JCOP: A Security Framework for JADE Intra Platform Mobility

Salman Qabaja
Department of Electrical and Computer Engineering
Palestine Polytechnic University
Palestine
sqabaja@student.ppu.edu

Radwan Tahboub
Department of Electrical and Computer Engineering
Palestine Polytechnic University
Palestine
radwant@ppu.edu

Abstract– Mobile agent is a new distributed computing paradigm where agent autonomously as a process migrates from host to anther carrying its data code and state to execute certain task on behalf of user (human, program). Hence Security is very fundamental in this paper we propose a security framework for JADE (Java Agent Development Environment) intra platform mobility. JADE is environment for developing multi-agent system that is fully implemented in Java. Our framework is based on enforcing security by applying authentication and access control which agent is defined it by itself in a user transparent method.

I. Introduction

Client server is one of the dominant paradigms that distributed applications relied on. Communication is one of the most important issues in distributed computing, in client server this part have passed throw different levels of evolution. Firstly message passing where communication accomplished throw message exchange between the communicated processes or objects [1],then remote procedure call(RPC) which allow computer program to execute in another address space[2], then remote evaluation(REV) which give a process the ability to evaluate a program expression at remote computer[3], the REV model had opened new prospects in distributed systems communication, the idea of making mobile objects that encapsulate both data and set of object operations that do the work on behalf of the user(human or program) . That was the born of mobile agent paradigm. So, agent is program that act on behalf of a user. And by that mobile agent is a program that represents a user in a computer network and can migrate autonomously from node to node, to perform some computation on behalf of the user [4]. Mobile agent had not faithfully accepted in the IT world because of the lack of security which is one of the most fundamental issues in digital communication. A lot of security frameworks where proposed and some of them where implemented [6, 7, 8]. Many mobile agent systems have been implemented (JADE, Aglets, Agent Tcl, Ajanta, Concordia and others) some of these platforms partially address the security problems, other ignore this issue [4]. In this paper we propose a security framework that for JADE (Java Agent Development Environment). JADE is fully implemented in Java, and is FIPA standard compliant for intelligent software agent. JADE provide agents called containers that acts as platforms that provides execution environment for agents. Multiple agents can run at the same container, and agents can migrate between them, also containers can be distributed over a network of machines. A container called Main Container acts like sever for Agent Management system and Directory Facilitator agents that provides the white page and yellow page services [5]. To be active and host agents container must join agent platform (registered with AMS). Containers within a platform can be either intra connected, where containers can only exchange data with other ones that registered with its home platform figure 1.1. Or inter connected, where containers connected with different AMS's can exchange data with each other.

At this paper we focus on intra connected platforms figure 1.2.

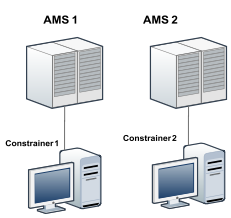

Figure-2 Inter Connected Platform

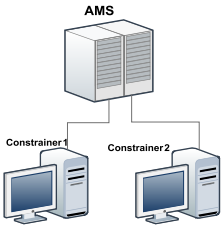

Figure-1 Intra Connected Platform

At begging we will address the threats that mobile agent suffer. We refer to platform that owns (created in) the agent as home platform. (I) Firstly, agent to platform threat, where malicious agent can gain unauthorized access to non home platform resources. (II)Secondly, platform to agent threat where malicious platforms can alternate agent code, data, and stat maliciously. (III)Finally, agent to agent threat where malicious agent can illegally affect other agents. From the seen one can conclude that the main problem is unauthentic and unauthorized access to resources whether agents or computer resources. To protect the mobile agent we focus on two security milestones authentication and access control. Here agents are authenticated using digital signature; every agent is signed by its owner (home platform, user). When agent migrate from host to another (from container to another in JADE perspective) it must be authenticated. Each platform has security manager to enforce and control security flow in the system. When programmer writes the agent code he/she must describe the agent job in simplified English language. When this agent is initiated the security manager explores its goal and determined the agent access policy. In this paper access control is simplified so agent has only two level of access control, full access or no access.

II. Methods

In designing security structures there are two Methodologies, the execution tracing methodology (reactive), or the security enforcement methodology (proactive). In this paper we discuss the reactive methodology.

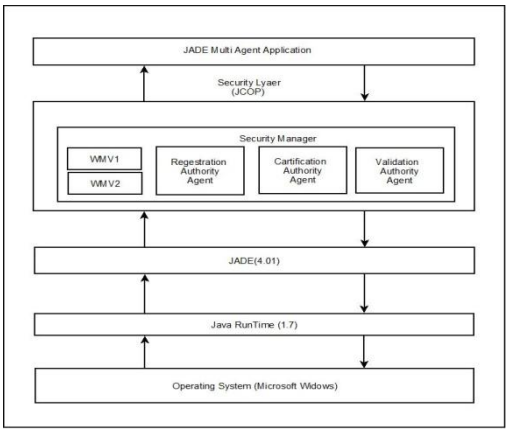

Figure-3 System block diagram

As appears in the above figure JCOP works as separate layer of the system. JCOP framework includes the following components:

A. Security Manager (SM)

SM is the main structure of the framework. It consists of a group of agents that cooperate to gather to provide the security service to the system. The security manager agents are:

1. Registration Authority (RA): when new agent created it must registered with the RA so it can live in the protected framework. Simply the new agent sends its goal (access policy) to the RA. If the goal accepted the agent will be validated to live in the platform, if not the agent will be killed.
2. Certification Authority (CA): the CA will receive certification request from the RA when new agent is validated. In response CA generate certificate signed by SM public key and store it in the cert store.
3. Validation Authority (VA): when agent travels from container to another the VA authenticate the agent. if the agent is valid then its free to go, if not then agent killed.
4. Watchman V1 (WMV1): this agent works as inspector agent that watches the agent creation event in the platform and inform the RA.
5. Watchman V2 (WMV2): this agent works as inspector agent that watches the agent

movement's event in the platform and inform the VA.

B. Security classes

Is a group of classes and data abstractions that used to activate the framework.

1. Certifier: class that agent uses to get certified.
2. Certificate: is data abstraction that is used to store digital signature.

III. Implementation

We implement the following architecture:

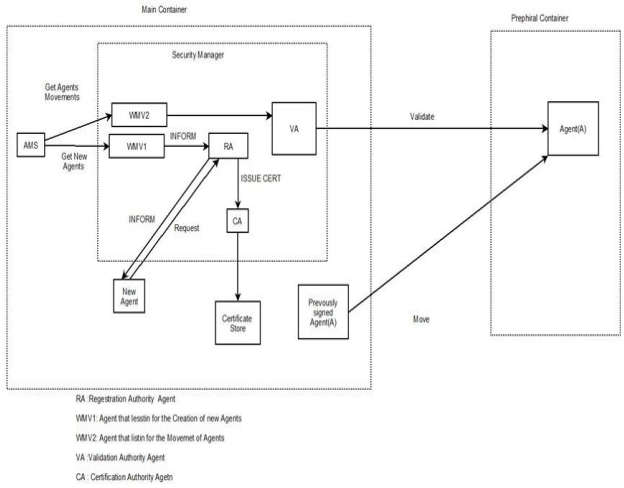

Figure-4

When new agent created in the platform the WMV1 will catch that event and send inform message to RA contains the new agent name. RA sends inform message to the new agent. In response the agent will send a message that contains its goal. The RA will analyze the message with simple text analyzer that will compare the agent goal message with previously defined list of words called black list that contains reserved words (actions) that agents are forbidden to do in the platform this list is defined as the SM access policy. If the agent goal does not match the SM access policy agent will be killed. Else RA will send request message to the CA to certify the newly created. The CA will issue a certificate that contains the agent data and CA digital signature of that data and store it in the CA Certificate store. If agent moves from container to another the WMV2 will catch that movement and inform the VA. The validation authority will then go to certificate store search for the agent certificate validates that certificate. If the agent certificate is not valid the agent will be killed. Else the agent will allowed living in the platform.

IV. Results

Many factors used to test the system, each one of these factors used to measure the efficiency and reliability of this system. In this paper the following factors used to test the system CPU, number of threads per the whole process, time VS number of certified agents all at a time, time VS number of authenticated agents all at time.

- CPU: the percentage of CPU that JCOP adds to the original JADE process CPU average. This means the CPU overhead that the JCOP agents add to the system to accomplish the security work.

Process	JCOP ON/OFF	Average CPU %
JADE	OFF (Trial 1)	0.75
JADE	ON (Trial 1)	0.85
JADE	ON (Trial 2)	0.88
JADE	ON (Trial 2)	0.90

Table-1

- Threads: the number of threads that JCOP adds to the original JADE Process. This shows the number of the JCOP agent and other agent running on the system. Since agent at least represented by one thread.

Process	JCOP ON/OFF	# of threads
JADE	OFF (Trial 1)	31
JADE	ON (Trial 1)	35
JADE	ON (Trial 2)	41
JADE	ON (Trial 2)	38

Table-2

- Time of certification and authentication VS number of Agents: here the time needed for agent to discovered and certified by JCOP is measured, and the reason behind that to discover the amount of time that agent can cause harm to the system before it been

discovered by JCOP, and the time needed to discover agent after traveling from host to host which is equally in importance to the time of certification, that's because it is assimilates the time period that agent can cause harm to system before it is authenticated.

Number of Agents	Certification Time	Authentication Time
1	80	3
2	132	9
4	259	21
8	514	43

Table-3

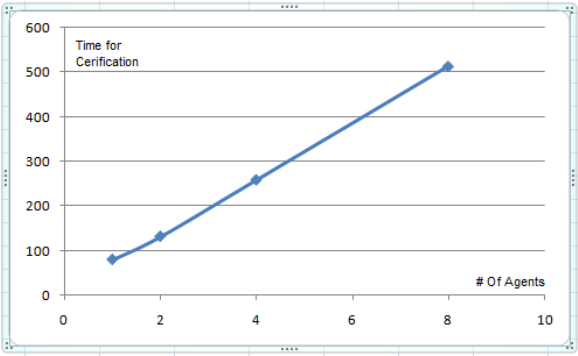

Figure-5 This figure describes the relation between the number of agent created at one time and the time it takes JCOP to certify them all at time.

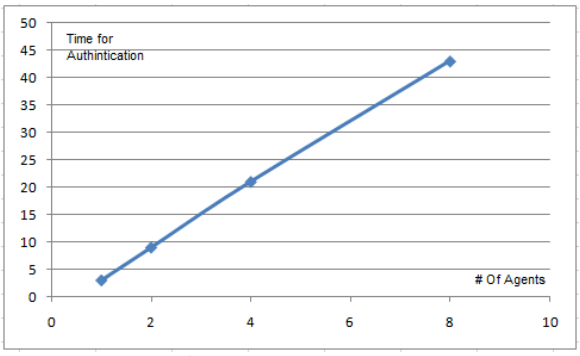

Figure-6 This figure describes the relation between the number of agents traveled from host to another one at a time and the time it takes JCOP to authenticate then all at time.

Conclusion

JCOP is a simple security framework it has no big overhead on the system performance and that clearly appears from the experiments results. The all the factors measure prove this, for CPU no great overhead added to the whole process CPU portion . For threads the number added to the system threads represents the JCOP agents. And clearly the for certification and authentication is small and the relation between both the number of agents newly and number of agents traveled form host to host at one time and time for certification and verification is very close to the linear relation and that proves the framework ability of handling increasing number of agent without being drawn. The maturity of this framework is no complete and still needs some work especially in the part of mobile certificates, since the certificate only stored on a central machine and that somewhat conflicts with mobility nature of the mobile agent.

Acknowledgment

For PHD. Ammal Aldwaik Wazwaz, Eng. khaliel Ikhlaiel and Eng. Saqer Atawneh. Thank you for your help and support.

References

[1] Tanadeau. Message passing -Wikipedia, the free encyclopedia. Wikipedia the free encyclopedia. Last modified at 03:32, 2 May 2012. Accessed at 18:10, 18 May 2012. **http://en.wikipedia.org/wiki/Message_passing**

[2]Tgeairn. Remote procedure call -Wikipedia, the free encyclopedia. Wikipedia the free encyclopedia. Last modified at 06:12, 18 May 2012. Accessed at 18:10, 18 May 2012. **http://en.wikipedia.org/wiki/Remote_procedure_call**

[3] J. W. Stamos and D. K. Gifford "Remote Evaluation" ACM Transactions on Programming Languages and Systems, Vol. 12, No. 4, October 1990.

[4] Karnik, N.M. Tripathi .A.R. "Design issues in mobile agent programming systems" IEEE Concurrency Vol. 6 Issue: 3 on page(s): 52 – 61 Jul-Sep 1998

[5] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa "JADEA White Paper" exp - Vol 3 - n. 3 - September 2003

[6] Victor Tan **"**COST EVALUATION OF A PKI-BASED SECURITY FRAMEWORK FOR MOBILE AGENTS**"** Symposium on Progress in Information & Communication Technology 2009

[7] Leila Ismail "A Secure Mobile Agents Platform" JOURNAL OF COMMUNICATIONS, VOL. 3, NO. 2, APRIL 2008

[8] Vandana Gunupudi Stephen R. Tate "SAgent: A Security Framework for JADE" http://jmvidal.cse.sc.edu/library/AAMAS-06/docs/P2gs388.pdf