Research Journal of Applied Sciences 9 (9): 583-589, 2014
ISSN: 1815-932X
© Medwell Journals, 2014

Techniques for Multi-Agent System Security

[1]Olumide Simeon Ogunnusi, [2]Obasan Adebola Olukayode and [1]Michael Kolade Adu
[1]Department of Computer Studies, Federal Polytechnic, P.M.B. 5351, Ado-Ekiti, Nigeria
[2]Department of Mathematics and Computer Science,
Kaduna Polytechnic, P.M.B. 2021, Kaduna, Kaduna State, Nigeria

Abstract: Security issues associated with mobile agents and host resources have raised serious obstacles in practical applications of multi-agent system paradigm which is a vital component of mobile computing applications. These security considerations have hindered its scope of relevance in the industry. Mobile computing is a hot area of research and a good number of researchers have made remarkable efforts at overcoming the security threats linked to multi-agent systems. The scope of this study, therefore, is limited to the survey and analysis of the existing security techniques for multi-agent systems. It studies the available security solutions by researchers for multi-agent systems and analyse them based on performance, requirements, complexity and their support for agent mobility.

Key words: Multi-agent system security, security threats, countermeasures, authentication, authorisation, agent mobility

INTRODUCTION

Multi-agent system is made up of multiple autonomous, intelligent and robust software agents (Aggarwal *et al*., 2012). Agent technology has become increasingly popular in the recent time due to its enormous research interests and increasing adoption in the commercial front. Researchers give kudos to researchers for unveiling the hideouts of the cyber criminals and proffer unbeatable security solutions to the menace. Mobile agent plays key roles in the performance of multi-agent systems especially in multi-agent system based applications where information are carried from one location (host) to another. For example, in e-Commerce, mobile agents carry payment authorization value, digital cash token or credit card number, negotiate and pay on behalf of their owners for purchase of goods and services on the internet. However, security issues associated with mobile agents have been studied by Garrigues *et al*. (2010), Karnik and Tripathi (2001), Marques *et al*. (1999), Jansen and Karygiannis (1998), Malik *et al*. (2010) and McDonald *et al*. (2005). In order to secure an agent host's attributes and visiting mobile agent against possible attacks by Malicious Agent System and/or malicious mobile agent, the system must be protected with strong security policy (Loulou *et al*., 2006). Protection of agent host resources against malicious agents is paramount likewise the protection of agents against malicious host. The data carried by agents can be tampered with by the malicious host since the host has full control over the agents running on it. As a result of this, trust has been embraced as a tool that could be used to guarantee the safety of mobile agent running on a host. Agent owners must establish a trust host community within which their agent traverse while the suspected and identified malicious hosts are blacklisted and reap zero patronage. This study identifies and analyses the various security solutions proposed for multi-agent systems.

AGENT SYSTEM SECURITY

The basic measures of reliability of any agent-based distributed system are confidentiality, integrity, availability, accountability and non-repudiation. Information stored on platform or carried by an agent are prone to confidentiality threat (Telepovska, 2007). Secure communication must be guaranteed in an agent system to ensure a messages sent by an agent is securely delivered to the right recipient. The agent execution environment also requires confidentiality protection. For example, the directory facilitator that provides yellow page for all agent must be adequately secure, otherwise a malicious agent can delete or alter its contents for a gain at the detriment of some other agents. A malicious agent can delete the address of the current location of another agent from the platform directory thereby rendering such agent untraceable when message is sent to it.

Corresponding Author: Olumide Simeon Ogunnusi, Department of Computer Studies, Federal Polytechnic, P.M.B. 5351, Ado-Ekiti, Nigeria

The principals in an Agent System are agent platform, agent owner, agents, resources and platform administrator. Each of these principals combats with security threats. It becomes mandatory for a good security mechanism to protect them and their communication against security threats (Oey *et al*., 2010). For instance, communication of sensitive information between two agents from the same or different platforms must be secure to prevent information leakage to the adversary. Perhaps the information in transit with the agents may be the confidential data of the agent owner. Similarly, malicious agent can corrupt or outrightly delete host resources. An agent sent to bid for a contract for example, on arrival at the destination host may manipulate the figures of other bidders in order to boost its chance of winning the contract. This kind of threat is peculiar to an open network environment and not usually common with multi-agent agent system running in closed network environment because both the agents and the platforms are assumed to be reliable, therefore, security in the environment is often implied.

Security threats to multi-agent systems and countermeasures: This study discusses the various security threats to multi-agent systems and the numerous counter measures proposed by researchers to overcome them.

Security threats to multi-agent systems: Threats are acts, incidents or confrontations that compromise security. Threats to multi-agent systems are broadly classified into confidentiality, integrity and availability (Cavalcante *et al*., 2012). Confidentiality threat is concerned with the breach of privacy of an entity such as agent, platform resources or agent owner. This class of threat can be posed by agent or platform. For example, a platform by virtue of its full control over agent can attempt to eavesdrop on agent sensitive data and breach the privacy of the secret data of agent's owner that must not be disclosed to others. Integrity threat is concerned with the modification to agent components, agent owner's data, platform resources or information in transit (i.e., agent communication). Agent and platform resources modification can also be posed by both agent and platform against each other.

Availability threat has to do with one entity wittingly depriving an entity from accessing resources for which it has privilege. For example, a host platform may decline communication support for agents set to collaborate for achieving their designed objective. The categories of threats to mobile agents such as agent to agent, agent to platform, etc. and the various threats associated with each category are further discussed by Borselius (2002), Ahuja and Sharma (2012), Aggarwal *et al*. (2012) and Picco (2001).

Countermeasures: The pre-requisite for securing confidentiality, integrity and availability identity management (Claub and Kohntopp, 2001) comprises naming and authentication. Naming, on one hand gives the opportunity to uniquely identify each principal in an agent system and to mitigate repudiation. The name given to a principal can be static (i.e., such name remain unchanged throughout the lifetime of the principal especially for humans and organisations) or dynamic. An agent can be given different names in order to reflect the current location of the agent (location dependent name). Also, dynamic name of an agent can be used to implement anonymity for such agent using pseudonyms. Anonymity is used to protect the agent against adversary. On the other hand, authentication provides justification for authorisation. For instance, for an agent to have access to a host resources, the host must be able to ascertain that the agent is a legitimate or harmless agent and know the sender host of the agent before granting permission to such agent. One well known authentication technique is the use of combined username and password.

The correct entry of the two parameters confirms the authenticity of the principal (host, agent, agent owner, etc.). However, this authentication scheme is grossly deficient as it is vulnerable to guessing attack and impersonation attack (Wen *et al*., 2014). The use of Public Key Infrastructure (PKI) provides a more robust authentication scheme using public key cryptography (Batten, 2013). With this scheme, every principal is issued a duly electronically signed certificate by the certificate authority. The certificate contains the principal's name, private/public key pair and the key validity date. Every principal publishes its public keys and makes it known to all while its private key is kept secret (i.e., not known to anyone except the principal itself). The legitimacy of a principal is verified when a message encrypted with its public key is forwarded to it and is able to use its private key to decipher the message. The message decryption task will be impossible if the principal does not possess the right private key for the encrypted message. The PKI ensures that a message that is not meant for a principal is non-intelligible to it even if the principal succeed in intercepting the message. Using PKI to solve authentication problem of agent systems, each agent, agent 's owners and host must have unique private/public key pair through which they can be authenticated and set up secure communication channels using SSL (Oey *et al*., 2010).

Mobility of mobile agent introduces other security risks since agent owners are not familiar with environment their agent traverse. For example, cyber attackers intentionally start agent platforms with the mission of exploiting any innocent vising agent and strip them of secret data. For example in e-Health (Su and Wu, 2011), a patient agent can be intercepted by the malicious host and be robbed of the secret information carried by the agent. Such stolen information can be used by the attackers to slander blackmail or carry out more devastating attack on the human patient. This malicious host problem is difficult to solve, however a number of mechanisms have been proposed such as code obfuscation, environmental key generation, code on demand and Encrypted Function Model. These mechanisms are established for agent code protection.

Role-Based Access Control (RBAC) mechanism are also proposed in Braubach *et al.* (2013) and Vieira-Marques *et al.* (2006) as an authorisation technique to restrict principals to access only resources for which they are given permission. The mechanism defines the resource(s) for which a principal can access and the extent to which such access is granted and bar it from accessing other resources for which permission is not granted. Cryptographic schemes are also used to strengthen permission rights. A typical availability threat can target the lookup service of a host platform. The lookup service contains the database of the agents' current locations. A malicious agent for example can delete or manipulate the information in this lookup service database and hence rendered it unreliable and consequently paralyse the entire agent system.

SECURITY ARCHITECTURE FOR MULTI-AGENT SYSTEM

Agent systems require middleware support which cannot be provided by the underlying operating system (Aggarwal *et al.*, 2012). The middleware, as an interface between the operating system and mobile agents, provides a suitable execution environment referred to as agent platform where the mobile agent applications can be developed and managed. It also provides the resources needed by mobile agents to permit collaboration, coordination, communication and accomplishment of their designed objectives. A number of agent system middleware has been developed by research groups and industry such as MANSON TCL, SOMA, GRASSHOPPER, ACE, AgentScape, AGLET, D'Agent, JADE, ARA, CONCORDIA, SPADE, HIVE, Ajanta, WSS, etc.

Security approach in Secure and Open Mobile Agent (SOMA): In SOMA (Bellavista *et al.*, 2000), authentication and authorisation are imposed at two levels: domain and place. The agent credentials are used to authenticate the agent before it enters the domain. A credential contains the name of the agent originating domain and place and the name of the agent principal whom it represents. The agent is granted permission to access the resources once it is authenticated but based on the current security policy. Every resource has a specific Role-Based Access Control (RBAC) list for all principals (Sandhu *et al.*, 1996). In Role-Based Access Control Model, permission are assigned to roles instead of individual user to ease manageability so that a group of roles is given the same permission right and any individual whose role falls within a particular role group automatically inherits the permission right given to the group of roles. SOMA protect the secrecy of agent hopping in untrusted environment using authenticated and encrypted channels while its integrity is protected using protocols suitable for different application scenarios.

Security approach in SAgent: SAgent is a general-purpose mobile agent security framework designed primarily for the protection of computations of mobile agent applications in potentially malicious hosts (Gunupudi and Tate, 2006). It is designed to work with JADE and it is concerned specifically with protecting the data that travel with mobile agents. The design view of SAgent separates the public and private functionality of agent into distinct pieces with two different views. One from the point of view of programmer that develop protection techniques for SAgent and the other from the point of view of developer who writes applications for SAgent. The architecture of SAgent is designed such that the software provider and the developer of application are unaware of each other and they develop protocol and application independent of each other.

Security approach in Mansion Tcl: Mansion framework (Van Noordende *et al.*, 2001) consists of a world containing a set of hyperlinked rooms each contains agents and objects. At an instance, an agent is in one room but has the privilege to move from one room to another and take along objects in the same room. It can also send messages to other agents anywhere in the world. In Mansion, no part of an agent can be accessed from outside by other agents. Mansion provides a middleware layer for multi-agent systems which provides the basic primitives for interaction with the world such as migration, agent communication, etc., mansion

middleware supports distribution and security policies and provides location transparency for all logical entities.

Security approach in SPADE: SPADE provides some mechanisms, developed in different security levels, to maintain system integrity (Gregori *et al*., 2006) such as:

- Username and password are required to log in to the XML router to prevent unauthorised connection to the platform
- Symmetric cryptography using SSL can be used to encode connection to the XML router. SSL provides data cyphering, server authentication, message integrity and client authentication for TCP/IP connections
- XML router guard against identity theft by providing another mechanism to ascertain message authentication

Security approach in AgentScape: AgentScape uses Public Key Infrastructure (PKI) whereby agent owners, locations and hosts possess public key pairs. This is to ensure that locations and hosts can mutually authenticate and set up secure communication channels using SSL. The agent execution environment also ensures that each agent has Globally Unique Identifier (GUID) for addressing the agent and perform operations on it such as message delivery, migrate, suspend and even kill. To ensure agent communication protection, every message transmitted between hosts including migration of agents are encrypted to facilitate confidentiality. AgentScape uses cryptographic primitives to create a digitally signed checksum of data transmission between hosts in order to provide integrity support (Quillinan *et al*., 2008).

Security approach in concordia: The Concordia Security Model provides security to:

- Secure transfer of agents using SSLv3
- Protect host resources attack by malicious agent
- Protect agent against attack by a malicious agent

However, Concordia does not protect agent against attack by malicious server. Once a server has been identified as Concordia server, it is assumed to be a trusted server. Concordia uses resource permission built on top of Java security classes to control access to files or network resources (Walsh *et al*., 1998). It controls the ability to create new threads or processes; the ability to change java virtual machine's operating properties; the ability to load non-java code and the ability to access the system's console or graphical interface.

Security approach in grasshopper: Grasshopper provides two mechanisms, one for internal security and the other for external security. Internal security protects resources of an agency from unauthorised access by agents and also protects agents from one another. The access control policy of grasshopper is based on identity and group membership initialized at start-up. This access control policy comprises an access control list containing several entries, one for each object treated in the policy where a subject can be a single identity or a group made up of 1…n members. In this policy, a set of permissions is associated with each subject, ranting access to all important parts of grasshopper agency. Each permission comprises a type, a target and optionally one or more actions. On the other hand, the external security protects remote interactions between the distributed grasshopper components using a security mechanism based on X.509 certificates and SSL. The SSL provides confidentiality, data integrity and mutual authentication of client and server. Grasshopper uses RSA with 1024 bit keys for authentication.

Security approach in Ajanta: Ajanta security manager is used to protect an agent's access to system level resources. It uses access control list defined by URNs to grant access to its system level resources such as network ports and system files. The identity of the agent's owner is used to determine which access is granted to the agent. Ajanta security manager restrain agent from creating threads in a group other than the thread assigned to it and also prevent it from creating and installing a class loader (Karnik and Tripathi, 2001).

Ajanta uses a challenge-response based authentication protocol to implement authentication in communication. Each entity in Ajanta holds its keys comprising El-Gamal public key for encryption and DSA public key for digital signatures which are registered with Ajanta 's name service. The DSA key and algorithm is used to securely authenticate a client to a server and vice versa. Replay attacks are prevented with the authentication protocol developed using a challenge-response mechanism with randomly generated nonces (Abadi and Needham, 1996). The authentication protocol of Ajanta is not a network-level protocol but rather operates at the application level only. The identities being authenticated are URNs of the entities such as agents, agent owners, agent servers and name resolvers.

Table 1: Comparative analysis of some multi-agent systems

Agent system	Agent Identification	Key distribution	Agent protection	Host protection	Communication protection	Support for agent mobility
SOMA	Uses agent credential containing originating domain name, place, agent principal name, and name of class implementing the agent	X.509 certificate	Use of authentication and protocol for agent privacy and integrity, respectively	The use of RBAC. Each host resource possesses specific access control list for all principals	Use of Secure Socket Layer (SSL) for channel encryption	Yes
AgentScape	Uses agent meta data comprising agent GUID, name of agent 's owner and signed hash of agent code	Public Key Infrastructure (PKI)	Migration of agent to trusted host	Uses authentication and authorisation mechanisms with RBAC mechanism for host protection	Secures communication channel using SSL and the use of cryptographic primitive to create digitally signed checksum of transmitted data	Yes
SAgent	Uses agent certificate to identify agent	No distribution method is specified	Use of encrypted circuit constructed to protect the sensitive and computational part of an agent	N/A	Use of encryption function to encrypt message before transmission	Yes
SPADE	Uses Jabber identifier to identify agent	No distribution method is specified	No specific protection for mobile agent	User must logging to XML router	Connection to XML router is encoded with symmetric Cryptographic algorithm using SSL, Provision of identity spoofing for message authentication	No
Concordia	Every agent is assigned a unique identity	N/A	Use of encryption to protect agent in client store and persistent store	Uses Concordia security manager to protect server resources	Uses SSLv3 protocol for transmitting agent information across network	Yes
Grasshopper	Unique agent identity and/or group identity	X.509 certificate	Authentication and authorisation of agent user	Authentication and authorisation of agent user	Uses SSL for agent communication protection	Yes
Ajanta	Uses agent credentials containing agent owner's signature, URNs of agent, agent owner, agent server and agent resolver to uniquely identify an agent	Uses cryptographic mechanism to declare agent data read only	Uses thread grouping and class loading to create protection domain for agents, one-way hashing and digital signature are used to detect tampering, eavesdropping is prevented using encryption, part of agent is declared read-only so as to detect tampering by agent host, allow the agent to create append-only container for storing data during execution	Uses Ajanta security manager to protect system level resources with the use of access list	Uses agent transfer protocol to secure communication between agent servers	Yes

ANALYSIS OF SECURITY TECHNIQUES IN MULTI-AGENT SYSTEMS

There are variations in the security focus of the multi-agent systems as shown in Table 1. Some of them focus on securing the sensitive data carried by mobile agent such as SAgent. For example, SAgent ensures that the sensitive agent's principal data and the intermediate data generated during its itinerary are returned safely to its principle without breach of their confidentiality and integrity; SPADE emphasises on agent communication channel protection against any form of attack that could compromise the confidentiality and integrity of message transmitted between two agents. The variations in the security concerns of multi-agent systems must be noted by multi-agent system application programmers in selecting the appropriate middleware for the deployment of applications.

CONCLUSION

There is no general security technique for multi-agent systems. The environment of multi-agent system based application determines the suitable middleware with the required security components that is appropriate for the application. For example, mobile agents that are carrying

sensitive and secret information must have such information protected against confidentiality breach. Also, mobile agents that must migrate and collect information from multiple hosts must be protected against possible tampering by malicious host. Similarly, in open MAS, agent hosts must be protected to guard against the possibility of attacks by malicious agents. This study has been able to discuss the security techniques adopted by some multi-agent systems to facilitate the protection of agent components, information carried by agents and the platform on which the agents are running. This could assist users to choose the appropriate MAS middleware that would be suitable for their application needs considering the performance, requirements, complexity and their support for specific attribute such as mobility.

REFERENCES

Abadi, M. and R. Needham, 1996. Prudent engineering practice for cryptographic protocols. IEEE Trans. Software Eng., 22: 6-15.

Aggarwal, S., S. Bhardwaj and P. Kumar, 2012. Security approaches for mobile multi-agent system. Int. J. Emerging Technol. Adv. Eng., 2: 681-687.

Ahuja, P. and V. Sharma, 2012. A review on mobile agent security. Int. J. Recent Technol. Eng., 1: 83-88.

Batten, L.M., 2013. Public Key Cryptography: Applications and Attacks. Vol. 16, John Wiley and Sons, New York, ISBN: 9781118482315, Pages: 224.

Bellavista, P., A. Corradi, R. Montanari and C. Stefanelli, 2000. How a secure and open mobile agent framework suits electronic commerce applications. http://wendang.baidu.com/view/742eca124431b90d6c85c73d.html.

Borselius, N., 2002. Mobile agent security. Electr. Commun. Eng. J., 14: 211-218.

Braubach, L., K. Jander and A. Pokahr, 2013. A practical security infrastructure for distributed agent applications. Proceedings of the 11th German Conference on Multiagent System Technologies, September 16-20, 2013, Koblenz, Germany, pp: 29-43.

Cavalcante, R.C., I.I. Bittencourt, A.P. da Silva, M. Silva, E. Costa and R. Santos, 2012. A survey of security in multi-agent systems. Expert Syst. Appl., 39: 4835-4846.

Claub, S. and M. Kohntopp, 2001. Identity management and its support of multilateral security. Comput. Networks, 37: 205-219.

Garrigues, C., S. Robles, J. Borrell and G. Navarro-Arribas, 2010. Promoting the development of secure mobile agent applications. J. Syst. Software, 83: 959-971.

Gregori, M.E., J.P. Camara and G.A. Bada, 2006. A jabber-based multi-agent system platform. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, May 8-12, 2006, Hakodate, Japan, pp: 1282-1284.

Gunupudi, V. and S.R. Tate, 2006. SAgent: A security framework for JADE. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, May 8-12, 2006, Hakodate, Japan, pp: 1116-1118.

Jansen, W. and T. Karygiannis, 1998. Mobile agent security. Defense Technical Information Center-DTIC. http://www.dtic.mil/docs/citations/ADA391526.

Karnik, N.M. and A.R. Tripathi, 2001. Security in the Ajanta mobile agent system. Software: Practice Exp., 31: 301-329.

Loulou, M., M. Jmaiel, A. Hadj Kacem and M. Mosbah, 2006. A conceptual model for secure mobile agent systems. Proceedings of the International Conference on Computational Intelligence and Security, November 3-6, 2006, Guangzhou, China, pp: 524-527.

Malik, N.S., F. Kupzog and M. Sonntag, 2010. An approach to secure mobile agents in automatic meter reading. Proceedings of the IEEE International Conference on Cyberworlds, October 20-22, 2010, Singapore, pp: 187-193.

Marques, P.J., L.M. Silva and J.G. Silva, 1999. Security mechanisms for using mobile agents in electronic commerce. Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems, October 19-22, 1999, Lausanne, pp: 378-383.

McDonald, J.T., A. Yasinsac and W. Thompson, 2005. Trust in mobile agent systems. Technical Report, Department of Host Science, Florida State University.

Oey, M.A., M Warnier and F.M.T. Brazier, 2010. Security in large-scale open distributed multi-agent systems. Autonomous Agents, 6: 108-130.

Picco, G.P., 2001. Mobile agents: An introduction. Microprocessors Microsyst., 25: 65-74.

Quillinan, T.B., M. Warnier, M. Oey, R. Timmer and F. Brazier, 2008. Enforcing security in the agentscape middleware. Proceedings of the Workshop on Middleware Security, December 2, 2008, Belgium, pp: 25-30.

Sandhu, R.S., E.J. Coyne, H.L. Feinstein and C.E. Youman, 1996. Role-based access control models. IEEE Comput., 29: 38-47.

Su, C.J. and C.Y. Wu, 2011. JADE implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. Applied Soft Comput., 11: 315-325.

Telepovska, H., 2007. Agent system security. J. Inform. Control Manage. Syst., 5: 363-370.

Van Noordende, G., F. Brazier, A.S. Tanenbaum and M. van Steen, 2001. Position summary: Mansion, a distributed multi-agent system. Proceedings of the 8th Workshop on Hot Topics in Operating Systems, December 2008, San Diego, CA., USA.

Vieira-Marques, P.M., S. Robles, J. Cucurull, R.J. Cruz-Correia, G. Navarro and R. Marti, 2006. Secure integration of distributed medical data using mobile agents. IEEE Intell. Syst., 21: 47-54.

Walsh, T., N. Paciorek and D. Wong, 1998. Security and reliability in Concordia™. Proceedings of the 31st Hawaii International Conference on System Sciences, Volume 7, January 6-9, 1998, Kohala Coast, HI., pp: 44-53.

Wen, F., D. Guo and X. Li, 2014. Cryptanalysis of a new dynamic id-based user authentication scheme to resist smart-card-theft attack. Applied Math. Inform. Sci., 8: 1855-1858.