Building a Distributed Secure System on Multi-Agent Platform Depending on the Principles of Software Engineering Life Cycle

Dr.Ghossoon M.Waleed Al-Saadoon ghowaleed2004@yahoo.com
College of Administrative Sciences,
Applied Science University
Kingdom of Bahrain ,Manama , Jufair, P.O.Box:5055
Tel : +(973) 17728777- 149, Fax: +(973)17728915

Abstract

Today, applications in mobile multi-agent systems require a high degree of confidence that running code inside the system will not be malicious. Also any malicious agents must be identified and contained. Since the inception of mobile agents, the intruder has been addressed using a multitude of techniques, but many of these implementations have only addressed concerns from the position of either the platform or the agents. Very few approaches have undertaken the problem of mobile agent security from both perspectives simultaneously. Furthermore, no middleware exists to facilitate provisioning of the required security qualities of mobile agent software while extensively focusing on easing the software development burden.The aim is to build a distributed secure system using multi-agents by applying the principles of software engineering. The objectives of this paper is to introduce multi agent systems that enhance security rules through the access right to building a distributed secure system integrating with principles of software engineering system life cycle, as well as satisfy the security access right for both platform and agents to improve the three characteristics of agents adaptively, mobility and flexibility, which is the main problem that depending on the principles of software engineering life cycle. There are 3 characteristics that satisfied using agent; mobility, adaptively and flexibility. Adaptively (which is the capability to respond to other agencies and/or environment to some degree). Mobility (the ability to transport itself from one environment to another) and Flexibility (can be defined to include the following properties; responsive, pro-active and social). This project based on the platform of PHP and MYSQL (Database) which can be presented in a website. The implementation and test are applied in both Linux and Windows platforms, including Linux Red Hat 8, Linux Ubuntu 6.06 LTS and Microsoft Windows XP Professional. Since PHP and MySQL are available in almost all operating systems, the result could be tested the platform as long as PHP and MySQL configuration is available.PHP5 and the MySQL (database) software are used to build a secure website. Multiple techniques of security and authentications have been used by multi-agents system. Secure database is encrypted by using md5. Also satisfy the characteristics for security requirements: confidentiality (protection from disclosure to unauthorized persons), integrity (maintaining data consistency) and authentication (assurance of identity of person or originator of data).

Keywords: Distributed Secure System, Platform, Multi Agent System, Access write, Lunix Read Hat , and DM5.

1. INTRODUCTION

Mobile agent technology offers a new computing paradigm in which a software agent can suspend its execution on a host computer, transfer itself to another agent-enabled host on the network, and resume execution on the new host. The area of mobile agent security is in a state of immaturity [1]. Numerous techniques exist to provide security for mobile agents, there is not at present an overall framework that integrates compatible techniques into an effective security model. The traditional host orientation toward security persists and focuses of protection mechanisms within the mobile agent paradigm remains on protecting the agent platform. However, emphasis is slowly moving toward developing techniques that are oriented toward protecting the agent, a much more difficult problem. Fortunately, there are many applications where conventional and emerging security techniques should prove adequate, if applied judiciously. The software was building the new platform using multi-agent system. The Unified Modeling Language (UML) used to build this prototype Model [2]. To make the software easier and systematic, the software engineer must incorporate a development strategy that encompasses the process, method and tool layers. This strategy is called Software Engineering Paradigm to develop Process Model. This paper reviews a web based system and the Prototype Model algorithm used for the system design [3, 4] as in Figure 1.

- The software designed a set of objectives to the users.[5,6].
- The software determines the requirements, and
- The user can review the existing software anytime.

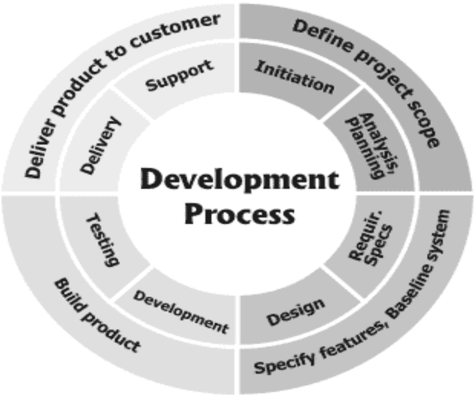

FIGURE1: Software Engineering to Development Process Model

2. LITERATURE REVIEW

There are many literature sources that encourage over this paper. It has been significantly too interpreted about agents and in depth security issues itself that elaborates. These are related sources of the issue in the mobile agents. **Adam Pridgen & Christine Julien ,2006**, they introduce a mobile agent system that enhances security functionality by integrating core software and hardware assurance qualities, as well as addressing security concerns from the perspectives of both the platform and the agent [7**]. Loulou; Mohamed Jmaiel; Ahmed Hadj Kacem and Mohamed Mosbah,2006**, In order to facilitate analysis, design and specification of mobile agent systems, the possible attacks that may occur in a mobile agent system, they associate the specification of the basic concepts that ensuring security such as: agent authenticity, authority access, security policy and its various kind [8]. **Robert S. Gray, George Cybenko, David Kotz, Ronald A. Peterson and Daniela Rus** ,2001, the mobile agent systems involved the relocation of both code and state information. The area of mobile agent security is in a state of immaturity. While numerous techniques exist to provide security for mobile agents, there is not at present an overall framework that integrates compatible techniques into an effective security model [9].

3. METHODOLOGY

System development methodology is a necessary process to develop software .The methodology consists of three main parts to build distributed Secure System using characteristics on multi agent system for this platform and depending on the principles of software engineering life cycle. From UML methodology, the software designer will able to identify the tasks on software developments to present the software architecture and the description of objects and their interactions with one another, as in Figure 2.The platform assigns a newly originated or incoming agent to a requested location or place, where it can compute and interact with other agents. Besides furnishing the engine on which an agent executes its code, typical services offered by an agent platform include the capability for an agent to clone itself, spawn or create new agents, terminate any spawned agents, locate other agents at the platform or a platform elsewhere, send messages to other agents, and relocate itself on another platform, all these process under the security rules and privilege from the management server – web server.

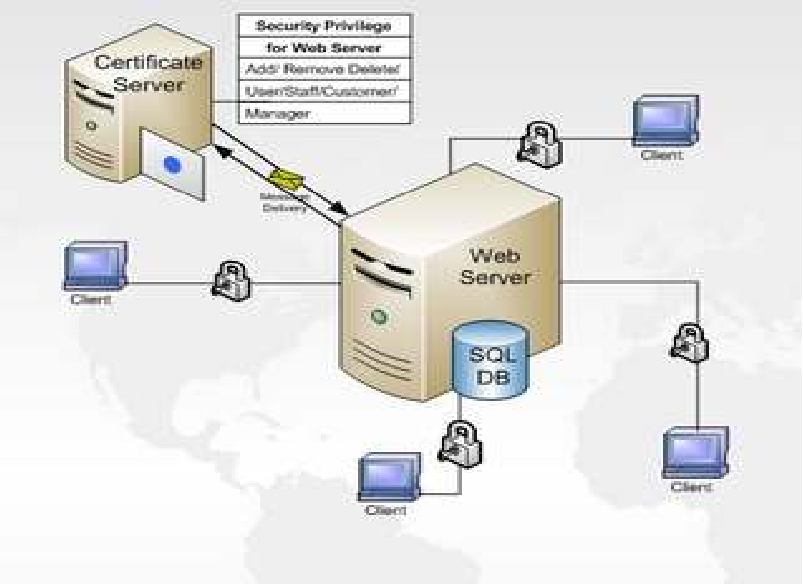

FIGURE 2: Security Platform of Server-Client Architecture

3.1 System Design

In this project has been assumed that client-server websites are used. The security levels have the rules and permission to access each data of clients-server.

The methodology will be divided into four main phases; Preliminary Requirements, Final Requirements, Analysis and finally Design. The 1st stage for the preliminary requirements includes activities which are defining the requirements, validate user requirements, define consensual requirements, establish keywords set and extract limits and constraints.

The 2nd phase in final requirements include some tasks for instance, characterize environment. The process is to determine the entities, define context and characterize environment. The determination is the use cases; it will draw up inventory of the use cases, identify cooperation failures and elaborate sequence diagrams. Follow up then is elaborate User Interface UI prototypes and validate it.

The 3rd phase is the analysis process to identify classes, study interclass relationship and construct the preliminary diagram to verify the global and local levels of mobile agents adequacy. These processes are to know-how study the entities in the domain context and determine agents between entities. All fields of study include the active-passive relationships, active entities relationships and agents relationship.

The final phase is the **design of the architecture and multi-agent mode** that determine packages, classes, design-patterns and elaborate component and class diagrams. Figure 3 shows the principles of software engineering applied in the security management life cycle,

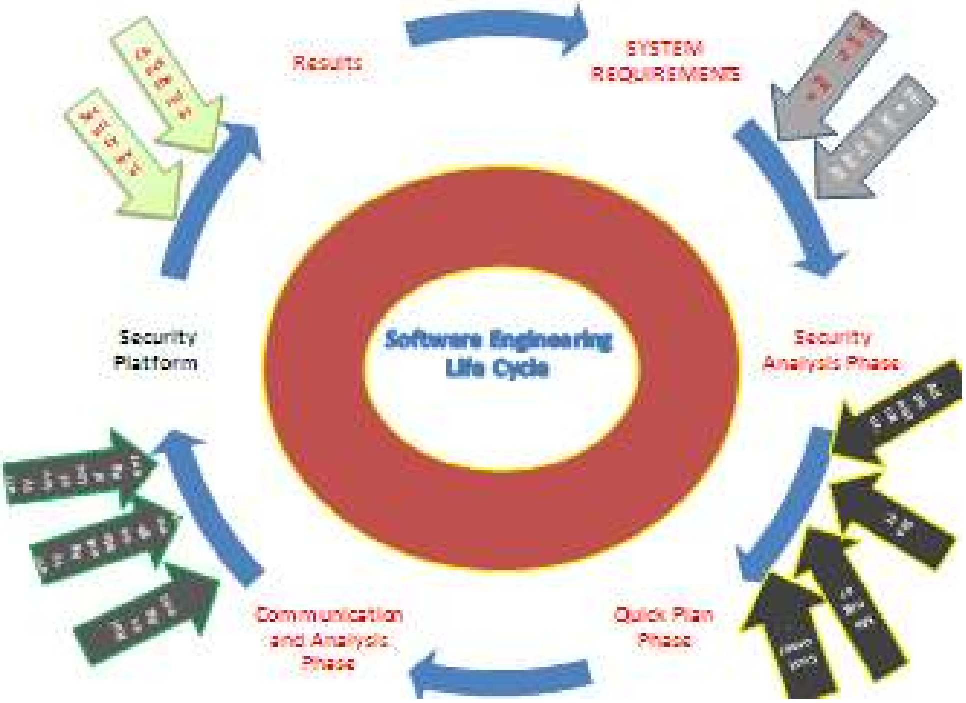

FIGURE 3 :software engineering applied in the security management life cycle

3.2 System Requirements

Web applications run in two locations: the server and the client. This means that both locations need to be developed to provide the best security for the user. The server needs to be developed in such a way that information being stored is not compromised; while the client needs to be developed to present and retrieved only the required information. A client that divulges too much information is not likely to be secure.

The server is where all the application's action is taken place. The PHP operates on a transitive level for the web page between the client and the server. Figure 4 shows the separation of the client and server.

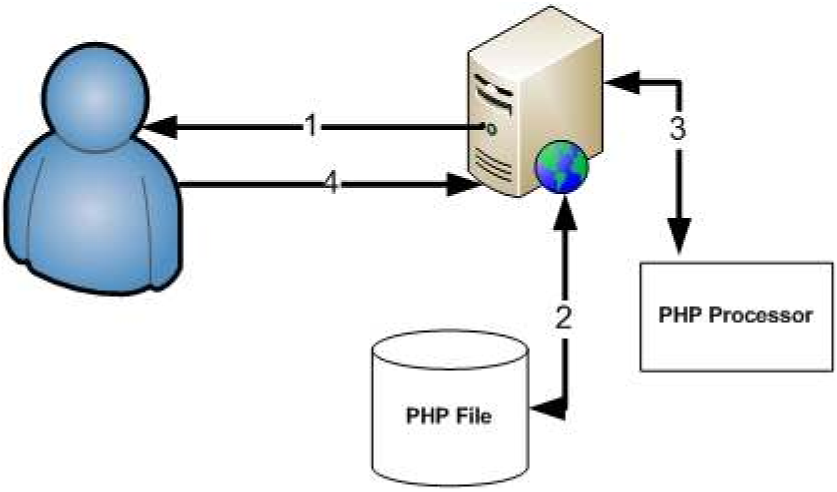

FIGURE 4: PHP Process for Distributed System

3.3 Security Analysis Phase

This phase includes security in PHP and PHP security audit.

I. Security in PHP : it includes these requirements.

1. Security is not an absolute.
2. Can always be more or less secure
3. Security is difficult to measure.
4. Security must be balanced with usability.
5. Security must be balanced with expense.
6. Security must be part of the design.
7. The filter input for the most input is obvious - form data ($_GET and $_POST), cookies ($_COOKIE), RSS feeds, etc. While the output Escaping is the process by which user escape any character that has a special meaning in a remote system. Unless user sending data somewhere unusual, there is probably a function that performs the escaping for. The two most common destinations are the client (use htmlentities ()) and MySQL (use mysql_real_escape_string()). If it must write down, make sure that it is exhaustive - find a reliable and complete list of all special characters.

II. PHP Security Audit

An audit is an examination and it does nothing should be off-limits. A PHP security audit primarily involves an examination of the source code. Other points of interest are software design, PHP configuration, and infrastructure security.

A. Setting the Bar

- How much security is needed?
- Start with a minimum level, and go from there. At the very least, a PHP application should have filter input and escape output.
- If a PHP application can't meet these minimum guidelines, it isn't worth a time.

B. Analyzing the Configuration

- The configuration of PHP is mostly dictated by php.ini.
- However, remember that PHP configuration directives can be modified in other places - httpd.conf, .htaccess, ini_set().
- Things to avoid: register_globals = On, allow_url_fopen = On, display_errors = On, magic_quotes_gpc = On

C. Analyzing the Design

- Have the design explained it first. No one knows an application as well as the developers.
- A poor or unnecessarily complex design is a security risk.
- Is tracking data difficult?
- Is distinguishing between filtered and tainted data difficult?
- Stream-of-consciousness is why so many PHP applications are insecure. This is why so many PHP applications are insecure.

D. Searching the Source: Input

E. Searching the Source: Output

F. Searching the Source: Potential Problems

- Check for dynamic includes that use tainted data: include, require.
- Check for client-side restrictions: maxlength, radio, checkbox, select, Client-side filtering.

G. Searching the Source: Bad Habits

- Error suppression :@
- Misguided trust of HTTP request
- Headers: Referer, Host
- Unescaping: stripslashes(), etc.

The most common mistakes are sending tainted, unescaped data to the client or a database. There are others that most website frequently used: storing the authorization level in a cookie; passing the authorization level in the URL; storing the username and password in a cookie; and storing includes within document root.

3.4 Communication And Analysis Phase

First, all relation information, such as hotel management information, are collected. A discussion is completed with staff members and management and the requirements of system are identified:

- To computerize the current hotel management system.
- To enable the customer to get information about the hotel.
- To enable management to view which staff is members conduct the process with the customers.
- Accessibility to the program must be controlled. The users can't access to the application without authority.
- All levels of users have their own username and password.
- System administrator has the authority to add new user account.
- Staff members have the authority to add new customer records.

3.5 Quick Plan Phase

The system will display the login form when the web page is first accessed. The user needs to enter their username and password. If the username and password is matched with the login information stored in the database, the login is passed. Then the system will check the user level of the login user. There are four user levels in the system:

Administrator, Staff, Manager and Customer.

Each user level has different functionality. Users cannot access additional functionality without authority, as in Table 1.

Administrator	Staff	Manager	Customer
View, add or delete announcement.	View and add announcement.	View and add announcement.	View Result
View all the activities among other users.	Add, delete and update customers.	View customer results.	Change password
Add or remove system user.	View customer results.	Add, delete and update staff.	
View customer result.	Change password.	Change password.	

TABLE 1: privilege for each security level

4. SECURITY FOR SYSTEM DESIGN

First the websites creates specific task with access right for Administration ID to display User Login, Delete, Add new Group, and Add new User. The mobile agents are the interpreted language. The language has to be interpreted, because moving of a running object requires access to the global variables, or better to the current execution pointer and stack. The best way to achieve this is using a virtual machine which executes the interpreter language.

4.1 Security Agent Management

The security agent management architecture is used to design the heterogeneous Database Networks. The main activities of the administrator agent are the following:

- Roles Agent for creating the privileges for security and access control list.
- Creates administrator and local servers.
- Determines the group agents.
- Creating User that can deal in this platform.

In multi agents system based security agent management architecture, two main functionality agents are recognized: Global agents and Local agents.

The management system in this project is districted from public access where the registered users have authority to login the system. There are four user levels in this application: Administrator, Staff, Manager and Customer. Each user has own name and password.

The login process is only performed if the textbox for username and password is filled. Otherwise, a message will appear to ask the user to complete the login form. After the user completes the form, the system will check the login information. If login is successful, the page will redirect to the user home page. If login fails, a message will appear to prompt the user to enter the correct username and password.

Accessibility user level: the Administrator and Staff. If the login is accepted, the page will be redirected to the user home page, where all information posted by Administrator and Staff will be displayed. The user menu will be displayed in the left side of each page after the successful login . The announcement author is the same as the name of the logged in user. The date of the announcement is the current date of the server PC when the announcement is submitted.

Administrator is the only user able to delete announcements. In Figure.6, to delete announcements, simply choose “Edit Announcement” and click the “Delete” link that appear in the bottom of the desired announcement.

Only the Administrator can add a new system user (Staff) to control for security of the system. The user is only added if all textboxes filled and the requested username does not exist in the record. The Administrator will need to create an account for Staff. Then the Staff can add customer information (users) into the database. Password for new user is initially encrypted. The user can change the user password. The Administrator can also delete the user account in case user information is incorrect or user has resigned or write some comments about security encryption . Accessibility user level: System Administrator, Staff and Customer. The user needs to enter their current password and new password twice. The password will be changed if :

4.2 Security Level Local Agents

The security of an administrator constitute a sub-set of hosts in a local network. It is composed of a group of Local Agent (LA)s, which have specific functions. One can distinguish two kinds of LA:

- Intranet LA several Intranet Agents. The intranet agents manage the security of a local network. It controls LA s and analyzes the auditing events reported by these agents.
- Internet LA. In each level, notes agents communicate and exchange their information of heterogeneous DBs and analysis for detecting intrusive activities in a cooperative manner.

4.3 ROLES AGENT

The access key attribute of an agent is that it is able to act autonomously. Agents can then take on a wide range of responsibilities on behalf of users or other system entities including services and entering into agreements. Additionally, an agent will often perform some tasks on behalf of another entity. For example, (a software agent could perform a task on behalf of a person). It could also perform on behalf of another piece of software (another agent), an organization, or a particular role (manager, system administrator).

5. CONCLUSION

The software project was tested in both Linux and Windows platform, including Linux Red Hat 8, Linux Ubuntu 6.06 LTS and Microsoft Windows XP Professional. Since PHP and MySQL are available in almost all operating system,.

- The security platform used to verify and validate multi-agents build the access write. The security platform using mobile agents can satisfied the three characteristics (adaptively, mobility and flexibility).The adaptively can be modified and updated the rules and privileges to the system. While mobility, are the agents itself can be transport from one another to the others platform.
- The configuration of PHP, MySQL and also Apache web server is quite different in different platforms. In Windows XP, the installation of AppServ will install PHP, MySQL, Apache, phpMyAdmin at once. The configuration is done automatically. The PC can run as web server after reboot. Anyway, the firewall of Windows XP need to turn off, otherwise the client PC cannot access the web site hosted by server PC.
- In Linux platform, PHP, MySQL and Apache web server normally is ready and installed. If not, we can manually install them from software package. The directory of the web is located in /vary/www/html. The services of apache and MySQL maybe not start automatically after login to Linux platform depend on the system setting. If the services don't start, we can type a code in terminal window to manually start the services.

This system has the following strengthens, which will increase the commercialization potential:

- It can be run in many platforms such as Microsoft Windows, UNIX and Linux.
- Easy to configure and install in web server.
- The PHP and MySQL are free and open sourcing software's , so very easy to install and used in this project.

6. REFERENCES

1. Dell'Acqua, P., M. Engberg, L.M. Pereira," *An Architecture for a Rational Reactive Agent",* [online] available at:http://centria.di.fct.unl.pt/~lmp/publications/online-papers/epia03-agent.pdf, 2003.

2. Marik, V., O. Stepankova, H. Krautwurmova, M. Luck*.," Multi agent systems and applications II", Springer*-Verlag Berlin Heidelberg, 2002.

3. K. Rabuzin, M. Malekovic, Miroslav Baca (2006), *"A SURVEY OF THE PROPERTIES OF AGENTS"* .pdf, University of Zagreb, Faculty of Organization and Informatics, Varaždin

4. *"An introduction to Agents, Todd Sundsted"* , JavaWorld.com, 06/01/98, [online] available at:http://www.javaworld.com/javaworld/jw-06-1998/jw-06-howto.html

5. Geppert, A., M. Kradolfer, D.Tombros: *"Realization of Cooperative Agents Using an Active Object-Oriented Database Management System*", Proc. 2nd Workshop on Rules in Databases (RIDS), Athens, Greece, 1995.

6. Hayes-Roth, B*." An architecture for adaptive intelligent systems,* Artificial Intelligence Vol. 72, 1995.

7. A. Pridgen and C. Julien*, A Secure Modular Mobile Agent System*, The University of Texas at Austin, TR-UTEDGE-2006-003.

8. M. Loulou, M. Jmaiel,A.Hadj Kacem and M.Mosbah, *"A Conceptual Model for Secure Mobile Agent Systems"*, computational intelligence and security, Proceedings of the 3rd international conference on , Nov 3-6/2006. [online] available at: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4072023.

9. Robert S. Gray, G. Cybenko, D. Kotz, Ronald A. Peterson and Daniela Rus ,*"D'Agents: Applications and Performance of a Mobile-Agent System",* Thayer School of Engineering / Department of Computer Science,Dartmouth College,November 28, 2001.