Influence of Mobile Agent Security in Distributed System

Akintoye, Kayode A.
Dept of Computer Studies, The Federal Polytechnic, P.M.B 5351, Ado-Ekiti, Ekiti State, Nigeria

Samson B. Akintoye
Dept of Computer Science, Babcock University, Ilishan-Remo, Ogun State, Nigeria

ABSTRACT
Due to the rapid growth of the network application, new kinds of network attacks are emerging endlessly. So to protect the networks from attackers and the Intrusion detection technology becomes popular. There is tremendous rise in attacks on wired and wireless LAN. Therefore security of Distributed System (DS) is become serious challenge. One such serious challenge in DS security domain is detection of rogue points in network. Lot of work has been done in detection of intruders. But the solutions are not satisfactory. This paper gives the new idea for detecting rouge point using Mobile agent. Mobile agent technology is best suited for audit information retrieval which is useful for the detection of rogue points. Using Mobile agent we can find the intruder in DS as well as controller can take corrective action. Mobile agents are gaining in complexity as they evolve and are now widely used in e-commerce. All phases of a business transaction, such as negotiating and signing contracts can be done using mobile agents. In this paper, a brief introduction to the recent researches & developments associated with the field of mobile agents is provided, various security threats is highlighting, and also, the weakest hot-spots of the field which need to be nurtured is touched.

Keywords
Mobile Agent, Intrusion detection system, Distributed System, Rouge Point

1. INTRODUCTION
Computer networks connected to Internet are always exposed to many kinds of cybercrimes. An Internet user with malicious intent can access, modify, or delete sensitive information present on other computers or make some of the computer services unavailable to other users. The infrastructure of current computer networks is so huge and complex that it is almost impossible to completely secure such networks. Therefore, an intrusion detection system (IDS) is needed to detect and respond effectively whenever the confidentiality, integrity, and availability of computer resources are under attack [1] Most of the current distributed IDSs use centralized Intrusion Detection (ID) models made of individual host and network monitors along with a centralized controller component.

The individual monitors send intrusion data to the centralized controller component that performs analysis of the information it receives from each of the monitors. Some of the issues with the existing centralized ID models are:

- Additions of new hosts cause the load on the centralized controller to increase significantly. As a result, it makes the IDS non-scalable.
- Communication with the central component can overload parts of the network.

Some of these IDSs contain platform specific components.

1.1 Motivation
Manual RF scanning is very time consuming and detects rogue AP only when scanning is applied. This leaves ample scope for an attacker to launch attack and finish its work before he gets detected. This is severe loophole of this method.

- *Lack of Efficiency: H*ost-based IDSs often slow down a system and network-based IDSs drop network packets that they don't have time to process.
- *High Number of False Positives:* False alarms are high and attack recognition is not perfect.
- *Limited Flexibility:* Intrusion detection systems have typically been written for a specific environment
- *Limited Response Capability:* IDSs have traditionally focused on detecting attacks. While detection serves a useful purpose, often times a system administrator is not able to immediately analyze the reports from IDS and take appropriate action.
- *No Generic Building Methodology:* In general, the cost of building IDS from available components is considerable, due in large part to the absence of a structured methodology. No such structuring insights have emerged from the field itself.

1.2 Reason for New Requirement
Network-level monitoring and distribution pose some new requirements on intrusion detection systems:

- Networks produce a large amount of data (events). Therefore, a distributed intrusion detection system (DIDS) should provide mechanisms that allow the Network
- Security Officer (NSO) to customize event "collectors" so that they listen for only the relevant events.
- Relevant events are usually visible in only some parts of the network (especially in the case of large networks).Therefore, a DIDS should provide some means of determining where to look for events.
- A DIDS should generate a minimum amount of traffic over the network. Therefore, there should be some local processing of event data.
- A DIDS needs to be scalable. At a minimum, "local" should interoperate with other DIDSs (possibly in a hierarchical structure).

- For maximum effectiveness, NIDSs should be able to interoperate with host-based IDSs so that misuse patterns include both network events and operating system events.

2. LITERATURE SURVEY

Picco in [2] explored the related research fields by showing evidence of the benefits mobile agents can potentially achieve, illustrated the foundations of architectures and technologies for mobile agents, and discussed some of the open issues still hampering a wider acceptance of this paradigm. The author took the concept of conventional distributed systems‟ environment and compares the configuration of the system in context of physical mobility & logical mobility. The goal of the work was to introduce the reader to the research field concerned with mobile agents. The paper presented the conceptual foundations that have their grounds in logical mobility at large, and provided the state of the art in an agent technology. Paper also presented the rationale for using mobile agents, hints at why and when mobile agents are preferable over other solutions, reviewed the basic architectural paradigms for code mobility, including mobile agents and lastly presented reflections on the present status and the future scope of the research area. This work studied two case studies, one is mobile agents for database access and secondly, mobile agents for network management were discussed to research in this field. Advantages of agents have also been highlighted. A distinction is also drawn based on whether the execution state is migrated along with the execution unit or not. Strong mobility & weak mobility has been supported by the systems in response to this distinction.

Knoll, Suri, & Bradshaw in [3] introduced a path-based security for mobile agents. The path-based security provided a mechanism that extended the security of the NOMADS mobile agent system in a multiple-hops scenario. NOMADS supported strong mobility and safe agent execution. Strong mobility allowed the execution state of an agent to be captured and moved with the agent from one host to another [2]. A lightweight protocol for tracking agent paths had been developed that was based on chaining IP addresses. A receiving host environment computed a trust level for the agent, which was then used to choose and apply a security policy to the incoming agent. A TTP (Trusted Third Party) was optionally supported to provide more reliable path information. However, this implementation used just three levels of trust: high, medium, and low. So, the system did not appeal much to support a finer granularity of trust levels. Also, the proposed implementation automatically assigned a low level of trust to hosts that were unknown. Hence, a mechanism must be explored that allows the trust levels to be derived through transitive trust relationships. Also, another mechanism must be incorporated, that dynamically vary the trust levels of hosts based on past history information regarding their behaviour.

Gavalas, Tsekouras, & Anagnostopoulos in [4] proposed a mobile agent technology for the management of networks and distributed systems as an answer to the scalability problems of the centralized paradigm. The authors considered the design and implementation of a complete MAP research prototype that sufficiently addressed the issues such as security mechanism, fault tolerance. MAP has been implemented in Java and optimized for network and systems management applications. They introduced the design decisions and implementation aspects of a complete MAP research prototype that sufficiently addresses all the aforementioned concerns. In comparison to existing approaches, this MAP consolidated several novel design
features, dictated by their design choices and reflected upon their research implementation: (a) a lightweight code distribution scheme, (b) a class loading mechanism that allows the modification of MA-based NSM tasks at runtime, (c) a tool that supports the user-friendly customization of service-oriented MAs, (d) a component that builds nearoptimal network-dependent MA itineraries. In addition, it satisfied other important NSM-related requirements, such as lightweight footprint on systems resources, security (authorization, authentication and encryption), recovery from basic software faults, modularity, incorporation of agent migration optimization techniques, platformindependence, etc. MAP‟s performance has been evaluated in realistic management application scenarios.

Gavalas et. al. also presented the evaluation results of prototype in real and simulated networking environments. Besides all this, there raised some issues which must be fixed such as extensive testing and practical evaluation of MAP in real-world monitoring applications by researchengineers in order to identify potential deficiencies and derive methods for improving the MAP‟s performance; extension of the manager platform with the implementation of a web interface that will allow human administrators to view updated management statistics through a typical web browser and remotely control their network. Future research may compare the performance in terms of latency and
network overhead between this proposed MAP, generalpurpose
MAPs and NSM-oriented MAP (when they become publicly available).

Xiaorong, Su, & Mingxuan in [5] introduced the mobile agent technology based on quantitative hierarchical network security situational assessment model. The researchers designed the distributed computing for large-scale network and evaluated the whole network security situation for future prediction. Network security situational assessment quantitative model is a Hierarchical network consisting of Quantification of security situational index. This index has further different levels: Service-level security situational index, Host-level security situational index, Network system-level security situational index. In this model, the various tasks have been accomplished by an agent: The whole network(system-level) agent network security situation assessment task; the subnet(host-level) agent
network security situation assessment task; and the device(service-level) agent network security situation assessment task. As the security situational assessment is widely applied to the computer network field, many scholars have designed and implemented a large number of network security situational assessment methods. Network security situational assessment technology took all aspects of security factors into account, reflects the overall dynamic state of network security and predicts the development trend of state, which provides a reliable frame of reference. Therefore, network security situational assessment has become a hotspot field in next generation network security technology. The security model proposed by Xiaorong et. al. is distinct and refined method as compared to other models since most works are based on local area network and single host, which hardly meet the demand of large-scale network security assessment. But the technical realization of the quantitative model for further prediction had not been discussed which degrades the quality aspects of the model.

Moussa & Agha in [6] presented the design of "Bosthan", a multi-agent-based simulation tool that managed resources consumption in multi-inhabitants smart spaces. Bosthan had been built on the top of ActorNet mobile agent platform to simulate different smart space topologies with varying numbers of residents. It allowed strategies for resolution of conflicts between mobile agents, and for preserving inhabitants‟ anonymity and untraceability inside the smart spaces. The proposed architecture for bosthan agent-based simulator consists of ten modules: bosthan simulator engine, motion patterns generator, floor planner, sensors layout creator, events generator, location predictor, identity hider, actornet generator, conflict resolver, runs analyzer.

Bosthon was designed in the hope that it would help to compare the efficiency of using mobile agents to allow smart spaces to act pro-actively and maintain anonymity and data privacy in multi-inhabitants environments. Bosthan had been used to study how proposed solution affects the performance and efficiency of smart computing environments without revealing their identity. Bosthan has been designed as a tool for agent-based discrete event simulations that would enable the analysis of different smart space scenarios, ranging from small environments such as
smart rooms, to large smart spaces such as smart homes, smart offices or even smart buildings. The authors used Bosthan to study how mobile agents can be used to provide a resource consumption management framework that preserves inhabitants‟ anonymity and untraceability rules through applying non-interactive evaluation of encrypted functions, without inhabitants‟ intervention, in multiple intersected contexts within a developed smart space. Their goal was to use Bosthan to investigate the effectiveness of strategies to resolve conflicts that may arise between competing agents due to these contexts intersections. This could be applied to simulate environments where residents‟ privacy is highly required. But all these developments are waiting to be implemented, as bosthon is still under under development in the open system laboratory (OSL), University of Illinois at Urbana-Champaign (UIUC), using Microsoft Visual Studio C# 2005.

Rizvi, Sultana, Sun & Islam in [7] provided a solution for securing mobile agent in an ad hoc network. The paper provided a solution for securing mobile agent in an ad hoc network. The authors used Threshold Cryptography in their model, because it provides solution to the problem of central certificate authority (CA) and trusted third party in PKI, by distributing trust among several network nodes. The model provides prime security services like confidentiality, integrity and authenticity. But mobile agent is not free from threats. Although, the model provides a way to secure not only mobile agent, but also the agent server and the agent platform, still it is tough to provide 100% security in an ad hoc network, to detect and prevent vulnerabilities and intrusions. According to threshold cryptography they have considered the value t as a threshold value for the ad hoc network. It means the system can tolerate up to compromised servers. However in this paper, they have not discussed about the cryptographic schema used to generate shares of the Key Management Service's private key. Also they have not discussed about the process of combining the partial signatures or the process to generate the private key of Key Management Service by t+1 servers. Besides these, how authorization and access control service will be provided has not been mentioned. Future research is needed to solve these issues.

Singh, Juneja & Sharma in [8] explained about the working of agent community that it works on the core idea of cooperation and delegation of tasks, which in turn should be prevented from any malicious usage. In order to avoid this malicious usage, an instrument for ensuring proficient and secure communication among these collaborating agents is trust. The authors proposed an elliptical curve cryptography based security engine which extends a novel architecture namely CNTEP which successfully established trust among agents. Encryption of mobile agents and
communicated messages is one of the solutions for ensuring security. However traditional encryption algorithm such as RSA [9], employ key sizes which are very large resulting in high time and space complexity. In contrast to this Elliptical Curve Cryptography (ECC) technique [10, 11] is a public key cryptosystem that besides using much smaller key sizes is able to provide a competitive security edge as that of other strong encryption algorithms. Most attractive feature of ECC is its relatively short operand length compared to that of RSA and also it is based on discrete logarithm in finite fields.

ECC can provide various security services in the form of key exchange, communication privacy through encryption, authentication of sender and digital signatures to ensure message integrity. This paper was an extension of Trust Establishment Protocol (proposed in their earlier paper) and hence an improved version of Contract Net Protocol which was termed as CNTEP [12]. A brief overview of CNTEP architecture as well as elliptical curve cryptosystem was also provided. The authors emphasize that security and trust are two sides of one coin and they must be considered in a way so as to provide a complete security solution for MASs. An attempt to provide such a solution has been made in their proposed model.

The singh et. al. focused on providing both trust among the communicating agents as the first step and after that ensuring the confidentiality and integrity of the messages communicated. Trust Establishment layer (TEL) and Secure communication layer (SCL) were the two components in the model developed. Trust matrix has been maintained to accomplish the process of trust establishment layer. This paper proposed an ECC based security engine for MASs, which provides two dimensional security. In addition to computing trust percentile, the proposed framework makes use of elliptical curve keys for encryption/decryption purpose, which increases message security to a larger extent. Thus this work adds greatly towards improvement of communication security of Multi-Agent Systems. Major advantage of this framework is that existing application will not have to be rewritten to utilize it, instead it can be implemented in the security layer of existing model of wireless communication. The focus of this paper was to propose a security engine for ensuring security of messages communicated in semantic cyberspace. Though, this work adds greatly towards improvement of communication security of Multi- Agent Systems, but the authors remained silent towards the security of agent hosting platforms. So, the need of the hour requires further research in the field of agent technology. Roth & Jalali-Sohi in [13] presented a mobile agent structure which supports authentication, security management and access control for mobile agents. They presented a flexible and extensible structure for the representation of mobile agents which supports hierarchical access control, and proposed an initial interpretation of this structure with respect to the roles

in a general mobile agent model. The structure maps well on existing and widely
distributed technology. It also facilitated security management and the implementation of security services for both while supporting basic agent tasks and operations – the management of data. A portable and interoperable agent structure which is acceptable to all major agent facilities must be established for improvement purposes. This could be achieved by probably feeding the results to a standardization process. The structure as well as the interpretation should undergo a refinement process while being explored within a security centered mobile agent
framework.

3. PROPOSED APPROACH

3.1 Introduction of mobile agent

The Distributed Intrusion Detection System (DIDS) is a project representing an extension of the NSM, with the aim of adding two features missing from NSM. These are the ability to monitor the behavior of a user who is connected directly to the network using a dial-up line (and who therefore may not generate observable network traffic), and the ability to allow intrusion detection over encrypted data traffic. The DIDS project is sponsored by UC Davis, the Lawrence Livermore National Labs (LLNL), Haystack Laboratory and the US Air Force.

- Host agent module: An audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security related events on the host and transmit these to the central manager.
- LAN monitor agent module: Operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the results to the central manager.
- Central manager module: Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion

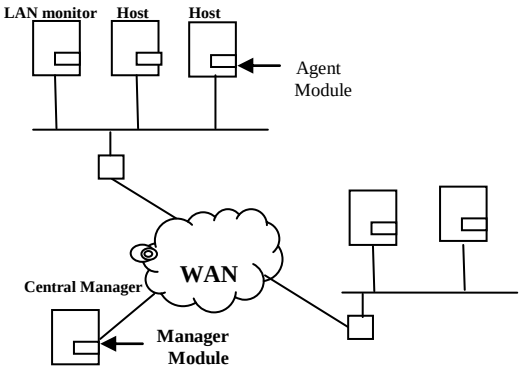

Fig 1: Architecture Distributed Intrusion Detection System

3.2 Characteristics Of Agents

The invasion of various approaches under the banner of ”agents” caused a need to classify and define this term. However it quickly became apparent that everyone had their own definition [14] due in part to the historical relationship with the AI community and the vague notion of intelligence. Numerous definitions for the agents have been proposed, but in core most have a set of defining characteristics that every agent must demonstrate. For instance, Woodridge and Jennings‟ weak agents [15] should be autonomous, reactive and social. A definition from Franklin & Graesser [16] lists autonomous, reactive, communicative, adaptive, mobile, flexible, goal-oriented, continuous and with some form of character or emotion.

- ***Autonomous*** - An agent should be able to execute without the need for human interaction, although intermittent interaction may be required.
- ***Social / Communicative*** - An agent should have a high level of communication with other agents. The most common protocol for agent communication is the Knowledge Query and Manipulation Language (KQML) [17].
- ***Reactive / Responsive*** - An agent should be able to perceive its environment and react to changes in it.
- ***Proactive*** - Proactive agents do not just react to their environment but can take active steps to change that environment according to their own desires.
- ***Adaptive*** - Adaptive agents have the ability to adjust their behaviour over time in response to internal knowledge or changes in the environment around them.
- ***Goal-oriented / Intentions*** - These agents have an explicit internal plan of action to accomplish a goal or set of objectives.
- ***Persistence / Continuous*** - Persistent agents have an internal state that remains consistent over time.
- ***Mobility*** - Mobile agents can proactively decide to migrate to a different machine or network while maintaining persistence.
- ***Emotion*** - Agents with the ability to express human-like emotion or mood. Such agents might also have some form of anthropomorphic character or appearance.
- ***Intelligence*** - Agents with the ability to reason, learn and adapt over time.
- ***Honesty*** - Agents that believe in the truthful nature of the information they pass on.

These agent characteristics lead to many advantageous features. The very nature of agents as independent, social entities that can respond to and change their environment provides a strong foundation for building reliable, robust, flexible, extensible and scalable systems. Agents can help ease user tasks and adapt to user requirements. Despite of their many benefits, agents are not the solution to every problem. One major disadvantage of building agent systems is that the complexity of agent interactions and dynamic nature of the agents themselves make it difficult to predict agent behaviour. It can cause problems in safety critical environments where outcomes need to be assured. However as the computer world is becoming increasingly networked and distributed, agents are likely to become the next engineering paradigm for system development.

3.2 Mobile Agent Architecture

The scheme is designed to be independent of any operating system or system auditing implementation. Figure 2 shows the general approach that is taken. The agent captures each audit record produced by the native audit collection system. A filter is applied

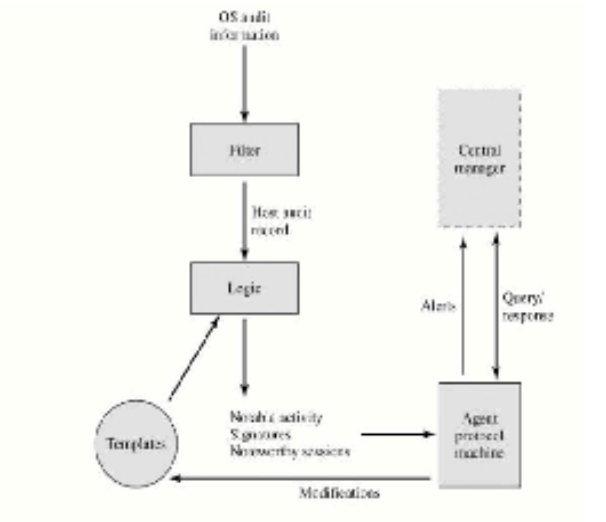

Fig 2: Architecture of Mobile Agent

that retains only those records that are of security interest. These records are then reformatted into a standardized format referred to as the host audit record (HAR).

Next, a template-driven logic module analyzes the records for suspicious activity. At the lowest level, the agent scans for notable events that are of interest independent of any past events. Examples include failed file accesses, accessing system files, and changing a file's access control. At the next higher level, the agent looks for sequences of events, such as known attack patterns (signatures). Finally, the agent looks for anomalous behavior of an individual user based on a historical profile of that user, such as number of programs executed, number of files accessed, and the like Here we propose architectures to detect rogue point using mobile agent. Below are abbreviations used in these architecture.

SA- Server Application
CA- Client Application
MAS- Mobile Agent System
MA- Mobile Agent

4. CONCLUSION

Agent technology has been used in many critical applications such as personal information management, electronic commerce, business process management, artificial intelligence, interface design, distributed processing and distributed algorithms. Besides its bright side, the technology has encountered many security threats. These problems are faced during the itinerary period of an agent traversing from platform to platform in the network. This paper have surveyed various recent developments, researches and proposals related to the field of agents and have thrown some light on the delicate areas that needs to be paid more attention to promote growth in optimistic direction.

5. REFERENCES

[1] Mohammad Zulkernine -" DIDMA: A Distributed Intrusion Detection System Using Mobile Agents" proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05)

[2] G.P.Picco,"Mobile agents: an introduction",Microprocessors and Microsystems 25(2001) pp. 65-74, Dipartimento di Elettronica eInformazione, Politecnico di Milano, Milan, Italy.

[3] G. Knoll, N. Suri, and J.M. Bradshaw, "Path-based Security for Mobile Agents", Electronic Notes in Theoretical Computer Science, Vol. 58, No. 2 , pp. 16, (2002).

[4] D. Gavalas, G.E. Tsekouras, C. Anagnostopoulos, "A mobile agent platform for distributed network and systems management", In Journal of Systems and Software 82 (2), 355-371, 2009.

[5] C. Xiaorong, L. Su, L. Mingxuan, "Research of Network ecurity Situational Assessment Quantization Based on Mobile Agent", Volume 25, 2012, Pages 1701–1707, International Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao.

[6] S. M.. Moussa, G.A. Agha, "Integrating Encrypted Mobile Agents with Smart Spaces in a Multi-agent Simulator for Resource Management", Journal of Software, Vol 5, No 6 (2010), 630-636, Jun 2010.

[7] S.M.S.I. Rizvi, Z. Sultana, B. Sun, and Md. W. Islam, "Security of Mobile Agent in Ad hoc Network using Threshold Cryptography", World Academy of Science, Engineering and Technology 70- 2010.

[8] A. Singh, D. Juneja, and A.K. Sharma, "Elliptical Curve Cryptography Based Security Engine for Multiagent Systems Operating in Semantic Cyberspace", In International Journal of Research and Review in Computer Science (IJRRCS), Vol. 2, No. 2, April 2011.

[9] R. Shanmugalakshmi and M. Prabu, "Research Issues on Elliptic Curve Cryptography and its applications", In International Journal of Computer Science and Network Security, Vol. 9, No.6, pp 19- 22, June 2009.

[10] N. Koblitz, "Elliptic Curve Cryptosystems", Mathematics of Computation, Vol. 48, pp. 203-209, 1987.

[11] V.S. Miller, "Use of Elliptic Curves in Cryptography", Advances in Cryptology- CRYPTO"85, LNCS, vol. 218, Springer-Verlag, pp. 417- 426,1986.

[12] A. Singh, D. Juneja, A.K. Sharma, "Introducing Trust Establishment Protocol in Contract Net Protocol". In Proceedings of IEEE International Conference on Advances in Computer Engineering (ACE"2010), pp. 59-63, June, 2010.

[13] V. Roth & M. Jalali-Sohi, " Access Control and Key Management for Mobile Agents", Fraunhofer Institute for Computer Graphics, Rundeturmstr. 6, 64283 Darmstadt, Germany, 8 November, 2001.

[14] H.S. Nwana, "Software Agents: An Overview", Knowledge Engineering Review, 11(3):1- 40, 1996.

[15] M. Woodridge and N. Jennings, "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review, 10(2):114-152, June 1995.

[16] S. Franklin, A. Graesser, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", University of Memphis, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag,1996.

[17] T. Finin, Y. Labrou & J. Mayfield, "KQML as an Agent Communication Language", J. Bradshaw (Eds), MIT Press, 291-316,1997.