International Journal of Latest Trends in Engineering and Technology
Vol.(11)Issue(2), pp.102-105
DOI: http://dx.doi.org/10.21172/1.112.18
e-ISSN:2278-621X

A COMPREHENSIVE APPROACH FOR SECURITY ASPECTS OF MOBILE AGENT SYSTEM

Arihant Khicha[1], Parveen Kumar[2]

Abstract. Mobile agent platforms are implementation environments for mobile agents on different computers, together with the guest platforms and home platform. A home platform of a mobile agent is in charge for creating, initializing, dispatching, receiving, and eliminating a mobile agent. Mobile agent research is generated by at least two different approaches, namely distributed artificial intelligence and distributed systems. Each of these play an important role in mobile agent research and therefore brings an exclusive perceptive and equivalent influence in the field. This paper, focused on the general issues in mobile agent security, paying special attention to the problem of malicious hosts. A classification of threats is given and some suggested solutions are examined in detail.
Keywords: Mobile agents, distributed artificial intelligence, security

1. INTRODUCTION

The first issue raised when designing a security framework is the kind of security policies that agents and hosts must support. One security requirement inherited from traditional distributed system security is isolation that user programs and agents must be protected from each other, and the host must be protected from agents. This is a requirement on the underlying agent architecture, and is no different to the security requirement of any system. In contrast to traditional programs, an agent is written to execute in different environments, and even to move during its execution. The owner of the agent can have different levels of trust in each host. An agent must therefore be adaptable to the environment in which it runs. This means that it must be programmed to respond to the differing trust levels of the hosts that it visits, and to adapt its defenses accordingly. For instance, a host may decide to encode a digital signature into its agent before sending it to another host, in order to authenticate that same agent when it returns. Similarly, an agent may decide to encrypt some of its data before moving to a less trustworthy host.

In the proposed framework, two further properties are defined for agent applications. The first is survivability. In the auction application for instance, it is especially important that a bid survive attacks. This means being able to replicate an agent and send the replicas on different itineraries. Replica results can then be voted upon. A further security property is believability. This means that there must be a way to verify the information furnished by an agent. A Hyper News agent for instance must prove that the contents it furnishes are the same as those published by the provider; a user must prove that the sum of money in his wallet is not forged. The auction example also shows a range of bindings that a bid agent has to prove to an auction server:

- Bid to user binding: in order to defeat masquerade attacks where an attacker forges a bid.
- Bid to quote binding: in order to detect attacks on the integrity of the bid's information by a competitor.
- Bid activeness: that a bid that presents itself is still valid, for instance that it has not received a kill signal from its owner.
- Bid to public key binding: that no revoke has been issued on the public key carried by an agent.

2. SECURITY REQUIREMENTS

Before considering the outcomes of this research effort, the objective and relevance of this work are revisited for a moment. The necessity for a framework can best be described by using an analogy to that of the processes involved in the building of a factory. The process of proposing a mobile agent security framework necessitates the establishment of criteria and subsequently a set of requirements to which the framework needs.

[1,2] NIMS University Jaipur, India

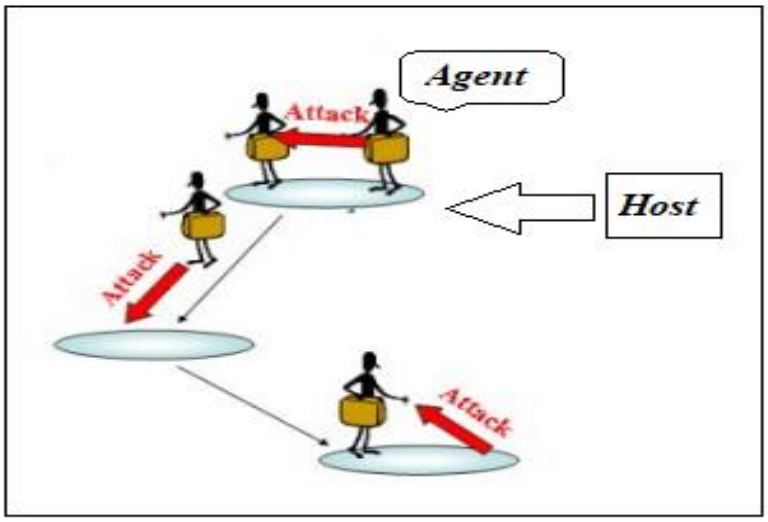

Fig 1: Mobile Agent Security

The criteria that protects a mobile agent against a malicious host is based on the fundamental concerns or requirements of users gaining access of computer network services, namely integrity, availability, confidentiality and authentication. These concerns together with the challenges discussed in the previous section, are used as the basis for establishing the requirements for an integrated mobile agent security framework. This paper proposed the following eight requirements for an integrated mobile agent security framework:
The framework must provide different levels of security, depending on the type of implementation environment in which the mobile agent would be deployed.
The framework must incorporate different levels of security depending on the type of application and agent objectives.
The framework must maintain and not hamper the autonomy and mobility factor of the agent.
Additional security implementations on the remote hosts (and the system as a whole) must be kept to the minimum, to reduce cost and time. This includes both additional hardware and software requirements.
The number of communication sessions between the remote hosts (and between remote hosts and other entities) must be minimized. There also needs to be no permanent connection between the agent and the local host.
Computational cost of implementing countermeasures and maintenance thereof must be as low as possible.
The cost of implementation should be affordable or at least minimized. The financial costs of implementing countermeasures need to be in direct relation with the degree of security required.
The host must possess intrinsic mechanisms to support the security requirements of the agent. This implies the provision and integration of additional security functions and services, according to the needs of the application and hence the agent.

3. SECURITY MECHANISMS

When designing mechanisms for the security framework's properties of adaptability, survivability and believability, one question considered was how mobile agents themselves could be used for security. After all, there are several existing examples of mobile code and agents aiding security. E.g., Data Verification One feature of the Semper electronic commerce framework is its conflict mediation functions that are used during transactions. The role of these functions is to keep each party informed of his obligations at each stage of a transaction and to propose corrective actions in the event of a protocol going awry. The functions could run (as an agent) at the host of each transaction party instead of at a third party host.
This approach has the advantage of allowing security checks to happen "off-line". Another verification approach for programs down-loaded to hosts is proof carrying code. This is a proof of program correctness that is evaluated as the program executes. This is close to the agent model in that the security program is dynamically distributed to hosts. Login Applets A simple login applet reads a name, password and perhaps other in- formation and then brings this data back to the server. Such applets are quite common on the Internet. The reason for using an agent is that the server does not need to block threads for the client while he is typing the password. The server can process the request fully and completely when the applet returns. Also, the login procedure for the user need not be known to him in advance. Active Networking Active networking, where programs can be distributed to network nodes to intelligently process application packets, is a natural target for security processing such as key management. The approach has already been used for mobile fire-walls and intrusion detection. There are two ways in which agents are being used in these examples. First, to move security processing nearer to the user or server; second, to dynamically distribute security programs.
This can be further exploited for survivability and believability. For survivability, the platform an help increase the chances of an agent's survival by providing replication mechanisms so that a computation can be split and sent on different routes in

order to tolerate attacks on individual agents from different hosts. For believability, active credentials can be executed to verify properties of agents. Agents acting as credentials have the advantage that they add behavior and flexibility to the framework. The case of verifying that an agent has not received a kill signal for instance is more conveniently done using agents since the meaning of validity is application-specific, and so requires code specific to the application. Active credentials are thus similar in spirit to proof-carrying code. Our goal is not to argue that agents are a panacea for security, but that their use can sometimes make sense. In practice, security policy modeling in the Internet context must include risk analysis. For instance, the commonly cited airline reservation scenario where an agent visits the server of competing airlines to learn the cheapest fare is becoming the reference example for the mobile host problem. The security risk is that a malicious server can alter the quote of a competing company. The scenario is too risky; a less risk prone use of agents in the airline example is for intelligent or batch bookings. For instance, an agent is sent to a server with a request "I want 2 seats on a flight to Paris with an overnight stopover in Bonn. If this costs me more than 100EUR, then reserve a direct flight". In this case, the request is being shipped to the server side for execution; the agent approach is useful because the client-server interaction is happening on the same machine, unaffected by slow network connections, and the user can be off-line. From the client's security viewpoint, the cost of an attack on his agent is the same as the cost of an attack on the messages that he would exchange with the airline server in the client-server approach.

Considering risk analysis in the HyperNews example, the key k used to decrypt the article on the client host is destroyed after the decryption to reduce the risk of the key being illegally copied. The browser used to view the articles is tailored, and does not possess printing or file saving capabilities. Of course, if the HyperNews platform has been tampered with, then the security is broken since access to the article key can be got and distributed, thus avoiding the need for payment. However, the security policy for HyperNews was designed as part of a business plan and contained an in-depth risk assessment. It was felt that the effort needed by an ordinary user to subvert his Java platform exceeds the gain - free access to a few articles which only cost a few centimes anyway.

4. EVOLUTIONS OF ANALYSIS

A number of mobile agent systems that can be available to use as a basis for the generation of mobile agent applications. A large number of these systems are the result of research projects initiated by academic and research institutions. As the acceptance of mobile agent systems is reliant on their ability to provide protection for the mobile agent, it is essential to evaluate the described mobile agent systems against the requirements for a security framework, in order to aid in the process of defining such a framework. In the proposed framework an evaluation is carried out with few analysis as discussed in a sequential manner.

4.1 Implementation environment

The analysis of the mobile agent systems and tools as described in the previous chapter, displayed that none of the systems provide for different levels of security depending on the type of implementation environment. A large number of these systems are built on the security designs of the underlying operating system, language or virtual machine and only make use of encryption and digital signature algorithms for providing security to the agent.

4.2 Autonomy and mobility

A large number of systems do not inhibit the autonomy and mobility of the agent. Systems that do however place a restriction on the autonomy or mobility of the mobile agent are for example Agent TCL [5], which requires the agent to register at the remote host before migration, aZIMAs [6] that makes use of a trusted set of hosts and Jumping Beans (Jumping Beans) that entails the agent being transferred to a trusted central host between migrations.

4.3 Additional requirements for implementation

Mobile agent systems such as (ADK (ADK), Aglets [7], Ajanta [8], are built on the Java platform, which require the installation of the Java virtual machine before the implementation of the agent systems. Agent TCL [5] and TACOMA [9] are built on the TCL scripting language systems such as ADK (ADK) and AMETAS [ ] incorporate digital signing of parts of the agent, which will require a certification authority for the provision of private/public key pairs. It is also possible that the certification authority can form part of the functions of the current host. Agent TCL [5] requires an additional server within a domain for registration and key management purposes of the mobile agent.

4.4 Number of communication sessions

Additional communication sessions for the distribution of keys will depend on the location (or use) of a certification authority. For example ADK (ADK) and AMETAS [10], make use of digital signing and will require the generation of public/private key pairs either by the host (no additional communication sessions) or a certification authority (additional communication sessions). Agent TCL [5] requires the agent to first register at a server for encryption and signing purposes, before being sent to the first remote host. This implies additional communication sessions.

4.5 Computational costs

Additional costs in terms of computations are considered in cases where the mobile agent system makes use of cryptography techniques for encryption and signing purposes. Examples of mobile agent systems that incorporate digital signing and certificates are ADK (ADK), Agent TCL [5] and AMETAS [10]. Ajanta [8] also incorporates the use of logs for detection purposes that have added computational costs.

4.6 Financial implications

A number of systems are being developed as research projects at various institutions, of which some progressed to become commercial systems. A mobile agent system that can be used for research purposes (but needs to be paid for if used commercially) is ADK (ADK). Examples of systems that are available for deploying mobile agent applications free of charge are Aglets [7] and Agent TCL [5] of which the latter also requires an additional server (such as a certification authority) for registering and signing the agent.

4.7 Choices of countermeasures

A large number of systems don't provide the owner or developer of the mobile agent with a choice of possible countermeasures. Systems such as ADK (ADK), only provide for the digital signing of parts (or whole) of the agent, while systems such as Agent TCL [5] also incorporates encryption techniques. It is however possible to incorporate possible additional countermeasures based on the system used for development and deployment of the mobile agent system. For example Java provides a number of possibilities such as encryption as well as different encryption algorithms and programs. Ajanta [9] provides three layers of protection, namely read-only containers, append-only logs and only accessible to certain hosts.

5. CONCLUSION

Through this paper able to identify the most salient characteristics in available security frameworks and mobile agent systems, but also isolate the drawbacks, which up to this point, still leaves a mobile agent vulnerable for malicious hosts attacks. It accumulated background knowledge and arguments were used to describe a dynamic mobile agent security framework that is based on the definition of multiple security levels, depending on the type of deployment environment as well as type of application.

6. REFERENCES

[1] Aerts, A.T.M., Szirbik, N.B. & Goossenaerts, J.B.M. 2002. A flexible, agent- based ICT architecture for virtual enterprises. Computers in Industry, 49(3): 311-328. AGENT DEVELOPMENT KIT (ADK). [Online]. Available at: <http://www.tryllian.com/development/DOCS2_1/ devguide /ch12.html>. Accessed: 08/01/2004.

[2] Albayrak, S. & Wieczorek, D. 1999. JIAC - A Toolkit for Telecommunication Applications. In: Proceedings of the Third International Workshop on Intelligent Agents for Telecommunication Applications. Sweden:1-18.

[3] Algesheimer, J., Cachin, C., Camenisch, J. & Karjoth, G. 2001. Cryptographic Security for Mobile Code. In: The 2001 IEEE Symposium on Security and Privacy, Oakland, California.

[4] An, L., Jiang, Q., Luo, X. & Ren, Z. 2002. Protecting Mobile Agents against Malicious Hosts. Term Paper.

[5] Robert S. Gray. "Agent Tcl: A transportable agent system", Proceedings of the CIKM Workshop on Intelligent Information Agents, Fourth International Conference on Information and Knowledge Management (CIKM 95), Baltimore, Maryland, December 1995.

[6] Nalla, A., Helal, A. & Renganarayanan, V. 2002. aZIMAs - almost Zero Infrastructure Mobile Agent System. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, Florida.

[7] Vigna, G., Cassell, B. & Fayram, D. 2002. An Intrusion Detection System for Aglets. In: Suri, N (ed.). Proceedings of the International Conference on Mobile Agents. Barcelona, Spain,

[8] Tripathi, A.R., Karnik, N.M., Vora, M.K, Ahmed, T. & Singh, R.D. 1999. Mobile Agent Programming in Ajanta. In: Proceedings of the 19th International Conference on Distributed Computing Systems (ICDCS '99). Austin, Texas: 190-197.

[9] "TACOMA- Fundamental Abstractions supporting Agent Computing In A Distributed Environment" Nils Peter Sudmann, Department of Computer Science, University of Tromsø, Norway, November 1996.

[10] Zapf, M., Müller, H. & Geihs, K. 1998. Security requirements for Mobile Agents in Electronic Markets. In: Working Conference on Trends in Distributed Systems for Electronic Commerce (TrEC'98), LNCS, Hamburg, Germany, Springer

[11] Asaka, M., Okazawa, S., Taguchi, A. & Goto, S. 1999. A Method of Tracing Intruders by Use of Mobile Agent. In: Proceedings of the 9th Annual Internetworking Conference (INET99). San Jose, California.

[12] Aslam, J., Cremonini, M., Kotz, D., & Rus, D. 2001. Using Mobile Agents for Analyzing Intrusion in Computer Networks. In: Proceedings of the Workshop on Mobile Object Systems. ECOOP 2001.

[13] BAEK, J. 1999. A Design of a Protocol for Detecting a Mobile Agent Clone and its correctness proof using Coloured Petri Nets. In: Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing. Atlanta, Georgia.

[14] Baumann, J., Hohl, F., Rothermel, K. & Straber, M. 1998. Mole - Concepts of a Mobile Agent System. World Wide Web Journal, 1(3):12-137.

[15] Robert S. Gray. "Agent Tcl: A transportable agent system", Proceedings of the CIKM Workshop on Intelligent Information Agents, Fourth International Conference on Information and Knowledge Management (CIKM 95), Baltimore, Maryland, December 1995.