Al-Aqsa University Journal (Natural Sciences Series),Vol.16, No.2, Pages 1-17 Jan. 2010
ISSN 2070-3155

Mobile-Agent-Based System for Controlling Anti-Virus over LANs

د. توفيق سليمان برهوم*
أ. هبة زهير اللحام *
أ. صفاء السوسي*

الملخص

كلما كان حجم الشبكات المحلية أكبر كلما زادت الهجمات الالكترونية عليها ويصاحب ذلك زيادة في الجهد اللازم لمقاومة هذه الهجمات والتحكم في الشبكة. وبشكل خاص فإن مختبرات الحاسوب في المؤسسات كبيرة الحجم تعاني من مشكلة انتشار الفايروسات وصعوبة متابعة المدير التقني للمختبرات لعدد كبير من الحواسيب بشكل مستمر، والتي تعتبر بدورها مهمة صعبة التنفيذ بشكل متقن. ومع الأخذ بعين الاعتبار استهلاك موارد الحواسيب، فإن الأجهزة تستخدم على الاغلب طوال اليوم دون راحة. بالإضافة إلى أن هذه الحواسيب لا تملك اتصال مباشر بالانترنت مما يؤثر على تحديث قاعدة بيانات فاحص الفايروسات.

تناقش ورقة العمل هذه مدى إمكانية توظيف تقنية الوكيل المتنقل للتحكم في برنامج مضاد الفايروسات المتواجد في حواسيب الشبكة المحلية، وتقدم الورقة آلية لتنفيذ ذلك. يتكون النظام المقترح من أربعة وكلاء متنقلين: الوكيل المدير، الوكيل المتحكم بعملية الفحص، الوكيل المراقب للعمليات والوكيل المراقب للبيئة. يستخدم لتطوير هذا النظام مجموعة من نماذج تطوير البرمجيات: هندسة البرمجيات المعتمدة على تجميع المكونات، ونموذج الإلقاء البعيد.

أحد نتائج ورقة العمل هو إمكانية بناء هذا النظام بناء على ما تم تحديده باستخدام تقنية الوكيل المتنقل.

كلمات مفتاحية: الوكيل المتنقل، مضاد الفايروسات، التحكم في أمان الشبكات المحلية.

* كلية تكنولوجيا المعلومات – الجامعة الإسلامية– غزة - فلسطين.

ABSTRACT

Abstract: The larger the LAN size, the more attacks arise and the more effort is needed to defend and control. Computer laboratories at large enterprises are suffering from the problem of the spread of viruses and the difficulty for laboratory technical administrator to permanently control such a large of number of computers within a LAN. With respect to the computer resources consumption, the LAN computers are extensively used during the day. These computers have no direct connection to the internet which may affect virus scanner's signature update.

This paper discusses the ability to introduce a mechanism to use mobile agent technology to control antivirus software that is available at LAN computers. The proposed system consists of 4 agents: *Coordinator Agent, Scan Control Agent, Process Monitor Agent and Environment Monitor Agent*. A combination of software development models are used to develop such a system, CBSE and Throw-Away Prototyping.

One of the findings of this paper is the applicability to build a system using Mobile Agent Technology.

Keywords: Mobile Agent, Anti-Virus, LAN Security Control

Introduction

In recent years, the damage caused by viruses and worms has rapidly increased and has proved to be a serious problem, especially in large LANs such as those available at universities and large enterprises. As a result, it has become urgent to protect all connected LAN-nodes from malwares. For example, Miwa Company [Yuki] has tried to accomplish this goal by performing the security inspection before the node's connection to a network, but doesn't guarantee nodes security after connection.

On the other hand, there has been a strong tendency to utilize Mobile Agent Technology features as performance and reduce network traffic while controlling applications over networks. Mobile Agents' mobility saves time and effort while installing and managing applications. Therefore, this project is intended to study the ability of building a mobile-agent-based system for controlling Anti-virus software over LANs, and introduce a mechanism for doing so.

It has been found that many Anti-virus software networks exist, but most of them need a permanent connection to remote nodes during scanning

and supporting processes such as EMCO Network Malware Cleaner [URL1]. However, this software causes heavy network traffic. ClamWin [URL2], a leading free open source Anti-virus which ranks the 38th place among Anti-viruses [URL3], supports on-demand protection based on an open source (GPL) anti-virus engine toolkit for UNIX called ClamAV [URL4]. It has a multi-choice command line scanner. Its virus definition is updated daily and automatically. In addition, it is compatible with most of other anti-viruses.

In recent years, Mobile Agents (MA) have been the focus of intensive search in the field of network security or network management in general

For example, Sparta [Krugel] team proposed an architecture that aims to detect security violations in a heterogeneous, networked environment. Other systems have been explained in section 2. Unfortunately, there have been no real attempts to use mobile agents in network nodes protection against viruses, except for two approaches. The first approach is performing anti-virus' virus signature updates from specified anti-virus software servers, and the second one is the "End-User Security Management with Mobile Agents System" [Yuki] that manages security applications involving anti-viruses. However, the latter approach is still a proposed architecture that has not been implemented yet. Other attempts that have tried to use mobile agents for network security and network management in general are explained in section (2). Unlike the previous systems, the system introduced in this paper will examine the possibility of the idea, introduce a suitable mechanism, and provide a reliable architecture and an initial prototype for the most critically needed functionalities.

1. Related Works

There have been many attempts to use mobile agents in ensuring the security of the network such as intrusion detection systems and network management. Here are a few of these attempts.

In ICSE [Krugel] Sparta group aims to detect security violations in a heterogeneous, networked environment. Its architecture has been designed to a broader range of applications, ranging from network management to intrusion detection. To support their detection algorithm and to address the problem of systems which only offer an implicit way of specifying attack scenarios, Sparta group designed an attack pattern language where only very few data has to be carried by agents during each hop and the detection

algorithm is performed by multiple agents in parallel. This improves scalability, fault tolerance and performance of the system when compared to a client-server variant. Sparta group provided PKI to manage their cryptosystem.

The authors of [Jansen] discussed the ability of using the useful characteristics of Mobile Agents (MAs) like autonomy and mobility in order to build a Mobile Agent Intrusion Detection Systems (MAIDS). They presented the shortcomings of the current IDSs designs and implementations. In addition, they proposed potential solutions offered by Mobile Agents (MAs). However, these solutions may enhance the performance and capabilities of IDSs and Intrusion Response Systems (IRS), but they do not enhance the method of detection or response itself. Although agents in general lack mature agent design methodologies and modeling tools, these problems are likely to be overcome as commercial demand for these products has increased and is eventually satisfied.

A Mobile Agent based Information sharing technique was proposed at [Alassaf]. The authors proposed a secure technique that helps knowledge workers and information seekers around a network. An infrastructure enabling efficient control and monitoring of the MAs and facilitating the users' collaboration and coordination was developed. The security of the technique results from the ability of MA to protect itself by changing its name, putting a password on the MAs which may prevent access to unauthorized information. MAIST consists mainly of two components: a Network Navigation Component (NNC), which has to establish the connection, and a File Handler Component (FHC) which is responsible for information sharing processes.

2. Objectives

The main objective of this paper is to study the ability of constructing a Mobile-Agent-Based system that controls Anti-virus software and introduces a suitable mechanism to do so. In other words, it involves the protection of LAN's nodes against the latest discovered viruses without the need to do this manually by the lab technical administrator.

Briefly, this work aims to implement a Mobile-Agent-Based System using a mobile agent platform that ensures:

Decreasing network traffic caused by connected remote scanning.

Increasing the efficiency by reducing the amount of computer resources needed to run the protection system.
Detecting the latest discovered viruses using an update mechanism.
Performing the scan process according to computers' state not only human order.
Methodology
In this section, four main techniques will be discussed. Each technique has specific features to meet the objectives stated before. The four techniques are:
Using one of the strong existing anti-virus software with the following features:
Periodically virus signature update: within short period.
Less resources consumption compared with others.
Scanning performance.
Selecting anti-virus software: an experiment has been done to compare the selected anti-virus performance with the most competitive one to assess the choice.
Selecting a Mobile Agent Platform that supports:
Java programming language due to the advantages of: Platform Independence, Secure Execution, Dynamic class Loading, Multithreaded Programming, Object serialization, Reflection.
The concept of Mobile Agent's advanced features (mobility, autonomy, reactivity and proactivity).
Using Throw-Away Prototyping model; CBSE will be used beneath some iterations of the used model.

3. Development

This section discusses the development process and its results. The main objective is to build a robust integrated system that gains the benefits from the mobile agent technology in order to control anti-virus software over LAN. The open source ClamWin anti-virus software has been chosen to be the first component of this system, and JADE [URL5] is the platform which our designed mobile agents will work through. Throw-away Prototyping Model will mainly be used in developing the system. As a result, there will be development iterations.

The general ideas of the system mechanism that control the four system agents are: “CA, SCA, PMA, EMA” (See Table 1), and their functions can be summarized as follows,

Controller Agent (CA) controls other agents; it holds the graphical user interface which is supposed to send commands to the other agents in order to work; however, it can work independently sometimes as PMA, and EMA agents can affect the SCA agent to work.

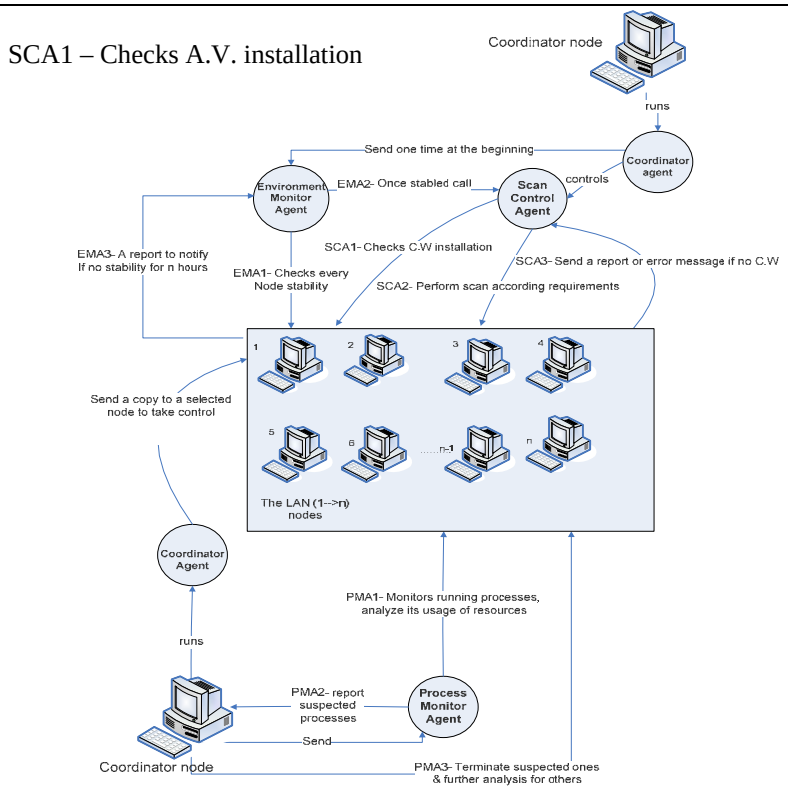
Scan Control Agent (SCA) will check the installation of the anti-virus software. It is sent by CA to perform scan and send back a customized report to the CA about the scanning process to show all types and names of viruses detected, the files infected and their locations in the system. This will be done according to the anti-virus features and capabilities. In case of anti-virus is not installed, an error message is sent back to CA.

A Process Monitor Agent (PMA) resides in each node to keep track of every suspected process, especially those executable files by monitoring and analyzing memory consumption with respect to those safe processes. This is done by using a mechanism of white and black lists of processes. Once there is a suspicious process, a message is sent to the CA and the SCA is sent automatically to that node to invoke the anti-virus software in order to perform the scan process.

An Environment Monitor Agent (EMA) will be sent by the CA parallel with the SCA to a specific node to test whether the node is stable -little resources are consumed- to begin scan. EMA is configured to send the CA a report explaining the environment stability during the day.

To sum up the idea: a multi-agent system with the four agents as listed in Table 1 will be designed. The system architecture and work scenario are briefly demonstrated in Figure 1.

Agent Name	Agent Task
CA	Holds the graphical user interface and coordinates other agents.
SCA	Controls scanning process at nodes.
PMA	Monitors processes' consumption of the system resources and terminates the suspected processes.
EMA	Tests whether a node is stable to begin scan



A.V.: Anti-Virus
EMA (1->n): Environment Monitor Agent work sequence (1->n)
SCA (1->n): Scan Agent work sequence (1->n)
PMA (1->n): Process Monitor Agent work sequence (1->n)

Figure 1: System's Architecture

5.1 - Antivirus Choice Iteration:

The anti-virus software is a critical component in the system; the selection of this component will be discussed in this section.
The CBSE model is followed here with the following sub-stages: searching, analyzing and comparing the pointed out software according to the criteria of the limited resources consumption in a network platform and the ability to be integrated with the selected mobile agent platform.

Two anti-viruses were pointed out: the first is an open source called ClamWin [3] and the second is a closed source called KasperSky [URL6]. An experiment was done to select one of them. According to the results of

comparison, the ClamWin has been selected which provides the required functionality and performance. The experiment and its results are discussed in the next sub-sections.

5.1.1 - Experiment: Comparison of the Performance of KasperSky & ClamWin Anti-viruses

- **The Objective:** To compare the performance of these two anti-viruses:

1. KasperSky (kis7.0.1.321): as one of the leading anti-viruses (ranked as the second anti-virus for the top-ten anti-viruses [URL7].
2. ClamWin (0.91.2): as a free open source anti-virus.

- **Constants:** Same circumstances in terms of hardware and software conditions, besides the scan process which is done using the kasperSky in one turn and the ClamWin in the next turn on the same folder.
- **Experiment:** Each of these two anti-viruses is run separately, and the following tests are done for each of them:

1. Run the anti-virus and determine an infected folder to be scanned.
2. The CPU usage and Memory usage are determined every 30 seconds for 10 minutes based on the operating system's task manager: as shown in Table 2, Figure 2 and Figure 3.
3. Ability to be integrated with mobile agent.
4. The following values are determined: Elapsed Time. Number of detected viruses. Number of deleted viruses.

- **Results:**

1. Kasper Sky:

A- CPU & Memory usage (each row of values determined for 30 seconds).

B- After scanning is finished:

– Elapsed Time: 12min & 33sec.
– Number of detected viruses: 114 viruses.
– Number of deleted viruses: 114 viruses.

2. Clam Win:

A- CPU & Memory usage (each row of values determined for 30 seconds).

B- After scanning is finished:

– Elapsed Time: 14min & 45sec.
– Number of detected viruses: 114 viruses.
– Number of deleted viruses: 113 viruses

- **Analysis and results:**

Back to the research objective the minimum resources consumption has the high priority then the detection takes place. KasperSky **(K)** and ClamWin **(C)**Anti-Viruses are both very successful in detecting and deleting viruses, but KasperSky has more resources consumption compared to ClamWin. So, ClamWin anti-virus is suitable to be used in this system for its advantages in the resources usage (an important consideration), and for its ability to be integrated with mobile agent technology.

Table2:CPU& Memory usage values in KasperSky (K) and ClamWin (C)

#	CPU usage (%)		Memory usage (KB)	
	K	C	K	C
1	30	37	34	33
2	21	1	50	33
3	14	10	50	33
4	3	29	44	34
5	28	25	43	34
6	4	22	43	34
7	74	38	75	34
8	65	10	97	34
9	60	42	97	36
10	74	29	99	34
11	64	23	97	34
12	30	30	68	34
13	59	35	152	35
14	65	32	173	34
15	76	27	184	34
16	50	12	137	34
17	55	23	152	34
18	60	19	98	34
19	77	9	190	34
20	43	5	83	34
AVG	**47.6**	**22.9**	**98.3**	**34**

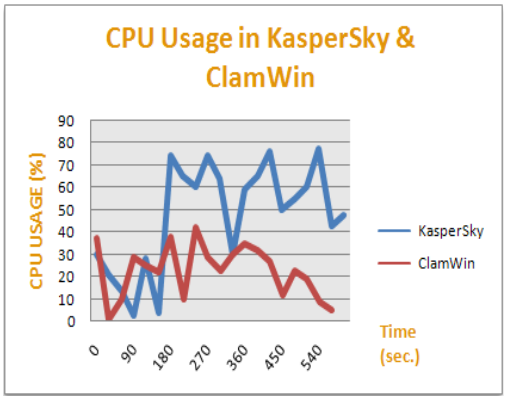


Figure 2: CPU Usage in Kasper & ClamWin Diagram

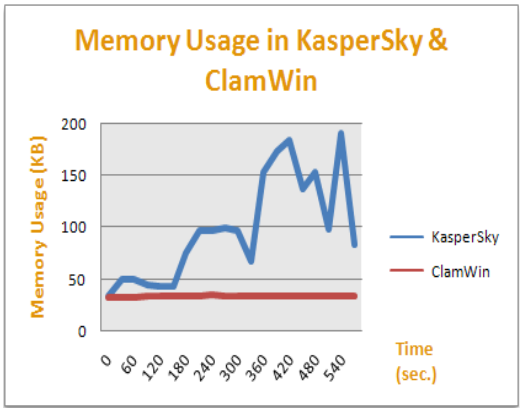


Figure 4: Memory Usage in Kasper & ClamWin Chart

5.2 - Mobile agent platform choice stage:

Mobile agent platform is also a critical component in the system; the selection of this component will be discussed in this section.

The CBSE model used here has the following sub-stages: search, analyze and compare the pointed out platforms which match the system requirements. Many mobile agent platforms pointed out: JADE [URL5], Odyssey, Voyager, Concordia [URL8], Aglets [URL9].

JADE simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. In addition to overcoming system requirements, its features include: (1) The

agent platform can be distributed across machines which do not even need to share the same OS. (2) The configuration can be controlled via a remote GUI. (3) The configuration can be even changed at run-time by moving agents from one machine to another one as and when required.

5.3 - Components Integration Iteration

Components integration is considered as the most critical iteration in such a system development as it would allow communications between the chosen components. ClamWin command-line-based ability leads to the real communication scheme within the application components. The Glue Code is written in JAVA in order to integrate the two technologies. A sample of the Glue Code is shown in Figure 4:

```
import java.util.*;
import java.io.*;
public class GlueCode {
   public static void main(String[] args) throws IOException{
      Runtime.getRuntime().exec("cmd /c start C:\\\"Program
Files/ClamWin/bin/clamscan.exe\" --verbose --quiet --infected --
show-progress --log=C:\\reports.txt --copy=C:\\copied --
database=C:\\\"Documents and Settings/All Users/.clamwin/db \" -
-recursive C:\\viruses/ ");
 Runtime.getRuntime().exec("cmd /c start c:\\reports.txt"); }}
```

Figure 3: Glue Code to integrate Clamwin and Mobile Agent system

5.4 - System Functions Iterations

This section provides a detailed description of the methods followed to develop the system's four agents: Coordinator Agent, Scan Control Agent, Process Monitor Agent and Environment Monitor Agent.

5.4.1 - Coordinator Agent (CA)

- **Specification:**

This agent is designed to be responsible for coordinating to somewhat the whole system. It runs on the "Coordinator" node as a graphical interface agent. It receives messages from the other available connected nodes.

It has the ability to choose a connected node in one of the available labs within the LAN and send a copy of itself running gaining portability characteristics.

- **Development:**

The real development is done on Computer Labs involved in this project as a case study; these labs are Lab A, Lab B and Lab C. Once a lab is selected a scan agent is created then a copy of it is sent to each connected node in this lab. When the scan process at one of the connected nodes is completed, a message returns back to the "Coordinator Agent" to label this node with whether the scan operation is done successfully or not, Figure 5 is the interface of (CA).

Besides, the (CA) supports the portability since it has the ability to choose a connected node in one of the available labs and send it a copy of itself, so any connected node could be a Coordinator Node.

- **Validation:**

CA is partially developed and tested successfully. As the process of sending the "Scan Control Agent" to every node in the selected lab encountered many errors, we have succeeded to send a "SCA" just to one selected node. On the other hand, we have succeeded to develop and test the portability function.

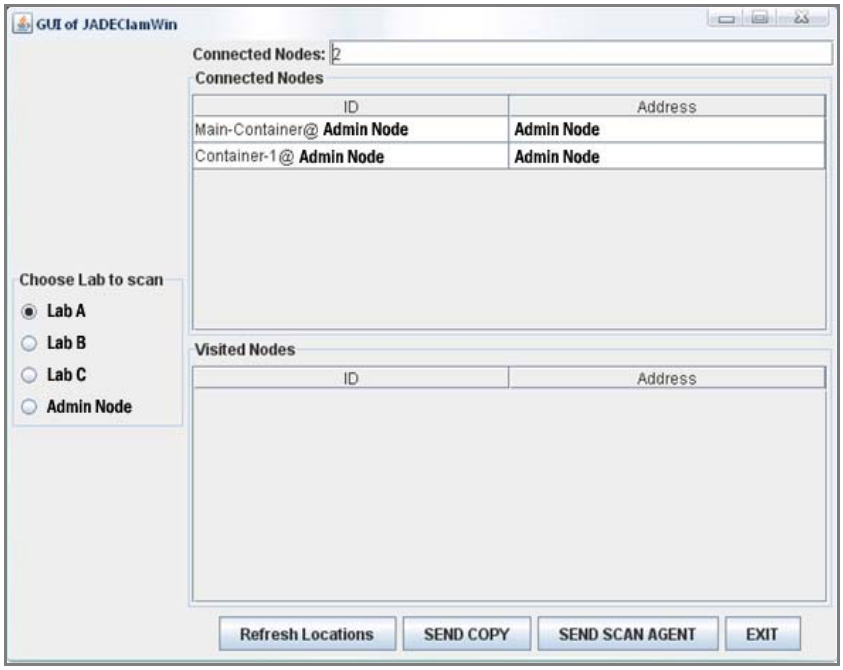


Figure 4: Interface of CA

5.4.2 - Scan Control Agent (SCA)

- **Specification:**

This agent is designed to be responsible for invoking the anti-virus to perform the scan process on the target node.

- **Development:**

This agent is created by the “CA” when a lab is selected to scan its connected nodes. A copy of “SCA” agent travels to every connected node of the selected lab; once the scan process is completed, it sends back a message whether the scan process is done successfully or not. When the task of this “SCA” is done, the "SCA" is killed by itself.

- **Validation:**

The "SCA" is developed and tested successfully.

5.4.3 - Environment Monitor Agent (EMA)

- **Specification:**

This agent will work to test the environment stability in terms of CPU and Memory consumption to decide whether to call the SCA or not, so we need a suitable programming language for this function to be integrated with the agent development code.

- **Development:**

In fact, this agent has not been implemented yet, but the main idea is to use a Java Code that helps in finding out memory and CPU consumption for all running processes.

- **Validation:**

The work on this agent wasn't completed to do the required validation.

5.4.4 - Process Monitor Agent

- **Specification:**

This agent role is to monitor and detect suspicious processes in the system to provide a real time protection. Since java has the ability to execute command lines as part of its code, also JADE is a Java platform which is planned to include some JAVA commands in the monitor agent’s work.

- **Development:**

In order to develop PMA, we tried the following two functions:

1. Obtain the tasks list which shows the active processes at a current time.
2. Check for Scheduled Applications, since it is possible (although rare) that malware may try and use the Windows scheduler service to launch an unauthorized application [URL10].

The work started on the first function. The aim was to get the active processes at the current time. This is an important point to determine whether the operating system files were introduced or modified by the attacks. By checking the changes in some areas, the infected systems are likely to have new processes introduced into their memory. This required us to save the list of tasks (processes) at the current time in order to save it in a specific location for further malware analysis. So, we found that executing the command (tasklist /v >TaskList.txt) will create a file called "TaskList.txt" in the current working directory that could be used by this agent.

4. Results

The Objectives were almost achieved:

1. The network traffic decreased by using JADE as a mobile agent environment for System's Agents to work within.
2. The efficiency increased by reducing the amount of computer resources needed to run the System as an anti-virus Software. 'ClamWin' used proved its performance advantage compared with the powerful one 'Kasper Sky'.
3. On the other hand, it's possible to perform automated scan process according to computers stat**e not only human order.**
4. The system succeeded to detect the latest discovered viruses by using an update mechanism.

But some obstacles appeared. These obstacles include:

1. Most Mobile Agent Platforms lack FIBA standardization and many weren't available to be used.
2. Most agent development tools are still new with security bugs and vulnerabilities that are yet unknown, as most of the platforms are 'Alpha' or 'Beta' versions.
3. No similar mobile agent based systems for controlling anti-virus was available while working on this system.
4. Getting the benefits of Mobile Agents Characteristics is partially achieved in the system's prototype, but totally achieved in our proposed system as shown in Table 3.

Table 1: Evaluation of the Mobile Agents Characteristics Achievement

#	Mobile Agent Characteristic	System's Prototype
1	Mobility	Yes
2	Autonomy	Yes
3	Asynchronous	Yes
4	Local interaction	Yes
5	Disconnected operation	Yes
6	Parallel execution	No
7	Intelligence	No
8	Communication	Yes
9	Cooperation	No
10	Adaptation	No
11	Persistence	No
12	Goal-Orientedness	Partially Yes
13	Loyalty	No

An evaluation is made of our proposed system compared to "End-User Security Management with Mobile Agents" System; this proposed system discussed the ability of using mobile agents for network security management. The differences are illustrated in Table 4.

Table 2: Evaluation of the Introduced System Prototype Compared to Others

	"End-User Security Management with Mobile Agents" System	Our proposed system
Area of Protection	Intrusion Detection Firewall Anti-virus	Anti-virus
Scan Process	Done by a scan agent , managed by agent manager	Using ClamWin and executed by Scan Agent which determines the parameters of ClamWin's scan process.
Implementation	Not implemented, only a proposed architecture is done.	Some functions of the original architecture are successfully implemented

5. Business Benefits

As a positive result of the ability of Mobile Agents to be used in controlling anti-viruses software over LANs, it's also applicable to build such a system with such objectives achieved in the previous results.

Furthermore, this system will be very helpful when it is used in scalable huge networks such as universities and companies as it reduces time and effort needed by the lab technical administrator to manage nodes' protection mechanism. It also extends the scanning process that will be automated according to nodes' state of resources in a way that increases the efficiency of such a system and process in addition to the human scanning order.

6. Conclusions

This paper has discussed the ability of using Mobile Agent Technology in controlling an anti-virus over LANs. A suitable architecture was proposed to control ClamWin Anti-virus via mobile agents running within JADE Mobile Agent Platform.

Getting benefit of Mobile Agents characteristics is a competitive advantage in network management. Regardless of its tool obstacles, the researchers have partially succeeded in implementing some of the system's functionalities using the Prototype Model with CBSE. Depending on the results of this paper, the researchers recommend the use of mobile agent in controlling anti-virus SW over LAN with respect to the issues raised in the next paragraph.

As a future work, the proposed system will be extended to a complete Mobile-Agent-Based protection system by involving: intrusion detection agent, firewall agent … etc. Achieving security for both agents against malicious hosts and vs. by these facilities, such a system will be fully applicable in LAN's security management and required functionality.

References:

1- [Alassaf] Alassaf N., Obeid N., Salah I.,(September 2009). "Facilitating Information Sharing Using Mobile Agents", European Journal of Scientific Research, 36(2),145-153,Jordan, Available from :
http://www.eurojournals.com/ejsr_36_2_01.pdf

2- [Jansen] Jansen W., Mell P., Karygiannis T., Marks D.,(October 1999). "Applying mobile agents to intrusion detection and response",National Institute of Standards and Technology, Computer Security Division,. Available from:
http://csrc.nist.gov/publications/nistir/ir6416.pdf

3- [Krugel] Krugel S., Toth T., (2001).“Applying mobile agent technology to intrusion detection”. ICSE Workshop on Software Engineering and Mobility, Available from
http://www.auto.tuwien.ac.at/~chris/research/doc/2001_01.pdf

4- [Yuki] Yuki K., Toshihiro T., Kouichi S., (2004). "End-User Security Management with Mobile Agents", Kyushu University, Japan.
5-[URL1] EMCO Network Malware Cleaner, [Last visit: 08 15, 2008], from (emco.is):
http://www.emco.is/products/network-malware-cleaner/features.php
6- [URL2] ClamWin, [last visit: 03 19, 2008], from wikipedia.org:
http://en.wikipedia.org/wiki/Clamwin
7- [URL3] Antivirus Ranking. [last visit: 06 03, 2008], from
http://www.che-mi.com/2007/05/11/new-antivirus-ranking
8- [URL4] Clam AntiVirus, [last visit: 03 18, 2008], from (wikipedia.org):
http://en.wikipedia.org/wiki/Clamav
9- [URL5] JADE Framework, [last visit: 09 15, 2008], (from jade.tilab.com):
http://jade.tilab.com/home-index.htm
10-[URL6] Kaspersky, [last visited: 03 22, 2008], from (wikipedia.org):
http://en.wikipedia.org/wiki/kaspersky
11- [URL7] Top Ten reviews, [last visit: 09 15, 2008], from (anti-virus-software-review.toptenreviews.com)
12- [URL8] Mobile Software Agents, [last visit: 08 20, 2008], from (comsoc.org):
http://www.comsoc.org/ci/private/1998/jul/Karmouch.html
13- [URL9] Aglets, [last visit: 09 03, 2008], from (wikipedia.org):
http://en.wikipedia.org/wiki/Aglets
14- [URL10] Antivirus Defense-in-Depth Guide, [last visit: 07 09, 2008], from (microsoft.com):
http://www.microsoft.com/technet/security/guidance/serversecurity/avdind_4.mspx