A Novel Solution of Mobile Agent Security:Task-Description-Based Mobile Agent

Haiyan Che, Dali Li, Jigui Sun, and Haibo Yu

College of Computer Science and Technology, Jilin University, Changchun, 130012 China (Symbol Laboratory)

Summary
A new definition and realization of mobile agent was advanced: task-description-based mobile agent, which is a kind of data package describing the tasks needed to perform on the agent platforms. A secure mobile agent system building on this kind of mobile agent was proposed. By way of using Proxy/Manage Agency to create and dispatch agents, Java 2 security model to protect local resources and the trust model to manage and adjust the trust relationships between the platforms the system provides a prior protection for agent platforms and a posterior detection of attacks for the agents.

Key words:
mobile agent, security, task description, Proxy/Manage Agency, trust model

1. Introduction

Mobile agents are autonomous software entities, which can migrate through a network of heterogeneous sites to perform tasks on behalf of their owners. The characteristics of mobile agents, mobility, activity, autonomy and adaptability, make them ideal for many applications. However, one of the main concerns currently impeding the wider acceptance and use of mobile agents is the issue of security.

Although a number of viable security techniques have already been developed, each of them has its shortcomings. The main difficulty stems from the definition and realization method of mobile agents. We proposed a new definition of mobile agent: task-description-based mobile agent and discussed its rationality. Furthermore, we designed a mobile agent system building on this kind mobile agents—Task Description Based Mobile Agent System (TDBMA) and analyzed its security. In the TDBMA system the agents are realized as a data package in XML, describing the tasks needed to be performed, as users required, on platforms. The Proxy/Manage Agency, a special agent platform, dispatches agent uniformly. Other agent platforms use Java 2 security model to protect their local resources. And a trust model is used to adjust the trust relationship between the platforms. To be distinguishable, the classic mobile agents accepted by most people are named mainstream mobile agents and the new definition advanced in this paper is called task-description-based mobile agents.

The rest of this paper is organized as follows: Section 2 describes the existing security solutions for the mainstream mobile agents, while within Sec.3 a new way to solve the security issue is introduced. The noteworthy thing is the novel definition of mobile agent and its rationality. Sec.4 depicts the TDBMA system and the security of this system is discussed in Sec.5. Sec.6 provides an example of TDBMA system's implementation and finally, Sec.7 concludes the paper and gives an outlook to further research.

2. Security Solutions for the Mainstream Mobile Agents

Each of the current security solutions has its deficiency. For the countermeasures of agent platforms: Sandbox architecture [1] is too strict. Safe Code Interpretation [2] enables to analyze the safety of agent's instruction. But it is difficult to analyze the combination of instructions. Proof-Carrying Code (PCC) [3] is very complex to implement [4]. Path Histories [5]'s obvious drawback is that path verification becomes more costly as the path history increases and the technique is also dependent on the ability of a platform to judge correctly whether to trust the visited platforms identified.

For the methods to protect agents: Cryptographic traces [6] is not able to reveal the attacker. State appraisal mechanisms [7] can detect a particular manipulation but not the attacker. Tamper-proof-devices [8] are hardware-based and therefore not suitable in open systems. Computation with encrypted functions [9]'s serious drawback is ruling out any active mobile code that performs some immediate action on the host. VDOT [10] was advocated to solve this problem but with the drawback of no automatic circuit generator. Code scrambling [11] requires numerous interactions with a trusted host at any hop on the agent's journey. An optimistic third-party protocol [12] proposed by D. Westhoff is not able to detect the originator of all kinds of active attacks, or detect whether a working context really started an agent. Paper

Manuscript revised February 8, 2006.

[13] described a novel Remote Distributed Scheme (RDS), which can protect any mobile computation, in any environment and for any required level of secrecy. But it needs to meet some assumptions and conditions.

Each of the methods above has its shortcomings. We can say that there is no very good solution to provide adequate protection for both the agents and agent platforms. By analyzing thoroughly we conclude the main reason lies in the definition and implementation method of the mainstream mobile agents.

3. Novel Methods to Solve the Security Issues

3.1 Define and Use the Task-Description-Based Mobile Agents

Mobile agents are defined and implemented as a special data package, including the description about the tasks needed to be performed on the host in order to satisfy the user's request. However, there is no detail about how the tasks should be performed. It's due the host to decide whether and how to provide the corresponding service. The task-description-based mobile agent has the following merits:

First, it can carry out tasks as the mainstream mobile agents can do and has the characteristics of mobility, autonomy, activity and adaptability. As long as the corresponding agent platform can receive, interpret, provide service and send this kind of agent and the necessary information are all included in its data package the effect of this task-description-based agent is same as the mainstream agent. All we need is to construct the collaborative platform. Second, it meets the ultimate evolution direction of the mainstream mobile agents, which is formalizing the agents to analyze them and thus offer appropriate protection to the platforms. Third, it can solve the difficult security issues well. The use of the task-description-based mobile agents eliminates the problems of agent attacking platform and agent-agent attacking radically. For agent, which is data package passing in the network, we can use cryptography to protect the content and detect malicious behaviors.

To sum up, the task-description-based mobile agents don't violate the essence of mobile agents and the advantages are easy to implement and easy to solve the security problems.

3.2 Other Facilitated Methods

Use the Proxy/Manage agency to dispatch agent. To solve the problem of unknowing the origin of mobile agents we proposed to use the Proxy/Manage Agency known by other platforms, which is the only platform with the ability of constructing and dispatching agents and managing the information of users, agents and all the platforms. This can help other platforms to authenticate and authorize the incoming agents.

Use Java 2 security model to protect local resources. With the usage of the task-description-based mobile agents what the platform need do is to be careful not to provide the agents service that will do harm to itself. So it can use the Java 2 security model to define permissions to specify the authorized access to a particular resource and security policy to assign permissions to incoming agents, which can restrict the platform to provide services to agents and protect local resources.

Use a trust model. Security issues in mobile agent system can not be resolved by pure technical methods. Some methods to penalize the malicious behaviors are needed. By using a trust model based on trust relationships to maintain and adjust the trust values between the entities of the system we can deal with the security problems, from a non-technical point-of-view, which are irresolvable by only technical methods.

4. TDBMA System

According to the methods above we designed a task-description-based mobile agent system (short for TDBMA system), which mainly deals with the resources maintained by agent platforms. The architecture is described in Figure 1. In TDBMA system there are two kinds of agent platforms: Proxy/Manage Agency and Mobile Agent Oriented Servers (short for MAOS). The Proxy/Manage Agency, as discussed above, manages all the entities in the system, creates agent by user's request and dispatches them to MAOSs. Finally, it can extract result from the returning agent and reply to the user. The agents in the system are task-description-based mobile agents, which include the description of the tasks user requested. MAOSs are usual agent platforms owning some resources and the ability to receive agents, provide services and send agents.

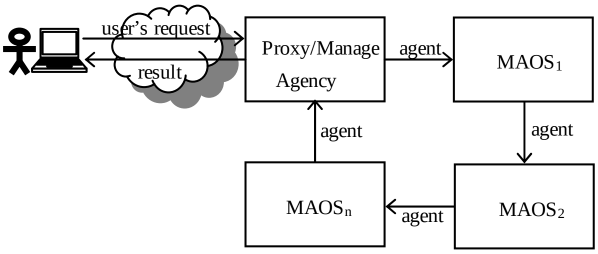

Fig. 1 Architecture of TDBMA system.

4.1 Definition of the Task-Description-Based Mobile Agents

The task-description-based mobile agent in the TDBMA system includes all the information needed to carry out tasks user required. On the basis of agent security model in Ajanta system [14] we divided the agent's state into two parts:

- ReadOnly part: includes the information about the agent's identity, task description and the agent's route. All of these are constant during the agent's travel. Entities in the system can only read but not change them. Any tampering with this part can be detected. The Proxy/Manage Agency will sign this ReadOnly part when producing a new agent and any platform on the agent's path will first verify the signature on receiving it before providing any service. If verify fails, the platform would do nothing but adjust the trust value to the previous malicious host.
- AppendOnly part: includes the data collected by agent from the sites it visits. Agent platforms are allowed to add content to this part, but not to make any subsequent modification. This part will be initialized by the Proxy/Manage Agency when creating a new agent. The MAOSs on the agent's route should update this part, such as adding result entry, self identity, signature, updating checksum and so on. The Proxy/Manage Agency will decode this part of the returning agent to extract the results and detect whether there are any malicious platforms.

4.2 The Proxy/Manage Agency

The structure of the Proxy/Manage Agency is illustrated in Figure 2. The key lies in the agent module which can create agent and the result module which can extract results. Both of them needs the help of the security module, which has standard cryptography functions and manages public key certificates of other MAOSs. The function of the xml module is parsing the agent, which is to transform the agent between the xml data and the java object by using the data-binding function of XML.

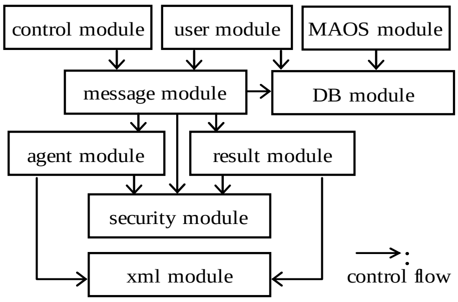

Fig. 2 Logical structure of the Proxy/Manage Agency.

4.3 The MAOS

As another kind of agent platform, MAOS maintains some resources and provides service for agents. Its structure is depicted in Figure 3. Here, the agent module can not create agent but can receive and dispatch agent. The security module can verify the digital signature of agent's ReadOnly part and sign the result item obtained from this MAOS. More importantly, the security module should assure the safety of local resource by inspecting the operations of every module according to the local security policy. The request processing module can read the task description carried with the agent and provide certain services. Moreover, this module can support different underlying resource servers.

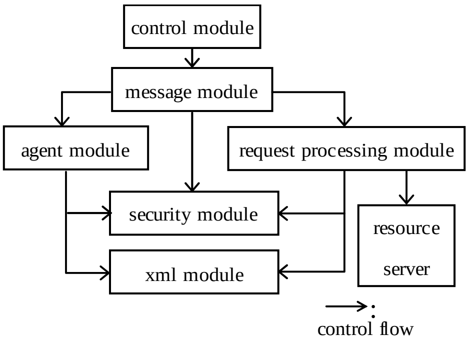

Fig. 3 Logical structure of the MAOS.

4.4 The Trust Model

The trust model in TDBMA system was based on the ones in [15, 16]. Every agent platform has a trust value table, in which there are, for each of other platforms, a trust value and the number of malicious behaviors having detected by current platform. The trust value table can be updated dynamically by some algorithm and can guide platform's cooperation with others.

5. Security Analysis of the TDBMA System

5.1 Security of the Platforms

The platforms in the TDBMA system are safe. For the agents are a kind of static task-description-based data package instead of autonomous program, the problem of agent attacking the platform disappears thoroughly. By using the Java 2 security model the platform can restrict the services provided to agents and protect local resources. Malicious platforms can attack other platforms by "replay", which is sending multiple copies of the same agent to some platform repeatedly. This kind of attack can be detected by storing the record about the agents having been received at the platform (for example the agent's identity and the information about the execution). Once replay attack is detected, the current platform would lower the trust value to the malicious one and would not receive any agent from it if the trust value were lower than the trust value threshold. The trust value may be then recovered after related entities having negotiated and resolved the problem.

5.2 Security of the Agents

The agent in the TDBMA system is a data package describing the user's task. The possible attacks from malicious platforms are: (1) tampering with the ReadOnly part; (2) tampering with the AppendOnly part; and (3) interfering with the agent, for example, discarding the agent without sake. The attacks of the first two kinds can be detected by verifying digital signature and checksum. Once detected the trust value to the malicious hosts would be lowered. For the third attack, the result is the agent not being able to return back to the Proxy/Manage Agency for ever. When the Proxy/Manage Agency discovers that there are agents not returning back by time of their life deadlines, it would adjust the trust values to all the platforms in the agent's route. No agent would be sent to the platform if the trust value to which is lower than the trust value threshold.

The conclusion is that the TDBMA system can provide a prior protection for agent platforms and a posterior detection of attacks for the agents. With the help of trust model adjusting the trust values between platforms the TDBMA system can run safely.

6. An Implementation

We have implemented a TDBMA system in Java language, in which the agent can access the file resources maintained by MAOSs according to the user's request. The task-description-based mobile agent is implemented as an XML file with all the fields needed. Used the IBM's XML security suite [17] to sign and verify, the Exolab.org's tool: Castor [18] to transform the agent between the XML data and java object. Realized the Proxy/Manage Agency and the MAOS. Defined new security manager such as AgentSecurityManager and some new permissions to protect local resources. The trust value adjusting algorithm is: $T(A,B) = t \times a^p$, in which $T(A,B)$ denotes the trust value of A to B; a is a real number in(0,1); p is a nonnegative integer denoting the amount of malicious behaviors of B; t is a real number in[0,1],whose value can be chosen conditionally. For example, a smaller t can be used when malicious platforms having been detected for sure; otherwise a larger t for all the suspicious ones. A smaller t is used when the malicious behavior is serious and otherwise a larger t. It is obvious that the choice of t deals with the problem of unable to reveal the exact culprit.

7. Conclusions

This paper advocated a novel understanding and definition of mobile agent: a data package describing the tasks user required and proposed the security architecture of TDBMA system. In the TDBMA system, the task-description-based mobile agents are used to behave on behalf of users. The Proxy/Manage Agency is responsible to create and dispatch agents to avoid any individuals sending agents. The agent platforms use the Java 2 security model to protect the local resources. And the trust model, from a system viewpoint, guarantees the system running safely by a non-technical way. All the measures above provide a comparatively good solution for the security issue. At last, we implemented a TDBMA system capable of searching files in ftp servers, which verifies the feasibility of task-description-based mobile agents and the TDBMA system. Compared with other mobile agents system the outstanding character of TDBMA system is the introduction of task-description-based mobile agent and the advantage is that the system is easy to implement and the security issues are easy to resolve.

The weakness of the task-description-based mobile agents is not flexible enough to represent the agent's task easily as the mainstream ones can do. The main reason is that the agents are resources oriented, but there's no good method to define, describe and categorize all kinds of resources. So searching for appropriate resources-representing and flexible task-describing method is the emphasis of our future work.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No.60473003; Program for New Century Excellent Talents in University; the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20050183065.

References

[1] L.Gong. Java security architecture (JDK1.2) .Technical report, Sun Microsystems (1998)

[2] J. K. Ousterhout, J. Y. Levy, B. B. Welch: The Safe-Tcl Security Model.Technical Report SMLI TR-97-60, Sun Microsystems (1997)

[3] G. C. Necula and P. Lee.: Safe untrusted agents using proof-carrying code. In Mobile Agents and Security, volume 1419 of Lecture Notes in Computer Science (1998) 61-91

[4] George C. Necula, Robert R. Schneck.: A Sound Framework for Untrusted Verification-Condition Generators. In Proceedings of IEEE Symposium on Logic in Computer Science, LICS03 (2003)

[5] J.J.Ordille: When Agents Roam, Who Can

You Trust? Proceedings of the First conference on Emerging Technologies and Applications in Communications, Portland, Oregon (1996) URL:http://plan9.bell-labs.com/cm/cs/who/joann/ordille2.html

[6] G. Vigna: Cryptographic traces for mobile agents. Mobile Agents and Security, LNCS 1419, Springer-Verlag (1998) 138-153

[7] W. M. Farmer et al.: Security for mobile agents: Authentication and state appraisal. In Proceeding of 4th European Symp. on Research in Computer Security, LNCS 1146, Springer-Verlag (1996) 118-130

[8] U. G. Wilhelm.: Cryptographically Protected Objects. Technical report, Ecole Polytechnique Federale de Lausanne, Switzerland (1997)

[9] T. Sander and C. Tschudin.: Protecting mobile agents against malicious hosts. In Mobile Agents and Security, LNCS 1419, Springer-Verlag (1998) 44-60

[10] Sheng Zhong and Yang Richard Yang.: Verifiable Distributed Oblivious Transfer and Mobile Agent Security. Proceedings of the 2003 joint workshop on Foundations of mobile computing (2003) 12-21

[11] F. Hohl.: An approach to solve the problem of malicious hosts. In Mobile Agent Systems (1997)

[12] Dirk Westhoff.: An optimistic third party protocol to protect a mobile agent's binary code. International Journal of Software Engineering and Knowledge Engineering 11(5), (2001) 607-619.

[13] Asnat Dadon-Elichai.: RDS: Remote Distributed Scheme for Protecting Mobile Agents. Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1 (AAMAS'04) (2004) 354-361

[14] N. M. Karnik and A. R. Tripathi.: Security in the Ajanta mobile agent system. Technical Report TR-5-99, University of Minnesota, Minneapolis, MN 55455, U. S. A. (1999)

[15] R. Yahalom, B. Klein, and T. Beth.: Trust relationships in secure systems－A distributed authentication perspective. In Proc. of the 1993 IEEE Symposium on Research in Security and Privacy, (1993) 150-164

[16] H. K. Tan and L. Moreau.: Trust Relationships in a Mobile Agent System. In Proceedings of Fifth IEEE International Conference on Mobile Agents 2240 Springer (2001) 15-30

[17] IBM's security suite: http://www.alphaworks.ibm.com/tech/xmlsecuritysuite

[18] http://castor.exolab.org/download.html

Haiyan Che received the B.S. and M. E. degrees from Jilin Univ. in 2000 and 2003 respectively. Now is working as a teacher assistant (from 2003) in the College of ComputerScience and Technology, Jilin Univ.. Her research interest includes mobile agents, description logic and software engineering.

Dali Li received the B.E. and M.E. degrees, from Jilin Univ. in 1997 and 1999, respectively. After working as a teaching assistant (from 2002), he has been an instructor at Jilin Univ. since 2005. His research interest includes distributed object, mobile agents, software engineering.

Jigui Sun received the Dr. degree in Computer Science from Computer Science Department of Jilin University of China in 1993, and was promoted professor in July 199**7.** He is currently head of Ministry of Education Key Laboratory of Symbolic Computation and Knowledge Engineering and associate dean of College of Computer Science and Technology, Jilin University. He is director of China Computer Association and committee member of the sub-division of computer science and technology for guiding higher education of the Ministry of Education. He is the Ministry of Education's Distinguished Talent for the New Century. Sun's research work focuses on intelligent information processing.