Mobile Agent security using Proxy-agents and Trusted domains*

Nikola Mitrović
IIS Department, University of Zaragoza
Maria de Luna 3
50018 Zaragoza, Spain
mitrovic@prometeo.cps.unizar.es

Unai Arronategui Arribalzaga
IIS Department, University of Zaragoza
Maria de Luna 3
50018 Zaragoza, Spain
unai@posta.unizar.es

ABSTRACT

Commercial or wide-network deployment of Mobile Agent Systems is not possible without satisfying security architecture. In this paper we propose architecture for secure Mobile Agent Systems, using Trusted Domains and Proxy agents. Existing approaches are based on security services at the level of an agent system, library or specific objects. Our concept uses proxy agents to enable transparent security services both to security-aware mobile agents and legacy agents. Per-agent and domain-level security is provided. Proposed concept can be used with non-compatible environments and legacy systems.

Keywords

Mobile Agents, Security, Proxy Agents and Trusted Domain.

1. INTRODUCTION

Commercial or wide-network deployment of Mobile Agent Systems is not possible without satisfying security architecture. This paper outlines a design for a secure mobile agent architecture.

Mobile agents and mobile agent platforms are exposed to various security threats. Attacks on mobile agents and platforms usually come in two main forms: active and passive [8]. While passive attacks try to collect data without authorization (e.g., eavesdropping), active attacks try to modify system and cause different behavior of system. The trust [2] in mobile agent systems plays an important role. By establishing a trust relationship mobile agents can gain access to resources, perform specific actions or delegate their rights.

Existing solutions are focused on several approaches. Usually, the proposed solution is some kind of library [4] or service [1] that provide security mechanisms for mobile agents and mobile agent systems. Many of security problems are resolved by using Public Key Infrastructure (PKI) [9]. Some approaches uphold "smart objects" (that are self-aware) [10], or security agents that provide secure communication [6, 14, 15]. In addition, some authors as in [16], create specialized agents ("privacy guardians") that are meant to protect the data and communication of agents. Trust solutions [11, 12, 7, 4] are mainly focused on how to delegate and negotiate trust between systems or agents.

In this paper we propose architecture for secure mobile agent systems, using trusted domains [3] and proxy agents [13]. We propose usage of a proxy agent paradigm for security services together with trusted domain and directory services for rights and authenticity distribution. Specialized *Security Proxy Agents* are used to provide security mechanisms to both mobile agent systems and mobile agents. This concept enables security-context for legacy systems, simplifies development of the agents and provides both domain-level and per-agent security. In addition, proposed architecture gives possibility of protecting the devices that does not have sufficient processing power (e.g. wireless devices).

This paper is organized as follows: in Section 2 we present proposed architecture. Sample scenarios are discussed in Section 3. In Section 4 we present conclusions and future work.

*This work has been supported by the spanish "Comisión Interministerial de Ciencia y Tecnologa" (CICYT), project TIC2001-1819.

2. PROPOSED SECURITY ARCHITECTURE

Security is a delicate issue. As the system is more secure, it gets more difficult to build, more complex to maintain. Complex systems with more components have higher possibility of failure or breach; on the other side, too simple systems can be vulnerable.

2.1 Security Proxy Agents and Trusted Domains

Having this as an idea, we propose architecture that eliminates certain aspects of complexity. We introduce *security proxy agents* as facilitators of security services for mobile agents and mobile agent systems. Notion of proxy-agents is not new [13]. Many authors used proxy-agents as agents that help other agents to do something, or to do something on the behalf of other agent [13]. Security proxy agent is mobile agent that provides security services to both agents and/or agent systems. This agent contains extensible set of security and cryptographic mechanisms that can be used by agent systems or agents autonomously. In addition, these specialized agents contain set of automatic actions that are transparently performed upon agents and agent systems. Each mobile agent system have one *proxy factory* that creates and associates the agents with the security proxy agent created within factory. Also, system assigns one or more security proxy agents to guard the "entrance" to the system. These security proxy agents check all incoming and outgoing agents in order to apply adequate trust policy and security checks. In addition, security proxy agents can be extended to support special requirements of some systems.

Our architecture relies on the concept of Trusted Domains. Fig. 1 shows proposed security architecture organized organized as trusted domains. We can see that every

domain has one or more places (agent systems) that deal with security.

One domain has responsibility to authenticate agents and agent systems, and to apply appropriate trust policy. Once in the domain, the agent can travel freely without any further security checks, since it is considered trusted. Local access restrictions are applied (the user may not be willing to share some of the resources with others). Exceptionally, additional tests can be forced, and agents or agent system can require additional services from security proxy agent.

If the agent is member of more than one domain, malicious agents could enter from another domain. In this case, trust relationship must be established between domains, and such agent system should have installed proxy factory in order to check and apply adequate trust policy for incoming agents from different domain.

2.2 Security Proxy Agents' operations

Security proxy agents perform several transparent functions. By checking the agents' signature [15] and by encrypting it, they provide secure transport and identification of the agent. Also, using agents' signature or the signatures of the agent systems' modules, the alteration of agent or system code is automatically detected. Similar actions are done on the security proxy agents to ensure authenticity and non-alteration. Other security or cryptography services that agents can request such as state appraisal [5] or transaction logging [8] are provided by security proxy agents upon request. Number of security functions supported by security proxy agents is extensible as some systems may provide or require additional security mechanisms. Agent systems also enjoy transparent trust and security verification of the incoming and outgoing agents, and if needed, can request additional services from security proxy agent.

Legacy agents, or agents that are not aware of security context will enjoy transparent services of security proxy agents. This leads to faster and easier development of mobile agents that do not require some specialized levels of security.

This architecture is built with public systems in mind. It is not focused on how to solve some of the specific attacks on agent or host, but on the architecture of a system that will include the features of the known solutions, and expose them in more transparent and efficient manner to the system. Proposed architecture can be applied on the public systems such as Internet Service Providers (ISPs), or the systems that require some levels of security, such as Local Area Networks (LANs).

3. SAMPLE SCENARIOS

In this Section, we present common scenarios that occur within this security model.

Let us suppose that one agent from the home-system A wants to travel to the remote-system B. From the Figure 1 we can see that Home-system A belongs to the domain D1 and remote-system B to the domain D2.

As the agent (from the system A) is in its home-platform, the agent will move to its domain controller (DC1). The proxy factory service located at domain controller will create security proxy agent that will be assigned to our traveller agent, and will equip it with agent's credentials. The agent itself do not have to be aware of this process. Security proxy agent will perform the signing and optionally enveloping of the agent-traveller. Then, agent-traveller and security proxy

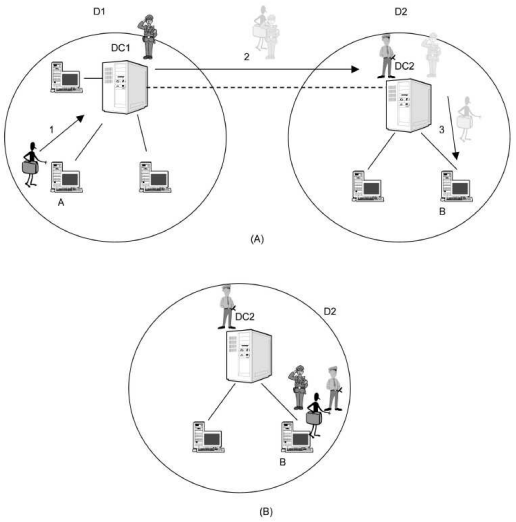

Figure 1: Sample scenario – agent trajectory.

agent will travel to the domain controller (DC2) of the destination domain D2. Upon arrival at domain controller DC2, security proxy agent will check the alteration of the agent-traveller and itself. If there is no alteration detected, security proxy agent of the agent-traveller and security proxy agent of the domain controller DC2 will negotiate possibilities of cooperation.

If the cooperation is possible and the security requirements are met, the traveller-agent will be prepared for execution (e.g., decrypted). Then it will continue its journey to the remote-host B. Once in the domain, as described in Section 2, the traveller-agent can run without limitation and within his environment, without any needs to be decrypted, checked or bounded in any way, except for the current host access privileges (sandbox). The security proxy agent will remain at the domain's entrance (DC2), waiting for agent to finish its journey at domain. This can be suitable for devices that do not have sufficient processing power, such as wireless devices. If the agents are trusted by the domain, it can travel to any mobile device within the domain without having to perform security or trust negotiations.

In case that agent-traveller needs some extra security operations, like transaction logging, encryption or signing, the traveller-agent will call its own security proxy agent to assist him (see Figure 1(B)). Similar behavior will occur if the agents from systems B want to use some security operations. In this case, domain D2's controller (DC2) will use its proxy factory to create specialized security proxy agent that will assist mobile agents that are "owned" by domain D2; in this case, for the agents from system B.

Upon completion of its journey on the domain D2, traveller-agent will meet once again with its security proxy agent, and upon authenticity and alteration check, the traveller-agent will be returned to "safe to transport" mode, and the agents will continue their journey to another platform. Similar behavior will be exercised on the mobile agent systems.

We examined normal operation of the system. However, we expect that the agents and systems are exposed to attacks. Here, we will discuss some of the situations when agents and agent-systems are malicious.

If the malicious agent is launched from the very domain, this agent can do harm only to a limited number of hosts. This kind of agents will be detected first time they meet with the domain controller, or a security proxy agent. The malicious agent will be detected, and the agent origin will be tracked. Alternatively, every agent that is launched from one system can be forced to pass trough domain controller, where it could be checked and approved. However, our approach is based on the idea that one domain is internally safe, so this kind of a measure is considered as unnecessary. Also, if some of the systems detect an malicious agent or vice versa, these systems or agents can be easily tracked and eradicated from the domain. The possibility of malicious agents performing unauthorized actions is also limited by the sandbox mechanism that is used on every platform, as described earlier.

Similar situations will occur in the malicious mobile agent system scenario. Tampered agents will be detected as soon as they arrive on domain controller, or perform an operation with security proxy agents. The malicious host will be detected and eradicated.

Legacy agents that are not aware of the proxy agents will be transparently processed by proxy agents. Agent systems that are not familiar with security proxy agents will treat them just as ordinary agents. Therefore, some levels of security will be conserved for legacy agents and systems.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a security architecture that uses security proxy agents and distributes security over trusted domains. The main features of this approach are:

- This architecture uses known solutions to security problems (known mechanisms).
- Proxy agents are used to provide security functions to both agents and platforms.
- Security proxy agents can be extended to support additional (and specific) features.
- Security is distributed over trusted domains, which facilitates management and trust tasks.
- This architecture supports legacy agents and systems.
- This concept can enable security context for devices that cannot perform security computation (e.g, mobile devices).
- Security proxy agents act transparently (easier agent development).

Our future work will be focused on implementing proposed concept and in implementing autonomic logic to the prototype. Also, as a continuation of this work, some form of distribution and caching of certificates for mobile agents should be investigated.

5. REFERENCES

[1] J. Bacon, K. Moody, and W. Yao. Access control and trust in the use of widely distributed services. *Middleware*, 2001.

[2] M. Blaze, J. Feigenbaum, and A. D. Keromytis. The role of trust management in distributed systems security. In *Secure Internet Programming*, pages 185–210, 1999.

[3] B. Crispo. How to build evidence in a public-key infrastructure for multi-domain environments. In *Security Protocols Workshop*, pages 53–65, 1997.

[4] Q. He and K. Sycara. Towards a secure agent society. *ACM AA'98 Workshop on Deception, Fraud and Trust in Agent Societies*, 1998.

[5] F. Hohl. A framework to protect mobile agents by using reference states. In *International Conference on Distributed Computing Systems*, pages 410–417, 2000.

[6] F. Hohl and K. Rothermel. A protocol preventing blackbox tests of mobile agents. In *ITG/VDE Fachtagung Kommunikation in Verteilten Systemen (KiVS'99)*. Springer-Verlag, Berlin Germany, 1999.

[7] Y. J. Hu. Some thoughts on agent trust and delegations. *The Fifth International Conference on Autonomous Agents*, May 28-June 1 2001.

[8] W. Jansen and T. Karygiannis. Nist special publication 800-19 - mobile agent security. *NIST*, 2000.

[9] U. Maurer. Modelling a public-key infrastructure. In *ESORICS: European Symposium on Research in Computer Security*. LNCS, Springer-Verlag, 1996.

[10] C. Meadows. Detecting attacks on mobile agents. *Foundations for Secure Mobile Code Workshop*, pages 64–65, March 1997.

[11] L. Moreau. A Fault-Tolerant Directory Service for Mobile Agents based on Forwarding Pointers. In *The 17th ACM Symposium on Applied Computing (SAC'2002) — Track on Agents, Interactions, Mobility and Systems*, Madrid, Spain, Mar. 2002.

[12] C. Ono, D. Kanetomo, K. Kim, B. C. Paulson, M. Cutkosky, and C. J. Petrie. Trust-based facilitator for e-partnerships. *Fifth International Conference on Autonomous Agents (AGENTS'01), AAAI*, pages 108–109, May 2001.

[13] S. Papastavrou, G. Samaras, and E. Pitoura. Mobile agents for WWW distributed database access. In *ICDE*, pages 228–237, 1999.

[14] V. Roth. Mutual protection of co-operating agents. In *Secure Internet Programming 1999*, pages 275–285, 1999.

[15] V. Roth and V. Conan. Encrypting java archives and its application to mobile agent security. In *AgentLink 2001*, 2001.

[16] R. Serban and R. van de Riet. Enforcing policies with privacy guardians. *SEMAS*, 2001.