Robust Algorithm for Securing an Agent Hosting Platform

Aarti Singh[1], Parul Ahuja[2]
[1]Associate Prof., MMICT & BM, MMU, Haryana, India
[2]Scholar, JMIT, Radaur, Haryana, India

Corresponding Author Email: singh2208@gmail.com

Abstract

Although agent based frameworks are finding applications in critical and sensitive applications; however with these applications are originating concerns for security & privacy of such intelligent systems. Mobile agent frameworks must be able to counter new threats as agents may opt to masquerade, generate Denial-of –service and may try to access unauthorized resources. This paper focuses on security issues of mobile agents, particularly narrowing towards agent-to-platform issues of security. This work proposes a robust two layer algorithm for ensuring complete security of the host platform by overcoming the above mentioned threats, the proposed algorithm is robust since even on failure of one layer it continues to provide security to the platform. In the proposed framework agent is required to earn a reputation score by behaving well on a host platform and thus provides genuine means to judge it.

Keywords: *Software Agents, Mobile Agent, Agent Platform, Security Issues, Reputation score.*

1. Introduction

An agent system generally [7, 13] comprises of two components , the agent and its execution platform. The agent platform provides the computation environment in which the agent operates. The platform from which, the mobile agent is originated is called home platform. The *home platform* is normally considered as the most trusted and safe environment for execution of a mobile agent. Figure 1 given below provides general agent system model.

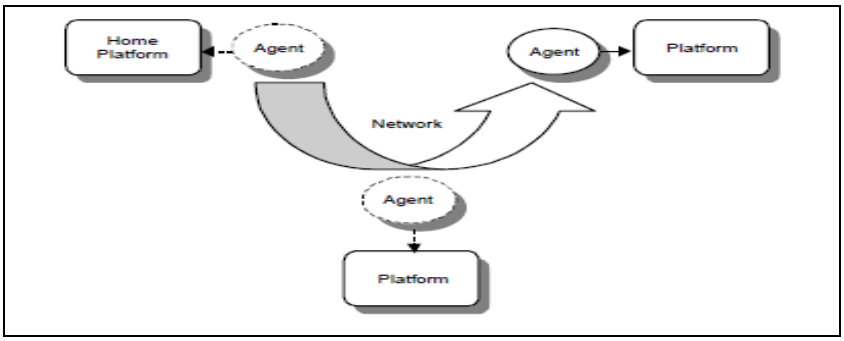

Figure1: Agent System Model [7]

An agent's execution unit contains its code, data, and execution information needed to carry out some computation. In order to fulfill the delegated task, the agent migrates autonomously

along a sequence of agent platforms. While migrating a mobile agent may encounter four types of threat: agent-to-platform, platform-to-agent, agent-to-agent, and others-to-agent platform. Focus of this research is on *agent-to-platform* security issues [6, 7, 13] representing the set of threats in which the visiting mobile agent may breach the security policy of the platform and launch attacks [9] against an agent platform. This set of threats includes masquerading, denial of service, and unauthorized access [11].

Masquerading refers to concealing the truth. In this attack, an agent may pretend as a legitimate agent in order to gain access to valuable resources of platform and services to which it is not entitled. While in *Denial of Service a* visiting mobile agent consumes excessive amount of computational resources available on the platform rendering it unable to serve legitimate requests. Additionally a mobile agent may try to access the resources of the platform without having adequate permission, or can access residual data stored in a cache or other temporary storage.

From the above discussion, it is clear that mobile agent hosting platforms must have strong security mechanisms to prevent themselves from attacks by incoming agents. Most of the researches done in this field, have been concerned about ensuring the security of the mobile agents [8, 10, 12, 18]; but very few have focused on securing host agent platform. Security breaches on the platform can result in significant losses. Thus, this research work aims to propose a robust platform security framework, which can prevent all threats from malicious agents.

The paper is structured as follows. Section 2 explores the work and views of eminent researchers in proposing countermeasures for dealing with mobile agent security issues. Section 3 provides details of the proposed framework. Section 4 concludes the paper.

2. Related Work

This section explores the available literature and highlights the areas, which need to be paid attention.

Researchers in [1,2,3], have proposed probable models, theories, architectures and implementations of mobile agents in various agent systems, but most of them have failed to incorporate trust and security of communicating parties as well as of the data. However, work by [4,5,7,11,13] have made an effort to include various security threats, security requirements that need to be met in order to alleviate those threats, but are missing on the practical implementation of the techniques and are still at the theoretical level. Though some of them have made an effort to propose the mechanisms and some countermeasures for the security of mobile agent and agent platform, but still none of the method has proved to be firm enough. Various authors [8,9,10,12] have provided strengthening techniques to include trust and security of messages exchanged but have remained silent and paid less attention towards platform security.

Many authors have proposed researches in this area revealing similar or sometimes distinct views. Knoll et. al in [16], have presented path-based security for mobile agents that extends the security of the NOMADS mobile agent system in a multi-hop scenario, and suggested extending the system to support a finer granularity of trust levels. While, Karnouskos in [17] tries to combine domains of active network & that of agent technology, and put forth the need to reinvent the wheel of code mobility and security every time in every new approach adopted. Borselius in [6], highlighted that trusted hardware is required to address security problems of

mobile agents which may be too expensive for most applications. For mobile agents to be more widely adopted, security issues must be paid more attention. Some authors [11] have also stated that a combination of various techniques may yield powerful solution. Having explored the literature it was discovered that a need for a complete security solution especially securing the host platform was felt therefore an effort to provide such a solution has been made in the upcoming section by proposing a unique security layer at the host platform level.

3. Proposed Work

The proposed framework identifies the un-trusted mobile agents requesting platform access and keeps an eye on activities of hosted agents to prevent masquerading, denial of service and unauthorized access from them. To establish initial trust level and prove its authenticity, every mobile agent is assumed to get registered with Central Certificate Authority (CCA). As is evident from [12] that a digitally signed trust certificate is issued to an agent at the time of registration. It is only after registration that a mobile agent is able to access and provide services. This work proposes a Platform Security Framework (PSF) which is activated whenever a mobile agent request for a service. PSF comprises of two layers namely *Authentication & Authorization Layer* (AAL), and *Supervision & Filtration Layer* (SFL) respectively. Figure 2 given below provides the High-level view of the proposed PSF.
Every incoming request for platform access first goes to AAL for authentication and privilege authorization. Once AAL approves the agent, it enters in Supervision & Filtration Layer (SFL). The composition and working of these layers is as follows:

- ***Authentication & Authorization Layer*** **(*AAL*):** This layer is responsible for entertaining every incoming platform access request. It comprises of an *Interface Agent* (IA), supported by a *List_of_Trusted_Entities* (LTE), and a *Reputation_Buffer* containing two slots namely, *Own_Platform_History_Reputation buffer* (OPHRB) & *Path_History_Reputation buffer* (PHRB). An agent requesting for platform access sends its DSTC along with reputation certificates received from all earlier visited platform. IA checks DSTC against the LTE. If it finds the name of certificate issuing authority in this list, then the agent is authenticated otherwise not. After approving DSTC, IA sends an agent authentication request to the mentioned home-platform of that agent and waits for its response. If the requested platform confirms the identity of that agent, only then IA proceeds for privilege assignment otherwise agent is refused access.

This technique ensures prevention of masquerading attack. Once the agent gets authenticated, IA reviews its reputation both at its own platform and at other visited platforms, in order to decide the privileges to be assigned. For this IA explores the OPHRB, to check if agent has visited the platform earlier, if yes, what was its reputation. OPHRB maintains reputation table illustrated as Table 1.

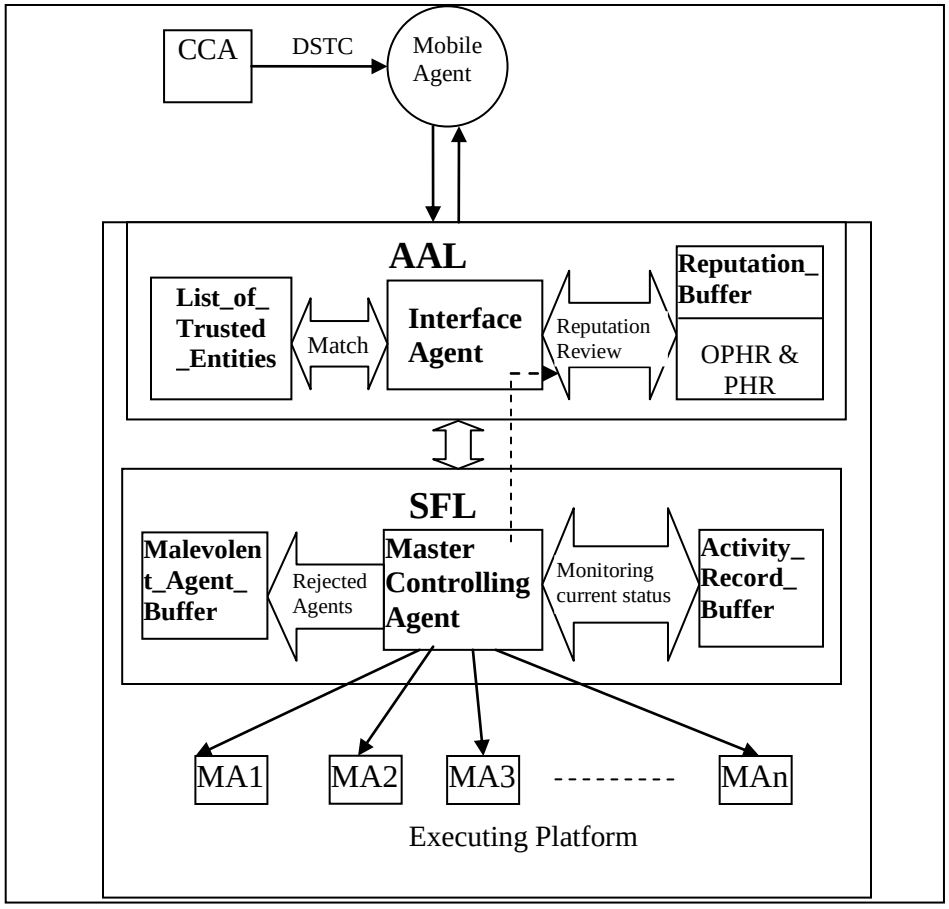

Figure 2: High-level View of the Proposed Platform Security Framework (PSF)

In case, if an agent arrives for the first time, its reputation is reviewed based on certificates obtained from earlier visited platforms and limited resource access is provided. *Reputation_History* parameter (in table 1) contains <No. of visit(s), score earned for the visit>. This coloum can record history for previous five or more visits. In this work reputation history is maintained for recent five visits however this number may increase depending on availability of memory on hosting platform. If an agent earns positive score in some of its visits, it does not mean that it will always behave well, since agents are by nature self motivated and their intention keep on changing. Thus, if an earlier good agent starts scoring -1 and continues to do so till its score reaches zero, leading to a poor reputation on the platform, its future access on the platform is banned .

Table 1: Structure of Own_Platform_History_Reputation_Buffer

Agent_Id	DSTC_From	No._of_Times_ Platform_Visited	Reputation_ History	Total_Reputation_ Score	Remarks
A101	IBM	2	(i) (1,1) (ii) (2,-1)	0	No Access
A202	DELL	0	Null	0	Not Applicable
A204	MSoft	3	(i) (1,1) (ii) (2,1) (iii) (3,1)	3	Good

After reputation evaluation IA assigns the privileges as per table 2 given below.

Table 2: Reputation based Privilege Assignment

Mobile Agent Type	Reputation Score Range(s)	Reputation_Review	Privileges_Assigned
New Arrival	0	Not Applicable	Restricted Access
Previous Interaction	(i) 1 to 5	Good	Read, Write
	(ii) 6 to 9	V. Good	Read, Write, Execute
	(iii) 10	Excellent	Own platform checking exempted
	(iv) <=0(less than or equal to 0)	Poor	No Access

Algorithm for the AAL is given in Figure 3 below.

```
AAL()
{ on input active IA()
  { 1 receive input certificates from requesting agent;
    2 match DSTC with LTE;
    3   if (found)
         { Send authentication request to Home platform;
            if (authenticated by home platform)
              Send Handshaking;
           else
             { send refusal;
              return;}
          }
        else
         { send refusal;
            return;}
     4   check OPHRB and PHRB and assign privileges to guest
        agent;
     5  return();
     }
  }
```

Figure 3: Algorithm for AAL

• ***Supervision & Filtration Layer (SFL):*** This layer contains *Master_Controlling_Agent* (MCA) supported with *Activity_Record_Buffer* (ARB) and *Malevolent_Agent_Buffer* (MAB). When an agent enters SFL, MCA assigns it an execution area along with initial resources based on its privileges. MCA keeps an eye over every request made by guest agents by putting it in ARB. Whenever an agent attempts to go beyond its assigned privileges, MCA halts execution of that agent immediately and creates an entry in MAB. MCA also observes the no. of times same request had been generated by the guest agent. If the number exceeds threshold value, MCA considers this an attempt for Denial of Service attack, and thus terminates the execution of agent, and finally creates an entry of this misconduct in MAB for future reference. Every misconduct by a guest agent leads to a -1 (minus 1) in its reputation score. MCA reports the agent behavior to IA at the end of its session, for reflecting it in OPHRB. On successful completion of session by an agent, its reputation score is incremented by 1 and, -1 otherwise. Algorithm for SFL is given in Figure 4 below.

```
SFL( )
 { on input activate MCA( );
     { 1 create entry for the incoming agent in the ARB and assign
         it an    execution area;
       2 make entry for requested resources in ARB;
       3 if((nextrequest==lastrequest)&&(requestnumber>threshold))
            { terminate session;
              create entry in MAB;
              send reputation score in OPHR;
            }
            else
            { record request in ARB;
              grant request;
            }
         4 if (request==session terminate)
           { close session;
             Send reputation score in OPHR;
             assign reputation certificate to guest agent;
           }
         5 return();
       }
  }
```

Figure 4: Algorithm for SFL

3.1 Flow Diagram

The flow diagram given in figure 5(a) illustrates the detailed working of the proposed framework. The mobile agent residing at the *Home Platform* wants to make transition to the *Foreign Platform* during its itinerary session. The flow of control on PSF is as follows:

1 Guest Agent requests for permission along with its certificates.
2 IA matches agent's identity against the trusted entities list LTE.
3(a) Concludes trusted agent based on the matched result of step 2.
3(b) Concludes un-trusted agent based on the unmatched result of step 2.
4 IA sends validation check message to home platform of guest agent.
5(a) Response (valid mobile agent) from home platform.
5(b) Response (invalid mobile agent) from home platform.
6(a) IA sends permission granted (handshaking) message to guest mobile agent based on the response in 5(a).
6(b) IA sends Access denied (unauthentic code) message to guest mobile agent based on the response in 5(b).
7 IA reviews the Reputation_Buffer and after exploring both parts, decides privileges.
8 AAL forwards the confirmed, trusted and authorized guest agents down to MCA in SFL for assignment of an execution space.
9 MCA assigns a separate execution area and initial resources to all guest agents based on the privileges assigned to them.
10 MCA monitors the current activity status of each executing guest agent by observing all the requests made by guest agents and records them in ARB.
11 MCA terminates session of an agent that tries to access any resource more than the threshold limit and creates an entry for the same in MAB.
12 on termination of session for an agent MCA sends its reputation score for the session to IA for maintaining in OPHRB.

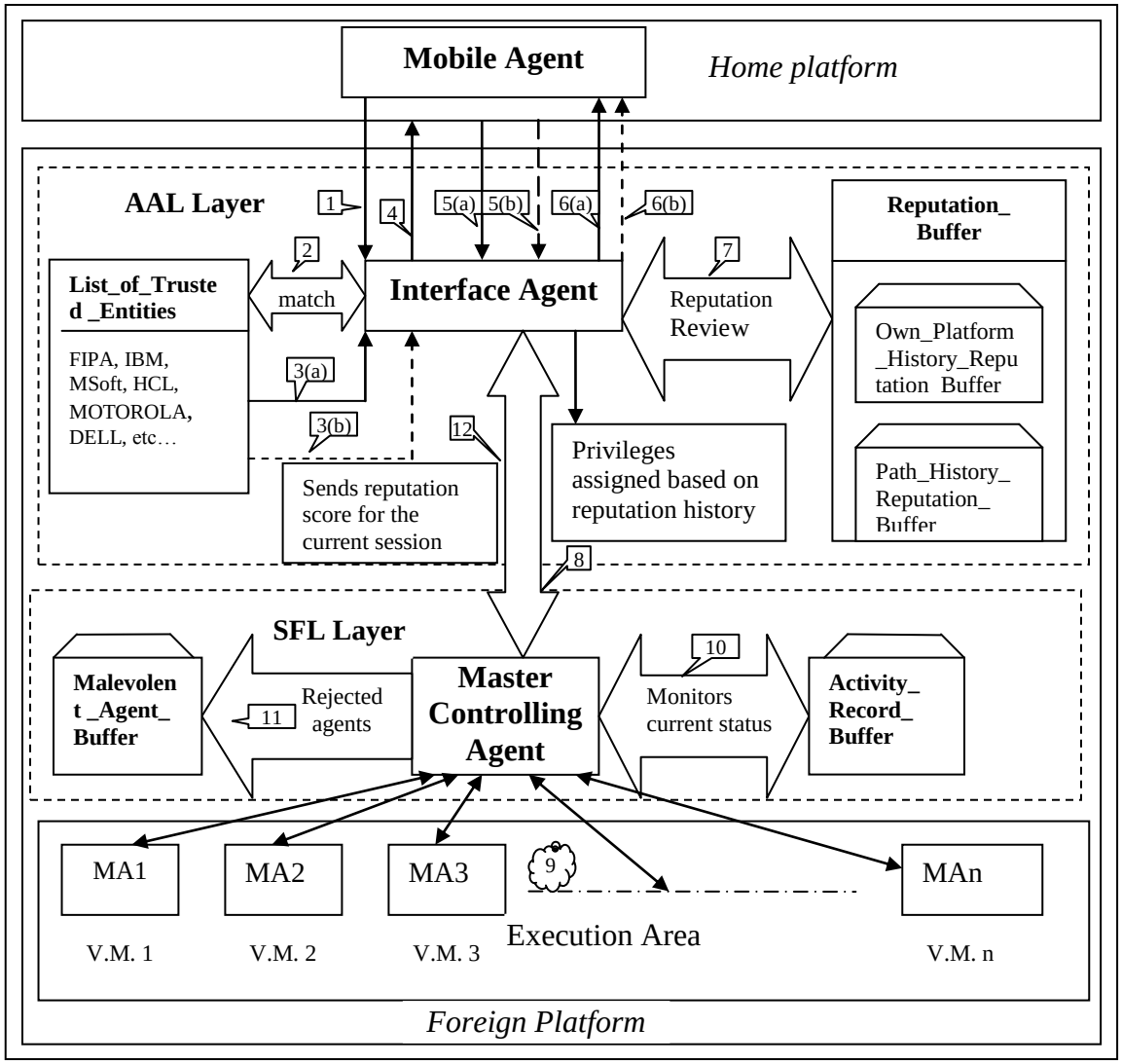

Figure 5(a): Flow Diagram of the Proposed Framework

Figure 5(b) given below provides algorithm for the PSF.

```
Algorithm: PSF( )
{1  Guest  Agent  requests  for  permission  along  with  its
    certificates;
  2  activate AAL();
  3  activate SFL();
}
```

Figure 5(b): Algorithm for PSF

4. Conclusions

Mobile agent paradigm relies heavily on security of both the agent as well as its host platform. This work has presented a robust platform security algorithm, which eliminates all of the security issues related to the host platform. This framework makes use of reputation score, which may be earned by an agent by appropriate behavior, and no agent is considered trustworthy forever. All the hosted agents are kept under the vigilance of SFL to look for any

abnormal or malicious behavior. Being designed in two layer, makes the framework robust since even if one layer fails the other layer continues to work and provides some security to host platform. Thus this framework may result in increased security of the host platforms. However, practical implementation of the proposed framework is still under progress.

References

[1] Picco, G.P.,:Mobile agents:An Introduction. In Microprocessors and Microsystems, Vol. 25, pp. 65-74, 2001, *Milan, Italy.*

[2] Jennings, N.R., Wooldridge, M.: Software Agents. IEE Review, January 1996 pp. 17–20 .

[3] Griss, M.L.: Software Agents as Next Generation Software Components. In *Component-Based Software Engineering: Putting the Pieces Together*,Addison Wiseley Publications, May 2001.

[4] Ojesanmi, O.A.:Security Issues in Mobile Agents. In International Journal of Agent Technologies and Systems, Vol.2, issue 4, pp. 39-55, 2010.

[5] Farmer,W.M., Guttman,J.D. and Swarup,V.: Security for Mobile Agents: Issues and Requirements. In Proceedings of the 19th National Information Systems Security Conference,Vol. 2, pp. 591-597. National Institute of Standards and Technology,Baltimore, Maryland, October 1996.

[6] Borselius,N.:Mobile Agent Security. In IEEE Journal of Electronics & Communication Engineering , Vol. 14, issue 5,pp. 211-218, October 2002.

[7] Jansen,W. and Karygiannis,T.: Mobile Agent Security. In NIST Special Publication, Vol. 800, issue-19, pp. 39, 1999.

[8] An,L.,Jiang,Q., Luo,X. and Ren,Z.: Protecting Mobile Agents Against Malicious Hosts. In CS685-002 Term Paper, Spring 2002.

[9] Dadhich, P., Dutta,K., and Govil, M.C. ,: Security Issues in Mobile Agents. In International Journal of Computer Applications,Vol. 11,issue 4, December 2010.

[10] Rizvi, S.M.S.I. , Sultana, Z., Sun, B. and Islam, Md. W.: Security of Mobile Agent in Ad hoc Network using Threshold Cryptography. In World Academy of Science, Engineering and Technology, 70- 2010.

[11] Alfalayleh,M. and Brankovic,L.: An Overview of Security Issues and Techniques in Mobile Agents. In International Federation for Information Processing (IFIP), Vol. 175, pp. 59-78, October 2005.

[12] Singh,A., Juneja,D., and Sharma,A.K.: Elliptical Curve Cryptography Based Security Engine for Multiagent Systems Operating in Semantic Cyberspace. In International Journal of Research and Review in Computer Science (IJRRCS), Vol. 2, No. 2, April 2011.

[13] Jansen,W.A.:Countermeasures for Mobile Agent Security.In Computer Communications, Vol. 23, Issue 17,pp. 1667-1676,November 2000.

[14] Bellifemine, F., Rimassa, G., Poggi, A., JADE - A FIPA-compliant Agent Framework. In Proceedings of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, London, 1999.

[15] Singh,A., Juneja,D. and Sharma,A.K.: Agent Development Toolkits. In International Journal of Advancements in Technology (IJoAT), Vol. 2, No. 1,January 2011.

[16] Knoll, G., Suri, N., & Bradshaw, J. M. (2001). Path based security for mobile agents. In Proceedings of the First International Workshop on the Security of Mobile Multi-Agent Systems (SEMAS-2001) at the Fifth International Conference on Autonomous Agents (Agents2001), (pp. 54-60). Montreal, CA, New York: ACM Press,

[17] Karnouskos, S.:Security Implications of Implementing Active Network Infrastructures using Agent Technology", Special Issue on Active Networks and Services, In Computer Networks Journal, Elsevier, Vol. 36, Issue 1, pp. 87-100, June 2001.

[18] J. T. Moore. Mobile Code Security Techniques. Technical Report MS-CIS-98-28, 1998. citeseer.ist.psu.edu/moore98mobile.html

[19] Pfleeger, C. P. and Pfleeger, S. L., "Security in Computing", 3rd edition,Prentice Hall 2003. ISBN: 0-13-035548-8.