusion scenario. Some rules may be applicable to more than one intrusion scenario. Expert system permits the incorporation of an extensive amount of human experience into a computer application to identify activities that match the defined characteristics of attacks. Unfortunately, expert systems require frequent updates to remain reliable.

2.5 Neural Networks

Ning (Ning., 2002) and Ghosh et al. (Ghosh., 2000), show the ability of a neural network that detects illegitimate network used through monitoring unusual user activity. The neural network is used to learn the user normal activity and to alert the abnormal user behavior. Despite the expert systems, which can provide the user with a definitive answer, if the characteristics which are reviewed exactly match those which have been coded in the rule base, the neural network will conduct an analysis of the information and provides a probability estimate that the data matches the characteristics which has been trained to recognize. The application of neural networks for intrusion detection has been investigated by a number of researchers to perform anomaly and misuse detection (Cuppens., 2002), (Ning., 2002) and (Ning., 2001). Most of these techniques require an accurate definition of attack signatures to establish a correlation between alerts. Also with a noisy environment and data disruption they cant give correct results. This inconvenient can be solved by unsupervised learning machine tools like neural network which has a great ability to treat noisy and incomplete unknown data and to produce accurate results by means of learning experiences. In our approach, we will use distributed intelligent agents based on neural network technique that collaborate to detect propagated attacks in network.

3 DISTRIBUTED INTELLIGENT AGENT FOR NETWORK SECURITY CONTROL

We propose a set of distributed controllers agents that survey the tasks of interconnected systems in order to detect the presence of network threats. The system is presented by a set of connected nodes. Each node is surveyed by a local agent controller. Once an attack is detected in a node, an alert describing the attack is diffused from the local agent controller to the centralized agent monitor. The latter receives alerts and shows the network components ”states”. The supervising process is divided in two parts; the local control at each sub network and the central control that collects important alert from the local controllers. The system control process can be described as follow:

- Capture of data process: this process has the role of a sniffer which collects data packets from the network interface. For the security reasons, this interface would be invisible to the users of network, to protect it from misuse.
- Storage of data process: it converts and records the collected data packets into specific data structures.
- Analysis of data process: this process has a delicate task to support the responders engines to make decision. It studies the incoming data to the network to detect the occurrence of distributed attack.
- It monitors activity in the network and compares them against known patterns of intrusive or hostile activities. If the result of the analysis is positives, then data packets are classified as suspect and can be added to the attacks base.
- Graph generation process: it draws a scheme to show the different states of network nodes. Altered nodes are filled with common color to indicate the propagation of a specific attack in the network.

4 DISTRIBUTED SECURITY SYSTEMS MODEL

Our purpose is to correlate distributed events to show the state of different network components in order to take the right decision in case of threats. The Figure 1 models distributed controllers agent in network, where each controller is presented by a node. Each node is surveyed by intrusion detection agent. Once an attack is detected in a node an alert describing the attack is diffused from this agent to the centralized supervising agent. The latter receives alerts and applies correlation algorithms to recognize correlated intrusion attacks. The role of agent is to handle incoming data and to infer significant patterns to take the right decision. Our system operates in two stages: first, it applies a set of local agent controller, and then uses a centralized coaching agent to combine the detectors outputs. Each local detector examines part of the network, extracts normal and attack signature from input patterns detects, and then throws the results to the monitor detector. The last merges then the incoming results, eliminates overlapping detections and extracts the relationships between input patterns. The critical role of the local intrusion detection agent is to extract knowledge about fundamental information of attacks from noisy and heterogeneous input data. The critical role of the centralized intrusion detection agents is to correlate between input distributed events to extract the relationship between them, to detect the networks components ”states”, and then to take the right decision.

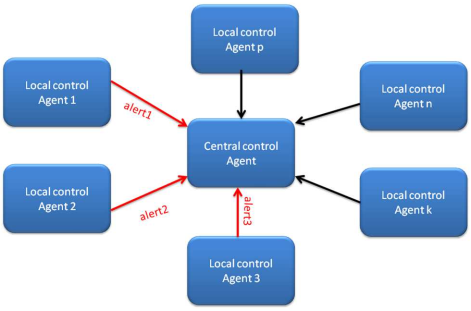


Figure 1: Distributed Control Agents.

5 EXPERIMENTATION RESULTS

The aim of the experimentation is to evaluate the role of the local control agent and the central control agent to protect network from malicious uses.

5.1 The Local Control Agent (LCA) Function

As shown in Figure 2, the Local Control Agent (LCA); based on machine learning technique, studies in real-time the input patterns and then extracts intrinsic features that classify data into two categories:

- Normal data: regular data,
- Suspicious data: attack signature,
- The LCA while using a classification algorithm would:
  - rely on different predefined and permanently updated scenarios stored in a database,
  - filter newly incoming data and ultimately,
  - Report any suspicious data to the Central Control Agent (CCA).

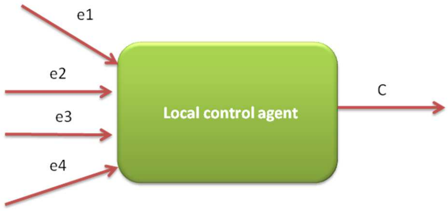


Figure 2: Local Control Agent Model.

Figure 2 presents the following notations:

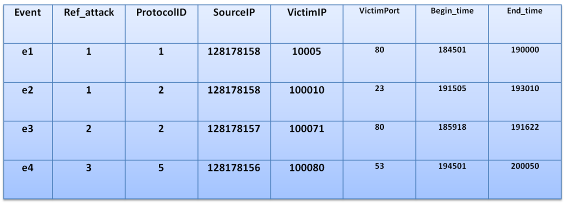
- e1, e2, e3, and e4: are input events about system activities,
- C: data classification in normal or suspicious class.

5.2 The Central Control Agent (CCA) Function

The Central Control Agent (CCA) is based on machine learning paradigm; it gives the final decision about the set of input data from local controllers. It specifies the state of the network, three observations are then possible:

- The presence of a propagated attack,
- The presence of independents attacks,
- The absence of any attack,

The CCA uses an algorithm of data extraction that allows identifying similar or varied suspicious data from different LCA. So this central controller correlates input data from different network nodes and takes the decision about the presence of propagated attacks in the network, and so to send directive responses to the local controllers nodes to either remove an unnecessary local response (firewall filtering rule), or to add a response (firewall filtering rule along an alternate attack path). It is expected to collaborate with the others network components; example the domains network management facilities, to select the optimal points in the network to block harmful connections in the presence of serious attacks. The CCA generates as output a graphical representation of the distributed network nodes. The affected nodes are marked by a distinguished color. These nodes are connected by a set of edges. The nodes and edges are described by a set of attributes. Nodes attribute is made up of pairs (Clef: Designation) of values that represents the address and the name of the network components. The edges are numbered to mark the order of attack propagation. When an attack is presented, the central controller constructs an illustrative oriented schema representing the spread of attacks among network from source to target nodes. This schema is based on input information about distributed network nodes as illustrated by Figure 3. So the sequence off input events e1, e2, e3,e4 ,.. Forwarded to the central controller is translated into sequence .

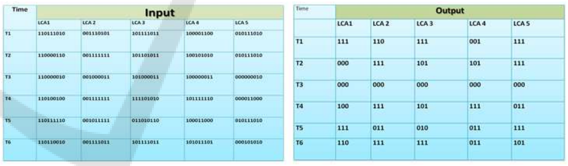


Event	Ref_attack	ProtocolID	SourceIP	VictimIP	VictimPort	Begin_time	End_time
e1	1	1	128178158	10005	80	184501	190000
e2	1	2	128178158	100010	23	191505	193010
e3	2	2	128178157	100071	80	185918	191622
e4	3	5	128178156	100080	53	194501	200050

Figure 3: Local Controller Input Information.

- 111281781581000580184501190000,
- 1212817815810001023191505193010 ,
- 2212817815710007180185918191622,
- 3512817815610008053194501200050...

For example, we assume that the presented attack on the network is Deny of Service (DoS) which is referenced by the number 3. Then the input data to the central controller that contains the sequence ”111” shows that the resource component is attacked by the DoS. The central controller activates nodes that represent network affected hosts by this attack (see figure). An example of the input data and the output data of the central control agent is illustrated by the Figure 4 and an example of the output central controllers graph that represents the attack propagation in network is illustrated by the Figure 5.



Time	Input				
	LCA1	LCA 2	LCA 3	LCA 4	LCA 5
T1	110111010	001110101	101111011	100001100	010111010
T2	110000110	001111111	101101011	100101010	010111010
T3	110000010	001000011	101000011	100000011	000000010
T4	110100100	001111111	111101010	101111110	000011000
T5	110111110	001011111	011010110	100011000	010111010
T6	110110010	001111011	101111011	101011101	000101010

Time	Output				
	LCA1	LCA 2	LCA 3	LCA 4	LCA 5
T1	111	110	111	001	111
T2	000	111	101	101	111
T3	000	000	000	000	000
T4	100	111	101	111	011
T5	111	011	010	011	111
T6	110	111	111	011	101

Figure 4: Central Controller Input and Output Data.

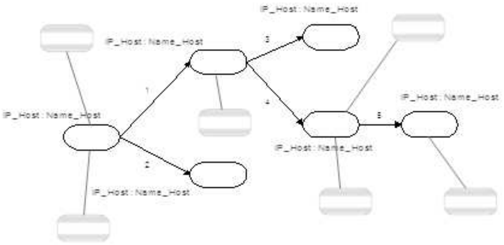


Figure 5: Network Attack Propagation.

6 CONCLUSION

To reach a best level of distributed system security, we have proposed in this article two types of controllers;

the Local Control Agent (LCA) that detects the presence of attacks on its operating system and the Central Control Agent (CCA) that detects the presence of propagated network attacks. Using LAC and CCA are only the two of many resources that can be deployed to increase visibility and control within a corporate computing environment, the concept of defense in-depth is the emphasis on using the best defensive technologies and mechanisms within your organization to craft the appropriate security environment. This paper suggests an architecture that employs both LAC and CCA technologies used together to strongly influence an organizational security posture, using both technologies in a harmony will ensure the needed tools and the appropriate defensive techniques to combat zero day and existing threats while also having the visibility into internal networks and the ability to supply forensic data and trend analysis. On the future works, we aim to develop the agents' structures that perform collaborated detection of composed attacks.

REFERENCES

Zimmermann., J., Ludovic., M., Christophe B. (2003). An Improved Reference Flow Control Model for Policy-Based Intrusion Detection. *In proceedings of the 8th European Symposium on Research in Computer Security (ESORICS)*.

Prigent., N., Bidan., C., Heen. O., Durand. A., (2003). Scurit des rseaux domestiques. *SSTIC'03, 1er Symposium sur la Scurit des Technologies de l'Information et de la Communication. Rennes*.

Ludovic., M., (2003). Dtection des intrusions dans les systmes dinformation: la ncessaire prise en compte des caractristiques du systme surveill. *HDR, Universit de Rennes 1*.

Michel., C., Ludovic., M., (2001). ADeLe: an Attack Description Language for Knowledge-based Intrusion Detection. *In Proceedings of the 16th International Conference on Information Security. KluIr*.

Cuppens., F., Mige., A., (2002). Alert Correlation in a Cooperative Intrusion Detection Framework. *In IEEE Symposium on Security and Privacy*.

Ning., P., Reeves., D., (2002). Constructing Attack Scenarios through Correlation of Intrusion Alerts. *In CCS*.

Ning., P., Reeves., D., Cui., Y., (2001). Correlating Alerts Using Prerequisites of Intrusions. *Technical Report, TR-2001-13, North Carolina State University, Department of Computer Science*.

Ning., P., Cui., Y., Reeves., D., (2002). Analyzing Intensive Intrusion Alerts via Correlation. *In Recent Advances in Intrusion Detection*.

Nguyen., H., Choi., D., (2008). Application of Data Mining to Network Intrusion Detection: Classifier Selection Model. *Springer-Verlag Berlin Heidelberg, pages 399–408*.

Ghosh., A., Michael., C., and Michael., S., (2000). A Real-Time Intrusion Detection System Based on Learning Program Behavior. *Reliable Technologies, USA*.

Selker., T., (1994). Coach: A teaching agent that Learns. *Communications of the ACM, Volume 37, Issue 7.*, pages 547–570.

Moreale., P., (1998). *Agents on the Move. IEEE Spectrum*, pages 34–41.

Green., S., Hurst., L., Nangle., B., Cunningham., P., Somers., F., and Evans., R., (1997). *Software Agents: A Review*. Technical report. Trinity Collega, Dublin, Ireland.

Spafford., H., Zamboni., D., (2000). Intrusion detection using autonomous agents. *Computer Networks, Volume 34, Issue 4.*, pages 547–570.

Jaisankar., N., Saravanan, K., Durai S., (2009). Intelligent intrusion detection system framework using mobile agents. *International Journal of Network Security and Its Applications (IJNSA), Volume 1, Issue 2.*,

Sazzadul., M., Abdul, M., Abu Naser B., (2012). An implementation of i ntrusion detection system using genetic algorithm. *International Journal of Network Security and Its App lications (IJNSA), Volume 4, Issue 2.*,

Anderson D., Frivold T., and Valdes A.,. (1995). *Next-generation intrusion detection expert system (NIDES): A summary*. Technical Report SRI-CSL-95-07, Computer Science Laboratory, SRI International, Menlo Park, California.