A Novel Security Framework for Agent Based Distributed Systems

ALTAMASH JILANI
Altamash.jilani@gmail.com

Abstract
The mobile agent (MA) paradigm seem to be a promising technology for developing application in open, distributed and heterogeneous environments, such as internet. Many application areas, such as electronic commerce, mobile computing, network management and information retrieval can benefit from the application of the MA technology. However the wider diffusion of MA is currently limited by the lack of comprehensive security framework that can address the security concerns arising in mobile agent application providing efficiency at the same time.

This paper describes the security framework to protect the host from attacks of malicious agent and a self executing security schema for protecting the agent from the attacks of malicious hosts and other agents.

1. Introduction
The Mobile Agent (MA) paradigm is an emerging technology for developing applications in open, distributed and heterogeneous environments, such as the internet. Mobile agents are software entities that can block execution in their allocation node and resume it later in another network node. Agents are able to decide autonomously where to act after migration. MA technology offers several advantages in many application areas, such as electronic commerce, mobile computing, network management and information retrieval. Mobile agents are designed to execute locally to the information they have operate upon, thus reducing network traffic and latency. In addition, the MA asynchronous interaction model can provide efficient solutions in the case of unreliable and low-bandwidth connection and in the support to mobile users that could disconnect while their agents roam in the network.

However, one of the main technical obstacles to a wider acceptance of the MA paradigm is security. Achieving security is fundamental for the successful development of mobile agents system. The MA paradigm can be of interest for the security concerns: both execution environments and agents are subject to unwanted attacks and require appropriate protection mechanisms. On the one hand, the problem of host protection against malicious agents has been extensively investigated. On the other hand, mobile agent protection against malicious behavior of execution environments is specific to MA technology and represents an important and challenging research area investigated only by a few proposals. In addition, mobility introduces new threats: hostile hosts may refuse to execute agent’s code and to transfer the agent to successive execution sites or may even tamper with agent code and state.

In this paper section 2 give a glance view of related work section 3 describes the Security framework for protecting host from malicious agent attacks section 4 presents the self executing security schema for protecting agents from malicious attacks, in section 5 implementation is presented. Finally the conclusion and future works in section 6.

2. Related Work
Distributed systems are heterogeneous in nature. So it is difficult to monitor the distributed system for management purpose. Mutual exclusions are very common in paradigm of agent based system. It is difficult to solve it with fault tolerance and platform independency.

In [1] a method of monitoring distributed system is proposed to reduce the complexity, using flexible agent and applying concepts from the field of relational database. The proposed method consists of the following steps: queries about the management information are presented in a standard query language like SQL. In [3] the idea of agent based information recommending system is proposed to increase searching quality, that used the static information and has aim to provide the information to user in which user interested and user really need. The system has two recommending methods. The first method is that the system selects information by user’s preference and recommends the information for user. The second method is that the system watches the user’s daily web browsing and
searching and deduces the information user needs from above user behavior, and recommends new information for user. In [5] a data consensus algorithm for synchronous distributed system using cooperating mobile agents is proposed to ease the consensus of data of synchronous distributed system for maintaining data consistency and ensuring performance. In [7] check pointing scheme using mobile agent is presented to reduce the overhead on network and delay time of message passing, that improve the system performance on wide area distributed system.

The centralized intrusion detection systems significantly increases the load on network traffic, due to this system reliability and scalability goes down and present intrusion detection systems are not fully capable to handle the attacks via network. In [8] a new intrusion detection system (IDS) called distributed intrusion detection using mobile agent is presented, which uses a set of software entities called mobile agents and static agents to perform aggregation and correlation of the intrusion related data without overloading the network bandwidth and increase the scalability, reliability and platform independence. The security of mobile agents is one of the important issue because mobile agents itself can cause damages to host or host can do harm to the mobile agent [8].

In [9] a new architecture of intrusion detection system is proposed. The proposed architecture of intrusion detection system is as follows: the network will divide the 4 zones network zone, internal network zone, wireless zone and DMZ (Demilitarized zone). This idea has to objective to not being the target that can attack easily and form the attacking. It also and can limit the damage when there is an attack.

In distributed environment it is very difficult to handle the exploitation of information contained in other information source through local schema. In [10] a distributed algorithm for query answering is proposed using multi agent structure to ease the exploitation of information contained in other information source through local schema. The architecture used in this paper is based on multi agent structure. In this structure there is no central agent, which plays the role of the mediator to integrate all information sources. In distributed system security management is very difficult where operating system doesn’t support the distributed security. In [11] Reliable and secured system architecture is proposed for host foreign applications which are charged for their resource consumption. The autonomous station does not rely on an external power supply system, but it comprises a unit for the generation of current in order to ensure its autonomy. The autonomous station is a multipurpose, customizable and extensible system, which may be deployed in many different configurations. In [12] A security architecture for a mobile agent system is introduced which guarantees security for the host as well as security for the agent. A security architecture for mobile agents is introduced that can be used on of existing systems. They assume that every participant in the agent system possesses a certified public key. In [13] Security framework is proposed for object based distributed system; it allows the development of secure distributed applications on existing operating systems that do not support distributed security. The proposed security framework is being developed in Orbix implementation of CORBA. The aim of design is: (i) to provide a security platform for distributed applications that makes the access control and authentication mechanism transparent to the application level, and (ii) to support the enforcement of access control policies that are specified using management domain. In [14] a security framework is presented to secure the agents from interference of un trusted and potentially malicious hosts. When an agent starts to execute in a host it is possible that its data and actions are spied. Also it is possible that the host interferes and manipulates the agent’s flow of execution and data. A malicious host may try to read and modify offers from other hosts or even change the flow of execution of the agent in order to force it to take an offer being proposed. Dependent computation of previous work is not supported if host is un trusted [14]. In [15] Agent Security requirements are analyzed to protect the agent’s by attack of malicious host, attack of other agents and unauthorized parties. Security in mobile agent system can be analyzed in four different perspectives: Protecting host from access by unauthorized parties, protecting hosts from attack of malicious hosts, protecting agents from attack of other agents. Public key cryptography is used in signing the agents, host authentication and used secure communication channel like SSL for protection of host from access by unauthorized parties. In [17] A Distributed Security Infrastructure (DSI) for handling security in...
clustered server is proposed to identify necessary distributed security services to enhance the cluster security and to protect the distributed system from public network attacks. Enhancement required in scalable cluster monitoring and intuitive human interfaces to security tools. In [18] a secure group solution is proposed using agent based multicasting for securing the multi agent ecommerce system with reduces complexity and better performance. The proposed method is only from selected environment not for all environments.

In [20] Security requirements are derived from general model of mobile agent based management system comes after analyzing the threats and attacks against mobile agent system used for management purpose. In [21] Policy Programming Language (PPL) is introduced using Domain Specific Language to provide the same level security to all resources and components in distributed system. In [21] Plan to optimize the process of conflicts detection and corrections.

In [22] Hybrid trust model for enhancing security in distributed application is proposed, this is a combination of the ‘hard’ and ‘soft’ trust relationship. The proposed hybrid trust model is a combination of hard and soft trust relationship. Hard trust denotes the trust relationships that can be derived from the underling cryptography based security mechanism, such as digital certificates, signatures. These trust relationship indicate one agent host’s belief in other in term of authentication of the relevant host’s identity. Soft trust is based on trust relationships derived from localization and external observations of system entity behavior, through social control mechanisms such as direct experiences, recommendations or a combination of both. In [23] the security model using trusted computing platform and trusted computing technology is proposed to improve they mobile agent security.

In distributed system the security is a major issue with the increase in market of electronic commerce and host foreign applications which are charged for their resource computation become less reliable and secured. In distributed system security management is very difficult where operating system doesn't support the distributed security. In a global open environment and electronic commerce every agent is exposed, so an agent can easily be victim of attack by malicious hosts. Those distributed systems which are dependent on external support for security; it is very much difficult to secure agents from malicious host. In cluster based distributed system the clusters are exposed for public network, so it is a challenging task to secure these types of system. Multicasting approach is used in multi agent ecommerce system. Due to its complex structure and required good performance security with providing efficiency at same time is difficult.

3. Security Framework for Protecting Host

This security framework offers a way to exploit the MA paradigm in several application areas, by providing a secure and open infrastructure for both execution sites and mobile agents. This framework provides a hierarchy of abstraction localities suitable for modeling the internet scenario; agent execute in place that represent physical nodes and can be grouped in domains abstraction that represent LANs possibly interconnected via gateway abstraction. In addition it also supports a naming infrastructure capable of finding mobile entities independently of their current location. It addresses, as distinctive feature, the most important security issues that emerge in the context of real MA applications running on internet.

To achieve security, this framework supports flexible security policies to govern the interactions of agents with both other agents and with the available resources in the execution site. The definition and enforcement of appropriate security policies can only be proceed after a precise identification of the principals.

The several different principals:
- **Agent** that is the mobile agent performing computation at different places;
- **Place** that represents the site responsible for agent execution;
- **Agent Creator** that represents the person / organization that implements the agent code;
- **Agent Owner** that represents the person / organization that executes the agent code;
- **Place Creator** that represents the person / organization that implement the agent execution environment code;
security framework for agent based distributed system

- Place Owner that represents the person / organization in charge of managing the place that hosts agent execution.

Any recognized principle owns specific temper

proof credentials needs for authentication and authorization. The unique identities of agent / place creators and owners are associated with X.509 certificates that bind a physical identity to a cryptographic key pair in a secure way. Thus, any cryptographic operation can be traced back to its performer with an immediate paternity responsibility.

With regards to agents, each agent carries a set of exclusive credentials as part of its state. Credentials bind agents to their creators and owner and are used as proofs that agent behave accordingly to the intentions of their creators and are correctly delegated by their owner to execute on their behalf. An agent / place creator should define his terms of liability for the behavior of implemented code, thus becoming responsible for any malfunction within these terms. In addition, agent owners should provide a proof of their agent execution to permit to enforce resource access control and to perform service accounting.

The domain defines a global security policy that imposes general authorization and prohibitions; each place can only apply restrictions to the domain −level set of permissions.

As shown in Figure 1 the security infrastructure is composed of:
- A Policy server for managing domain policies; only administrator with the necessary privileges can use the policy editor to activate / de-activate the policies at domain level
- A Domain server is in charge of maintaining consistent reference to the resource visible in the domain.
- An Authentication server for authenticating users, agents and execution sites.
- An authorization server for permitting access to resource.

This framework supports both push and pull model for initial propagation and update of domain policies. In any domain one Policy Server is in charge of storing the domain policies. When a new place enters the domain, it may acquire the domain policies by sending appropriate mobile Policy Agents to the Policy server for policy retrieval and successive installation. One approach to policy update is to rely on a policy monitoring agent running on the policy server targeted at monitoring policy status; if any policy change occurs, the policy monitoring agent is in charge of coordinating the propagation of the update to the places addressed by changing policies by creating several mobile policy update
agents. The policy update agents interact with domain server to discover the current location of places. The goal of policy update agents is to perform the update of policies. If the task fails for some reason, for instance when a place is temporarily unreachable, a policy update agent can behave differently to recover from the failure: it can either notify the policy mobile agent of the error or figure out the origin of the problem or wait until the failure is recovered and then immediately propagate the policy changes.

With regard to credential management, agent and host owners join the domain through a specified procedure that provide them with the set of necessary credentials. First owners register offline to the domain in order to be identified and to be assigned to their access rights, then acquire their credentials through an online certificate process. Authentication and authorization are implemented as distributed services and are enforced at both and host level. At the place level, agents are permitted to access resources on the basis of the intersection between domain and host policies. When an agent is transferred from one host to another host, the authorization server checks the agent credentials and access rights for that agent has access rights then authorization server allows the agent to execute their code on that host. For security and integrity of agent code is encrypted and digitally signed by the current host. The destination place accepts the incoming agent only if it is trusted and verified the authenticity on the basis of current site credentials.

Hosts are exposed to various attacks because they have to support the execution of agents coming from generally unknown and untrusted sites. This architecture protects hosts against potentially malicious agents by supporting agent authentication and authorization at both domain and place level. Agents are authenticated on the basis of owned credentials. At agent launching, the agent owner is asked for instantiated to start its migration, it is associated with its owner credentials. Agent initial state, unique identifier and code are digitally signed by the agent owner so that the computed agent credentials cannot be misused on other owner.

When agent is loaded to be executed at remote host, its credentials are verified in terms of authenticity. The agent owner signature is firstly extracted and verified. Once authenticated, agents are authorized to interact with the resource on the basis of the current security policy. In addition agents can also be authorized on the basis of their preferences. Agent preferences are a set of protected information that includes security policies expressed by the agent owner. On the one hand, an agent owner should be able to securely specify the tasks to delegate to his agent depending on the application contexts or the level of security to be ensured by agent computation. On the other hand, the hosting execution site can decide to satisfy or reject the policies contained in the agent preferences. When an agent enters a place, it is first assigned to a protection domain before starting execution.


A self executing security schema built for agent security should allow the agent to manage its own security related functions effectively without having to depend on the agent server for resource. The self executing security schema is a schema under which the agent is programmed to carry its own security implementation, as a scanning algorithm.

Step 1: Agent execution starts.
Step 2: Scan code elements and store values.
Step 3: Compare scanned value against code map supplied by home base.
Step 4: Continue with business processing statements.
Step 5: If Agent is awaiting resource or default time period for scan reached scan current state of code elements, Compare with values obtained from previous scan IF MISMATCH Set ERROR flag GOTO Step 7.
Step 6: Continue with business processing.
Step 7: If business processing complete performs final scan check.
Step 8: If ERROR flag set. Return to Agent Home Base Else Continue to next server in the agent itinerary.
This scans the agent code and verifies the integrity of the agent at random intervals. There are two aspects to the implementation of this schema. The first is the scan algorithm and the second is hiding the algorithm within the business function of the agent. To implement the scheme, the agent’s code is divided into different virtual zones. On arriving at a particular remote host, the agent reads its own binary images of its own code form these different virtual zones. The creation of zones is handled internally by the agent and is transparent to the rest of the schema. The values returned by this scan are compared against a code map carried by the agent. This code map is provided to the agent by its creator and is protected using a digital signature. Every scan of the agent code is compared against the code map. If the server at which the agent is docked is malicious and attempts to modify the agent code, in order to subvert or to manipulate the agent’s behavior, the agent is able to detect these attempts. By re-executing the scan algorithm at various random intervals, the agent is able to verify its integrity at run time, without having to rely on the host server for any additional resources such as execution of cryptographic routines to perform security checks. This schema does not require any external condition for execution.

5. Implementation

To validate the security framework for protecting host, I develop the small local area network in which there is a domain server who managing all resources availability and visibility on domain, a policy server who managing the policies at domain level, a authentication server which is responsible to authenticate the agent / hosts and maintaining a list of authentic agents / hosts, a authorization server which is responsible for authorizing the resource to specific agent or host.

As shown in figure 2 there is a container which contains the senders IP, agent and destination host IP. This container used to execute the agent on its destination host.

As shown in figure 3 when agent is loaded to be executed at remote host, its credentials are verified in terms of authenticity. The agent owner signature is firstly extracted and verified. Once authenticated, agents are authorized to interact with the resource on the basis of the current security policy. In addition agents can also be authorized on the basis of their preferences. Agent preferences are a set of protected information that includes security policies expressed by the agent owner. On the one hand, an agent owner should be able to securely specify the tasks to delegate to his agent depending on the application contexts or the level of security to be ensured by agent computation.

The prototype implementation of Self Executing Security Schema currently gives the agent an option of either initiating the security check or altogether skipping it. In trusted domains, the security schema can be suspended while the agent continues with its normal processing. While travelling through a non-trusted domain; the security function can be activated. The algorithm described in section 3, illustrates the schema in operation from the point the agent arrives at the
agent server. On initiating the security option the agent will perform a scan of its code. After each scan
Depending on their business functionality and resource requirements agents are moved between various agencies and hosts. While on one hand, this flexible mobile architecture allows the agent to react in various ways while attempting to meet its business requirements.
Unfortunately, this feature also makes the agent a target for malicious attacks. This schema allows an agent to scan its code at any particular instant of execution. The scan returns a list of agent code element values. These results are compared against the code map assigned to the agent at its home base. The MAS creating the agent digitally signs the code map. This code map refers to static chunks of agent code that are not intended to change during the course of its execution.

6. Conclusion

Despite the advantages offered by MA systems, a wider diffusion of this technology is limited by the lack of comprehensive security framework suitable to address the protection of both agents and hosts without introducing significant performance constraints. A challenge is to build security mechanisms that do not limit the features of mobile agents, such as autonomy and efficiency. These considerations have been taken into account in the development of the security framework that supports the protection of host as a distinctive feature. Efficiency and scalability have been key driving factors in the design of security solution. The advantage of the self-executing security schema for agent is the self-reliance and independence it gives to agent with respect to its security function. Further, the simplicity and openness of this schema encourages agent servers to accept agents rather than view it suspiciously and block its action.

References

[1] Leander Conradie, Maria-Athina Mountzia
Department of Computer Science Technical University of Munich
Arcisstr. 21, 80333 Munich? Germany

[2] F. Ahmad, N. Santoro SCS, Carleton University, Ottawa, ON K1S 5B6, Canada
A. Nayak SITE, University of Ottawa Ottawa, ON KIN 6N5, Canada
N. Goel, A. Patnaik, A. Srivastava Tecsis Corporation Ottawa, ON K2E 7L5, Canada

[3] Toshihiro TAKETA Depart of Electrical and Information Engineering Faculty of Engineering Yamagata University
Hirosi NUNOKAWA Faculty of Software and Information Science Iwate Prefectural University

Department of Computer Science, University of Brasilia,


[7] Partha Sarathi Mandal and Krishnendu Mukhopadhyaya

[8] Pradeep Kannadiga and Mohammad Zulkernine
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)

[9] Sartid Vongradhiph, Ph.D.
Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University. Wichet Plaimart Digital System Engineering Lab. (DSEL) Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University.

[10] Luciano Serafini ITC – IRST 38050 Povo, Trento, Italy
Chiara Ghidini
Department of Computing & Mathematics
Manchester Metropolitan University

CoCo Software Engineering GmbH
Margaretenstr. 22/9, 1040 Vienna, Austria
Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID’02)

[12] Hartmut Vogler, Thomas Kunkelmann, Marie-Louise Moschgath
Darmstadt University of Technology
Information Technology Transfer Office
Alexanderstr. 10
Proceedings of the 1997 International Conference on Parallel and Distributed Systems (ICPADS ’97)

[13] Nicholas Yiakelis Morris Sloman
Department of Computing, Imperial College
180 Queen’s Gate, London SW7 2BZ, United Kingdom,
Proceedings of the 1996 Symposium on Network and Distributed System Security (SNDSS ’96)

[14] Paulo Jorge Marques*, Luis Moura Silva, João Gabriel Silva
Departamento de Engenharia Informática, Universidade de Coimbra, Portugal
Security framework for agent based distributed system

Departamento de Engenharia Informática, Universidade de Coimbra, Portugal

[16] John Page, Arkady Zaslavsky, Maria Indrawan
School of Computer Science & Software Engineering, Monash University, Melbourne

[17] Makan Pourzandi, David Gordon,
Open Systems Laboratory, Ericsson Research
8400 Decarie Blvd, Town of Mont-Royal, QC, Canada
William Yurcik, Gregory A. Koenig
National Center for Supercomputing Applications (NCSA)
University of Illinois, Urbana-Champaign, USA

[18] Tieyan Li, Kwok-Yan Lam
Centre for Systems Security
School of Computing
National University of Singapore

[19] Antonio Corradi, Rebecca Montanari
Cesare Stefanelli
Centre for Systems Security
School of Computing
National University of Singapore

[20] H. Reiser, Munich Network Management Team
University of Munich, Germany
G. Vogt, Munich Network Management Team
University of Technology, Germany

[21] Hédi HAMDI and Mohamed Mosbah
Université de Bordeaux, LaBRI
351, cours de la Libération F-33405 Talence cedex, France
ADEL Bouhoula, Sup’COM Tunis
Rte de Raoued Km 3.5 2083, Ariana Tunisia

[22] Ching Lin and Vijay Varadharajan
Department of Computing
Macquarie University, Sydney, Australia

[23] Zhidong Shen, International Software School
Wuhan University
Xiaoping Wu, School of Computer
Wuhan University