Implementation and Performance Evaluation of a Protocol for Detecting Suspicious Hosts

Oscar Esparza, Miguel Soriano, Jose L. Muñoz, and J. Forné


{oascar.esparza, soriano, jose.munoz, jforne}@entel.upc.es

Abstract. Mobile agents are software entities that consist of code, data and state, and that can migrate autonomously from host to host performing some actions on behalf of a user. Security issues restrict the use of code mobility despite its benefits. The protection of mobile agents from the attacks of malicious hosts is considered by far the most difficult security problem to solve in mobile agent systems. In this paper the authors present the implementation and the performance evaluation of a new protocol for detecting suspicious hosts in the mobile agent environment. The protocol is based on limiting the execution time of mobile agents in the executing hosts and was previously presented in [6]. The authors also present how this protocol can improve the usability of the cryptographic traces approach [12].

1 Introduction

Mobile agents are software entities that consist of code, data and state, and that can migrate from host to host performing actions autonomously on behalf of a user. Therefore, mobile agents are especially useful to perform functions automatically in almost all electronic services. Unfortunately, massive use of mobile agents is restricted by security issues despite their benefits.

There are two prior entities involved in this scenario: the mobile agent and the host. Protection is necessary when trustworthy relationships between entities cannot be assured. Host protection from malicious agent attacks can be achieved by limiting the execution environment with sand-boxing techniques and a proper access control. Communication security is based on well-known cryptographic protocols. Nevertheless, there is no proper solution to protect mobile agents completely against the attacks of malicious hosts.

In [6] the authors presented a protocol that detects hosts that are suspicious of malicious behavior. The protocol measures the agent’s execution times in the hosts, and these times are the evidence of the good or the bad behavior of the executing hosts. In this paper we present the implementation and some performance evaluation results of this protocol. Additionally, we present how the protocol can improve the usability of the cryptographic traces approach [12].

The rest of the paper is organized as follows: Section 2 presents the state-of-the-art related solutions; Section 3 presents how to use the protocol with the
cryptographic traces; Section 4 presents some implementation details and some performance evaluation; finally, some conclusions can be found in Section 5.

2 Malicious Hosts

The attacks performed to a mobile agent by an executing host are considered the most difficult problem regarding mobile agent security. Assuring the integrity and authentication of code, data or results that come from other hosts is possible by using digital signature or encryption techniques. However, it is difficult to detect or prevent the attacks performed by a malicious host during the agent’s execution. Malicious hosts could try to get some profit of the agent reading or modifying the code, the data, the communications or even the results due to their complete control on the execution.

There are some published approaches that deal with the problem of malicious hosts, but none of them solves it completely. Yee introduces the idea of a "Sanctuary" [13] as a closed tamper-proof hardware subsystem where agents can be executed in a secure way, but this forces each host to buy a hardware equipment and to consider the hardware provider as trusted. The environmental key generation [9] makes the agent's code impossible to decrypt until the proper conditions happen on the environment, so previous analysis from hosts is avoided. However, there is no protection once the code has been decrypted. Hohl presents obfuscation [8] as the mechanism to assure the execution integrity during a period of time, but this time depends on the capacity of the malicious host to analyze the obfuscated code. The use of encrypted programs [11] is proposed as the only way to give privacy and integrity to mobile code. Further improvements in this sense were addressed in [5,4]. The difficulty here is to find functions that can be executed in an encrypted way. Vigna [12] introduced the idea of cryptographic traces that the agent takes during execution. The agent sender asks for the traces if it wants to verify execution, but verification is only performed in case of suspicion. Roth [10] presents the idea of cooperative agents that share secrets and decisions and have a disjunct itinerary. This fact makes collusion attacks difficult, but not impossible.

3 Usability of the Protocol

In [6] the authors presented a protocol that detects hosts that are suspicious of malicious behavior. As malicious hosts need time to analyze and modify a mobile agent in order to take some profit, we can detect suspicious behaviors by controlling the time spent executing the agent. Each executing host counts the running time with regard to a time reference. When the agent reaches the origin host, a set of time checks evaluates if the hosts spent more time than expected executing the agent. A host is suspicious if it does not pass any of the time checks. The protocol also protects the mobile agent while it is migrating from host to host by using cryptographic techniques.