Journal of Computer Science 7 (4): 488-492, 2011
ISSN 1549-3636

A Secure Mobile Agent System against Tailgating Attacks

[1]P. Marikkannu, [1]Adri Jovin and [2]Purusothaman
[1]Department of Information Technology,
Anna University of Technology, Coimbatore, India
[2]Department of Computer Science and Engineering and Information Technology,
Government College of Technology, Coimbatore, India

Abstract: Problem statement: A lot of real time applications have extended their hands towards Mobile Agents to accomplish various tasks, due to its flexibility in its functionalities. Since Mobile Agent Systems are used in a distributed environment, it is obvious that they may be vulnerable to various security threats. Most of the security threats faced by the mobile agent systems are overcome by existing security algorithms and architectures. One of the major threats which are not of much focus in a mobile agent system is the Tailgating attack. **Approach:** In this study, mobile agent system architecture has been proposed to overcome Tailgating. The proposed architecture uses the mechanism of Dual Check-point to preclude Tailgating attacks. Also, to support the mechanism, we use fragmentation and defragmentation techniques. This assures a free flow of data within the system. **Results:** The results obtained prove that the system is much efficient in its operations as well is immune to Tailgating attacks. **Conclusion:** The security of the system is improved with the implementation of the Dual Check-point method.

Key words: Mobile agent system, tailgating attacks, dual check-point, malicious mobile agents, authentication table, authentication check, agent generator, fragmentation module, traditional system

INTRODUCTION

The concept of mobile agents has revolutionized the world of computing by introducing mobility to the code and by reducing the traffic in the network. Because of this, the usage of mobile agents has been introduced in a number of real time applications. With the growth in the technology, the growth of threats against the technology has also been found to be increasing. As a result of this, the trust over the technology may get affected. However, with the growing threats, security mechanisms to overcome the threats have also been proposed by various experts related to the domain. Some of the major threats have been identified and solutions are proposed to overcome those. Some may not be identified easily, but could be identified only through keen observation. On such a context, by our keen observation over mobile agent system failure it was identified that an attack which had been used in the physical world has been introduced in the world of mobile agents. The attack that had been used was the Tailgating attack which has less significance in the domain of Information Security, since it is not electronically used widely. But now, it has been identified that it poses a major threat towards the security of mobile agents.

In this study, a technique named Dual Check-point has been used to combat with the Tailgating attacks. Also, this technique incorporates the fragmentation and defragmentation of data in order to provide complete support to the Dual Check-point security technique. The main advantage of this technique is that the system, as a whole, could be controlled by the administrator. Also, this would ensure a secure mobile agent environment which is immune to Tailgating attacks.

Background: Mobile Agents are piece of codes or modules which are capable of moving from one location to another in order to accomplish a task. Mobile agents departed from Mobile Host, migrate and search to get the information appropriate to user's requirements (Punithavathi and Duraiswamy, 2010). The location may be a platform in some other host or a platform in the same host. The mobility of the code is considered to advantageous which would help in the reduction of load in the network. A mobile agent is capable of collecting data from various sources. These collected data could be transferred to some other system

Corresponding Author: P. Marikkannu, Department of Information Technology, Anna University of Technology, Coimbatore, India

which integrates and processes the data. The mobile agent itself is capable of processing the data it holds. The mobile agents are used in various applications related to a distributed system where transfer of information from one part of the system to another is a mandatory one. A mobile agent supports asynchronous and autonomous operation, allows dynamic and flexible adaptation to a changing environment, allows reduction of communication cost and allows encapsulation of protocols (Baumann, 2000). Also the mobile agent exhibits the features such as autonomy, social behavior, reactivity and proactivity which overcome the drawbacks of the client-server architecture (Braun and Rossak, 2005).

Tailgating is a security threat which is often encountered in the physical world. It involves a process in which a person, whether an employee or not, passes through a secure door without the knowledge of the person who has legitimate access to the particular secure door (Hayden, 2010; Tipton and Krause, 2004; 2008). The same case is applicable for mobile agents also. In the mobile agent system, a mobile agent may possess the access to a secure system. Some malicious mobile agents which keep track of the legitimate mobile agent enters the system when the legitimate mobile agent enters the same system.

Scenario: While considering the tailgating attack, not much literature is available of the mode of attack or the nature of attack with respect to a mobile agent system. By our keen observation, we had found that the malicious mobile agent appends with the legitimate mobile agent either by attacking the mobile agent or by injecting the malicious code within the mobile agent. This may make it possible for the mobile agent system to allow the malicious agent within itself thereby allowing the entire system to be affected.

Consider a secure bank system. A legitimate mobile agent, which works on behalf of a user carries the account number and the Personal Index Number within it. The attacker desires to access the account details or transfer the credits from the user's account to his account. So the attacker designs a mobile agent which would append with the legitimate mobile agent without disturbing the internal functionalities or the data within the agent.

When the legitimate agent, which carries the malicious mobile agent reaches the gateway of the secure bank system gets access by using the account number and the Personal Index Number. The malicious mobile agent, as soon as it enters the secure bank system detaches from the legitimate mobile agent and starts functioning autonomously and gains access over the system. Further, it starts performing its malicious functionalities which may affect the entire system.

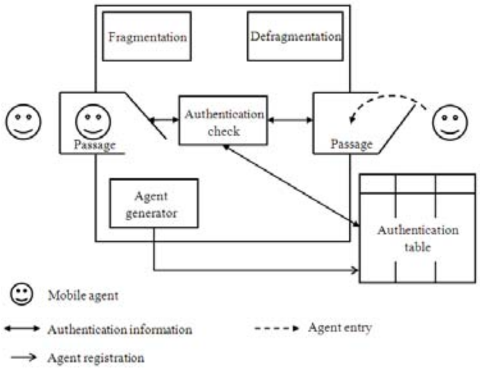


Fig. 1: Secure mobile agent system architecture

Table 1: Authentication table

Agent ID	Host ID	Size (in kb)
001	001	28
028	003	29
023	002	32
027	001	31

In this study, we concentrate on a mobile agent system architecture which would protect the system from Tailgating attacks and maintain the stability of the system. Also we assure the Quality of Protection by modeling the system using the proposed architecture.

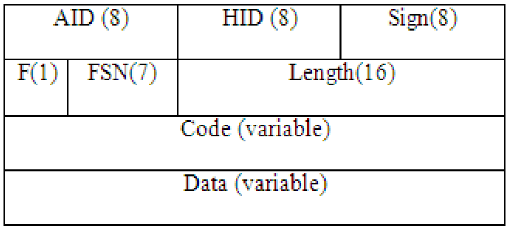
Proposed solution: The proposed architecture to overcome the tailgating attack is shown in Fig. 1.

The Secure Mobile Agent system proposed in this study contains the following components of which most of the components are integrated within the Agent Platform:

- Authentication table
- Passage
- Authentication check
- Agent generator
- Fragmentation
- Defragmentation

Authentication table: The authentication table is a centralized database which contains details of the mobile agents generated in the system. The mobile agent system registers all the mobile agents created by it to the authentication table. Only those mobile agents which are registered with this table are allowed to access the gate. The authentication table is kept in a secure environment. The structure of the Authentication table is shown in Table 1.

The Agent ID corresponds to the individual identity of a mobile agent. The Host ID is the identity



AID (8)		HID (8)	Sign(8)
F(1)	FSN(7)	Length(16)	
Code (variable)			
Data (variable)			

Fig. 2: Mobile agent packet format

of the Host from where the Mobile Agent was generated. The size corresponds to the total size of the Mobile Agent along with the data.

Passage: The passage plays a very important role in the mobile agent system. The passage uses the Dual Check-point entry scheme, which is very helpful in controlling Tailgating attacks. The passage has two gates namely the External gate and the internal gate. The passage contains a buffer of definite size which could be determined by the administrator. The mobile agents carrying the data must be of the size of this buffer. The process is discussed in the later part of this study. The passage is connected to the Authentication Check which checks whether a mobile agent is legitimate or not. The passage is just a locker which holds the mobile agent inside itself at the time of verification. When a mobile agent enters the passage, it is forced to be in a zero-execution state in order to prevent damages, incase if a malicious mobile agent tries to breach it.

Authentication check: The Authentication Check is one of the most important components of the Mobile Agent System. It obtains authentication details from the central authentication table. It performs authentication checking at the external as well as internal gates of the passage. This also could instruct the passage to kill a mobile agent if it is found to be malicious.

Agent generator: The agent generator is responsible for the creation of new mobile agents which would transmit data from one platform to another.

Fragmentation: The fragmentation module is one which makes a main contribution to the sizing of the mobile agents. When the size mobile agent in combination with the data is larger than the buffer size of the passage, the fragmentation module fragments the data and helps to maintain the stability of the system. The excess data is carried by new mobile agents generated by the Agent Generator. Those mobile agents do not perform any special task other than carrying the data. They get destroyed once they reach the destination and offer the data to the destination platform. The Fragmentation module sets a fragment bit and a fragmentation sequence number for the data that is being carried by the mobile agents. The format of the mobile agent with the data will be like that shown in Fig. 2.

In the Fig. 2, the various fields in a mobile agent data packet are shown. The notations or numbers given within the brackets indicate the maximum allowable size of the field in bits.

AID refers to the Agent Identifier which uniquely distinguishes the mobile agent in the system. AID is an 8 bit field. HID refers to the Host Identifier which refers to the identity by which a host is distinguished in the entire system. It is also an 8 bit field.

Sign refers to the digital signature which is a proof of authentication. F refers to the Fragmentation bit which is a one bit field. If it is 0, then no fragmentation of data has occurred. If it is set to 1, fragmentation of data has occurred. FSN refers to the Fragment Sequence Number. This field will hold some number if the Fragment Bit is set to 1. Length refers to the length of data that is being carried by the mobile agent. The code corresponds to the Mobile Agent code which may be of some variable length. The data is the data that is being carried by the mobile agent.

Defragmentation: The Defragmentation module performs the reverse process of fragmentation. It checks the fragmentation bit. If the fragmentation bit is set to 0, it performs no action. If the bit is set to 1, it checks the Fragmentation Sequence Number, collects the data, sequence the data and integrate it for the system to process the data.

Working mechanism: Consider the Secure Banking System which follows the proposed Secure Mobile Agent System Architecture. Now, a client wishes to access some details from the bank. The client therefore generates a mobile agent and the data it needs to carry is provided by the user. The size of the mobile agent should be pre-determined by the administrator before determining the buffer space size in the passage. The size of the mobile agent along with the data should be to the maximum equal to the size of the buffer in the passage. If the size exceeds, the data is fragmented in order to make it attain the required size and the data are carried by dummy agents whose purpose is just carriage of data.

Now the client registers the Agent ID, Host ID and the size of the Mobile Agent along with the data in the authentication Table. The Authentication Table acts as a lookup record which could be referred by the Authentication Check in any platform which has registered to the Authentication Table.

The data is now being transmitted to the secure bank system. Now the mobile agent reaches the external gate of the passage which performs verification of authentication using the digital signature it possesses. If the mobile agent does not pass the authentication check, it is not allowed inside the passage. If the mobile agent passes the check, it is allowed into the passage buffer where the mobile agent is subjected to another authentication check. The external gate is closed as soon as the mobile agent enters the passage buffer. No other mobile agent is allowed to enter the passage buffer unless the verification inside the passage is completed and the mobile agent is let inside the platform. If the mobile agent along with the data is larger than the passage buffer, it is expelled out of the system. The mobile agent is kept in 'zero execution' state while this check is performed. The passage uses the Authentication Check to perform this. In case, the mobile agent fails this test, it is killed immediately inside the passage itself.

As soon as the mobile agent enters the platform the Defragmentation module checks whether the fragment bit is set or not. If the fragment bit is found to be set to 0, then the mobile agent is allowed to access the resource it needs. In case if the fragment bit is set to 1, then the mobile agent is kept in hold unless all the data in the sequence are received. Once all the data had arrived, the defragmentation module integrates the data and appends it to the actual mobile agent. It keeps all the dummy agents unless the mobile agent completes all the process in the platform. Once the task is accomplished and the result obtained, the mobile agent with the data is subjected to size check by the fragmentation module. If fragmentation is needed, it performs fragmentation and appends the data to the dummy agents which were kept waiting. In case, if there is no need for fragmentation the dummy agents will be destroyed. If the number of dummy agents is less than the number of fragment of data, new dummy mobile agents will be created by the secure mobile agent system's Agent Generator. The resultant data is send to the client which requested the service.

The above mechanism, though pose some restriction over the size of the data, is found to be much efficient to combat with the Tailgating attacks. Since the mobile agent along with the data is of a consistent size, it is not possible for the malicious agent to travel behind or append with the legitimate mobile agent. Since, the dual check point system performs various types of checks; it is not so easy to by-pass or breach the security. Moreover, the zero execution environments help to secure the system from being collapsed by the malicious agent at the time of check.

MATERIALS AND METHODS

An experimentation to compare the ordinary authentication verification based on digital signature and our proposed architecture was done. The banking system was taken for this task. A mobile agent system was designed using the IBM Aglets 2.0.1. As described in the scenario, the client uses the Account number, Personal Index Number and a digital signature to authenticate itself with the Banking System. In the secure system, the passage buffer was set to 16KB. Therefore the fragmentation module fragments the data and mobile agent to 16KB combined. The experimentation was done parallel in 2 rows, with 12 PCs with Intel Dual Core Processors in each row using Windows XP Operating System. Totally 24 systems were used. 11 systems acts as clients and one system contains the central banking system in each row. One row contains the banking system with our proposed architecture and the other with the traditional digital signature based authentication system.

RESULTS AND DISCUSSION

The experimentation was performed for a period of nearly 6 h. Malicious mobile agents which have the capability to inject or append code to other mobile agents were introduced inside the network from an external source. Various results were obtained and performance comparisons based on various parameters were done.

The first comparison was based on the number of malicious agents introduced against the number of Malicious Agents detected by the system. Figure 3 clearly shows the distinction in detection between the proposed system and the traditional system.

The graph is based on the following experimentation. Malicious Agents were introduced to move around the system in order of increasing numbers. At first, one malicious agent was introduced in the system and was verified whether any change is observed. It was observed that the system with the proposed architecture detected the malicious agent and killed it.

The same malicious agent when introduced in the traditional system could not identify the malicious agent and had fallen prey to the malicious agent. Now both the systems were reset to their initial conditions. Two malicious mobile agents were introduced. At this moment also, the traditional system was found to be non-functional with respect to detection.

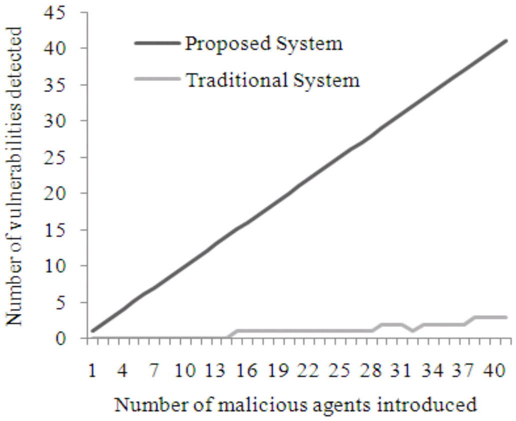


Fig. 3: Detection of malicious agent

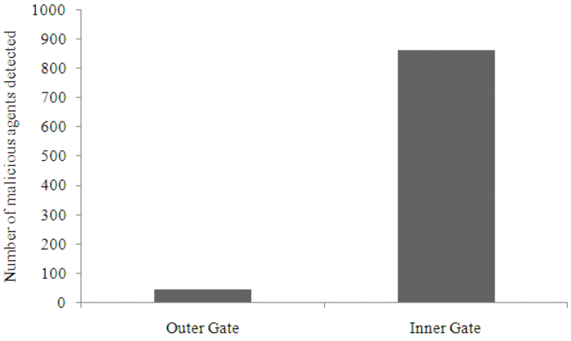


Fig. 4: Malicious agents detection

This experiment was done again and again with increasing count of malicious mobile agents up to 41. It was observed that the performance of the proposed system was consistent in detection when compared to the traditional system. It could be seen that the traditional system could not detect any malicious mobile agents up to 14. It was because those malicious mobile agents were specially designed to perform tailgating. Whereas after 14, it could be found that the traditional system could detect those malicious mobile agents which possessed a false digital signature or those which provided irrelevant Account Number or Personal Index Number.

In such a view, it was found that the proposed system was much advantageous in detecting and offending tailgating attacks.

The next analysis is based on the number of malicious agents that were detected by the outer gate which is based on authentication check by Digital Signature and those detected by the inner gate which is based on the Digital Signature as well the size of the mobile agent that has entered the passage. Those mobile agents which were expelled due to non-accommodation space within the passage buffer due to excess size are also taken into consideration as detection by the inner gate. The observed result is displayed in Fig. 4.

From the Fig. 4, it could be observed that the vulnerabilities detected by the Outer gate are less when compared to that of the Inner Gate. Thus, a conclusion could be made that the Dual Check-point system is much efficient in finding the malicious agents which are responsible for tailgating attacks.

CONCLUSION

In this study, an efficient architecture to overcome the tailgating attack has been designed for a mobile agent system. The introduction of the Dual Check-point system and the application of constant size for mobile agents prove to be effective in encountering the tailgating attacks. Also, the fragmentation and defragmentation process included in this system supports the architecture to the maximum extent and thereby help the system to identify the tailgating attacks. Another main objective of this study, which is to bring awareness about tailgating attacks in a mobile agent environment, has also been accomplished by description of the nature of attack in the mobile agent system and an efficient method to overcome this attack has also been achieved in this study.

REFERENCES

Baumann, J., 2000. Mobile Agents: Control Algorithms. 1st Edn., Springer, USA., ISBN-10: 3540411925, pp: 180.

Braun, P. and W. Rossak, 2005. Mobile Agents-Basic Concepts, Mobility models and the Tracy Toolkit. 1st Edn., Morgan Kaufmann, USA., ISBN-10: 9781558608177, pp: 464.

Hayden, L., 2010. IT Security Metrics: A practical framework for Measuring Security and Protecting Data. 1st Edn., Tata Mc-Graw Hill Education, New York, ISBN: 0071070907, pp: 408.

Punithavathi, R. and K. Duraiswamy, 2010. A fault tolerant mobile agent information retrieval system. J. Comput. Sci., 6: 553-556. DOI: 10.3844/jcssp.2010.553.556

Tipton, H.F. and M. Krause, 2004. Information Security Management Handbook. 5th Edn., CRC Press, USA., ISBN-10: 0849319978, pp: 2036.

Tipton, H.F. and M. Krause, 2008. Information Security Management Handbook. 6th Edn., Auerbach Publications, Japan, ISBN-10:1420067087, pp:456.