Mobile Agents Security Issue: Malicious Hosts Attack

Wahyuning Diah
weezery2002 at yahoo dot com dot au

1. INTRODUCTION

In general, the term of agent means a person that does a task on behalf another person that assigns it to (representative of). The example can be a real estate agent, a travel agent, an insurance agent and many more. The reason why someone delegates that task is to make the task easier and be completed soon. For example, if someone wants to buy a house, he can search the house through the real estate magazine, find out about the house by calling the owner and then see it. Instead of searching the house one by one, he can come to a real estate agent and choose several interesting houses and find out all the information about those houses from the real estate agent. This information finding agency is really a big help, since this saves much of his time and money.

In distributed computing area, there is a concept called mobile agent. The force driving behind the mobile agent technology is the growth of Internet. The Internet ability to connect millions of computer in the whole world makes it to be the perfect place to distribute data among them, and the mobile agent technology is one of the ideas to make the distribution data becomes more valuable (Murch and Johnson 1999 p.14). There are some other benefits that enforce the development of mobile agent such as: it reduces the communication cost in the network by filtering data transferred from the server (Baumann et al. 1997), it can be used to solve complex and big problems that needs distributed computing (Nwana and Ndumu 1997 p.8) and also it can act as a personal assistant that can learn from the user or from other agents (Maes 1994).

The agent in this concept also holds the same meaning with the above general agent definition. As for the mobile agent definition, there are a number of definitions that describe about it. These differences are much related to the researcher's vision about mobile agent implementation on his/her work field. Two definitions of agent related to this paper will be presented here:

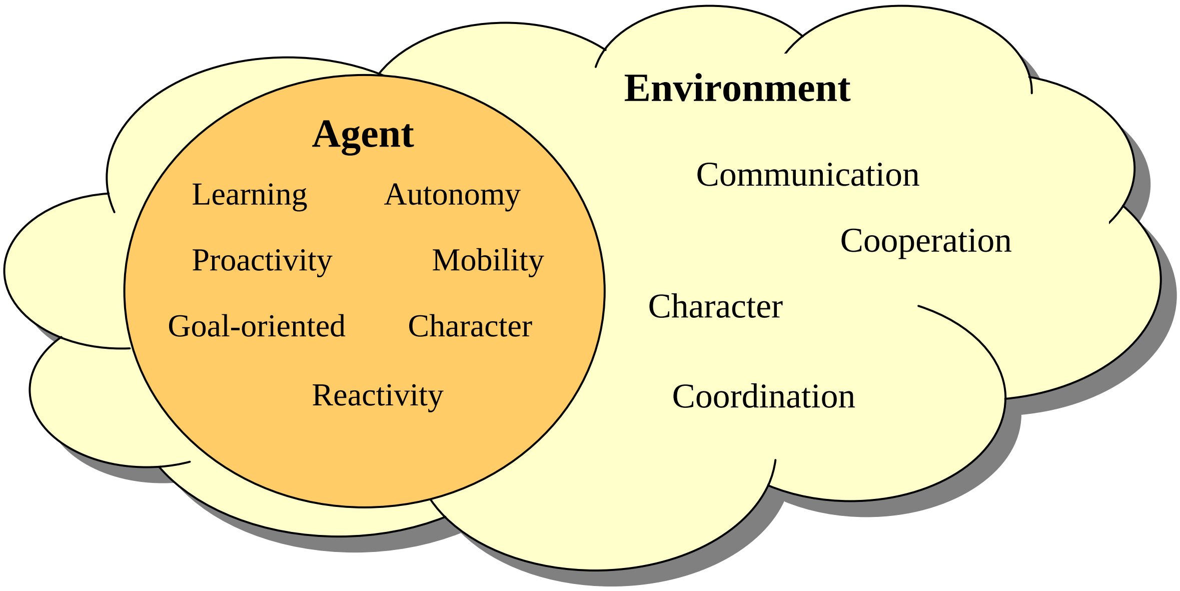
Michael Coen in (Franklin and Graesser 1996) describes agent as
"Software agents are programs that engage in dialogs [and] negotiate and coordinate transfer of information."

The IBM Agent definition in (Franklin and Graesser 1996) is
"Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires."

The first and second above definitions mainly states that agent is a kind of messenger that carries information and set of operation in order to perform the user's task. The agent can travel from one host to another host, perform a task and then return back the output to the user.

Besides those two definitions, mobile agent also has characteristics, which are divided into two groups, internal properties and external properties (Brenner et al. 1998 p.23). The properties of agent can be summarized in the following figure.

Figure 1. Characteristics of agent (Brenner et al. 1998 p.23)



The internal properties that in the above figure labeled agent, illustrate the kinds of actions that an agent can perform (learning, autonomy, proactivity, mobility, goal oriented, character and reactivity). Environment that labels external properties, include communication, cooperation, coordination and character. The external properties are the agent characteristics that relate to agent interaction with other agent or the user. From above figure, character property is assigned to two groups because it provides trustworthiness, honesty, reliability appearance (internal characteristic) that is used to deal with other agents so that trusted communication can be achieved (external characteristic).

One of the applications that make use of Internet and also mobile agent technology is e-commerce. The agent can support secure electronic transaction, buyer-merchant negotiation and on-line auctions. Therefore, the security of agent and the host becomes a big issue in the distributed mobile computing. Since the agent that carries data and set of operations may face some problems in its travel to another host. This paper presents the kinds of attacks that may happen to agent in Section 2. In Section 3, the discussion will be more specific to host-to-agent attack, malicious host attack. In the next section, it covers several approaches to prevent malicious host attack and focuses on Time Limited Blackbox Security. Section 5 concludes this paper.

2. SECURITY THREATS

In general, Lange and Oshima (1998) discuss that the attacks to mobile agent can be divided into 2 types: active and passive attack. The active attack is when the agent changes. But in passive attack, the agent and the carried information remain the same.

In this paper the word host and platform is interchangeably. Those two words mean the environment in which an agent can execute its operations. The host where the agent comes from is called home host/platform for the agent.

According to Jansen and Karygiannis (National Institute of Standards and Technology), security threats in an agent can be divided into four categories:

2.1. Agent-to-Platform (Agent-to-Host)

An agent can cause harm to the host where it arrives. This kind of attack can be classified as:

2.1.1. Masquerading

When an agent pretends to be an authorized agent, this kind of attack is called agent masquerading. This fake agent, which does not have the privilege, tries to access on services and resources. This action can cause the legitimate agent losses its reputation in the society.

2.1.2. Denial of Service

One of the causes of this attack is the programming error. The programmer may intentionally write code that make the agent performs a task that needs a great amount of resources that the visited host cannot handle.

2.1.3. Unauthorized Access

The host should provide access mechanism to identify the incoming agent (i.e. authentication) so that the agent is only allowed to access services and resources within its privilege.

2.2. Agent-to-Agent

When an agent travels to another hosts, it may interact with another agent on the destination path. The incoming agent can cause harm to the agent that lives in that host. This agent-to-agent attack can be classified as:

2.2.1. Masquerading

The agent can try to deceive another agent by pretending an authorized agent and asking for valuable information such as bank account and credit card number. This fake agent harms the authorized agent and the agent that has been deceived.

2.2.2. Denial of Service

One of the examples of this attack is when an agent tried to block or interrupt other agent in completing its task.

2.2.3. Repudiation

The repudiation is a condition when an agent denied the transaction that has been occurred. The host should keep the transaction record detail so that when the agent tried to deny a transaction, it can show enough evidence to prove it.

2.2.4. Unauthorized Access

The host where the home agent lives should be made secure. It is a way to prevent a visiting agent attack. An attack that happens may change the home agent's data or code or turn it into a malicious agent.

2.3. Platform-to-Agent (Host-to-Agent)

On the travel to perform the task, an agent may arrive to un-trusted hosts. These hosts can do harm to the visiting agent. This kind of attack can be classified as:

2.3.1. Masquerading

The host pretends to be a trusted host by faking its address so that the visiting agent will think that it is the right destination path. Soon the agent arrives at a fake host, the host tries to steal important information that is carried by the agent.

2.3.2. Denial of Service

The denial of service attack may happen when the host does not try to complete the visiting agent's task. The host ignores the request services asked by the agent and it makes the agent is waiting for a response from the host. This condition can lead to a deadlock situation.

2.3.3. Eavesdropping

The eavesdropping attack happens when the host can monitor the communication and the control flow of the agent program. The host can keep the sensitive information and share it with other agent that is not on the visiting agent list.

2.3.4. Alteration

The alteration problem may arise when agent has to move to several hosts to complete its task. It is called “multi hop” problem. The agent should only visit trusted host environment so that the final result of its task can be the correct agent’s goal. Once the agent move to another host, it is hard to track down the action that has been done in the previous host.

2.4. Other-to-Platform (Other-to-Host)

Other in this category can be stated as external entities that include agents or hosts. For example, an agent attacks another agent in different hosts. This section is much related to network communication problem. This kind of attack can be classified as:

2.4.1. Masquerading

The service request asked by an agent can be located locally or remotely. The agent can deceive a remote host to ask for a particular service that should only be provided to an authorized agent.

2.4.2. Denial of Service

The communication between a remote host and an agent can be disrupted by the denial of service attack. This could be caused of the communication protocol error.

2.4.3. Unauthorized Access

The access of services provided by the host should only be given to the authorized agent or user. The distributed computing system may lead to a remote maintenance host that should only be given to the authorized administration staff.

2.4.4. Copy and reply

An un-trusted party may intercept information that the agent carried while it moves from one host to another host. This information can be copied and then replied to agent as if it comes from the trusted party.

3. MALICIOUS HOSTS ATTACK ON MOBILE AGENTS

In this paper, malicious host attack can be considered on the platform-to-agent (host-to-agent) attack that has been discussed in the previous section. Malicious host can be described as a host that can attack an incoming agent that tries to access it (Hohl 1998). The attack that occurs can be categorized in passive or active attack. In order to avoid malicious hosts, the agent should only travel to trusted hosts.

The word 'trust' in the buying and selling activities is a very important point to consider. The customer will not buy from the user if the customer cannot trust the seller. That goes the same in term of trust in agent and shop online relationship. The reputation of the host can be one of the means to establish trust (Jansen and Karygiannis National Institute of Standards and Technology). Another 'trust' approach can be introduced from the organizational solution (Hohl 1998). Only trusted parties can operate host, therefore it is not every company can open a host. General Magic applies the example of this approach on its agent PersonaLink using the third party company AT&T.

Hohl (1998) identifies 12 kinds of attack that might occur in this host-to-agent attack. The description of these attacks will be discussed using this two sample codes that are taken from Hohl (1998). In general, Hohl (1997) discusses the codes below as the agent that will request a service from list of providers or shops online for BuyFlowers. The agent will act on behalf its user to find the cheapest roses price among the available shops. The agent will travel from one shop to another shop to get the price of the flowers and compare with its own money and the lowest price so far. The agent will store the shop address with the cheapest price and visit it to buy the flowers.

The data block that will be carried by the agent is as the following:

```
Address home = “PDA, sweet PDA”
Money wallet = 20$
float maximumprice = 20.00$
good flowers = 10 red roses
Address shoplist[] = empty list
int shoplistindex = 0
float bestprice = 20.00$
Address  bestshop =empty
```

The host in which the agent arrives will execute the following procedure:

```
public void  startAgent()
{

        if (shoplist == null)
        {
                shoplist = getTrader().
                   getProvidersOf (“BuyFlowers”);
                go (shoplist[1]);
                break;
        }

        if  (shoplist[shoplistindex].
                askprice(flowers) < bestprice
                {
                        bestprice = shoplist[shoplistindex].
                                        Askprice(flowers);
                        bestshop = shoplist[shoplistindex];
                }
        if (shoplistindex >= (shoplist.length – 1))
        {
                // remote buy
                buy(bestshop, flowers, wallet);
                // go home and deliver wallet
                go (home);
                if  (location.getAddress() = home)
                {
                        location.put(wallet);
                }

        go (shoplist[++shoplistindex]);

        }
}
```

The malicious hosts attacks to the agent can be as the following (Hohl 1998):

1. Spying out code
   The code that is carried by the agent should be readable to the host since the host will execute the code. It means that the host can find out the whole steps that will be performed by the agent to accomplish its task. The host also can find out the whole classes that will be performed by the agent in another hosts. It means that the host can guess what kind of output that is expected from the task.

2. Spying out data
   From above data, money variable is the sensitive information that a host can see. This means a host can compare the available money and the flowers that the agent is looking for. The host can use the maximum price and best price variable to offer a bit cheaper price from the best price offered so far in the list, even though the original price in that host should be much cheaper than that.

3. Spying out control flow
   The control flow of the code can represent the next step that will be performed by the agent. From the code above, we can find out the state of the agent right now. It will be seen whether the agent already found the best offer from the best price available.

4. Manipulation of code
   If the host can get into the code memory, it can modify the agent code. For example, the hosts will point out the agent to a particular flower shop without paying attention on the price offer.

5. Manipulation of data
   The host can modify the data by cutting down the shop list data after the host put the local shop as the best offer shop.

6. Manipulation of control flow
   The manipulation of the control flow from above code can be carried out in the second or third if statement. The host can force the agent to choose the host recommended shop as the preferred shop.

7. Incorrect execution of code
   The host can also modify the execution of the code in which the result will be the same as the above attack.

8. Masquerading of the host
   When an agent travels, the host can pretend to be the right destination host so that the agent will visit it. Soon the agent comes to the host, the process may lead to read attack of the agent code and data by the host.

9. Denial of execution
   The denial of execution of the host can be a trick to keep the agent stays in the host. It may be because the host knows the time limited of another host that has a best offer. Soon the best offer of another host is expired, the host can use this chance to put itself as the best offer.

10. Spying out interaction with other agents
    If the host can spy on the buy transaction between an agent and the remote host and it knows about the money that the agent carried, the host can spend the stored money.

11. Manipulation of interaction with other agents
    When the host can manipulate the interaction with other agents, the host can redirect the agent to go to another host and buy something there.

12. Returning wrong results of system calls issued by the agent
    In the code above, when the agent execute this statement `if (location.getAddress() = home)`, the host can pretend to be the home so that the agent will deliver the wallet to the host.

4. APPROACHES TO PREVENT MALICIOUS HOSTS ATTACK

Gray (et al. 1998) discuss that one of the most difficult problem in mobile agent security is protecting an agent from malicious host. In their paper Jansen and Karygiannis (National Institute of Standards and Technology) present the possible techniques that can be used to protect agent from malicious hosts. The techniques are as the following:

4. 1. Partial Result Encapsulation

This method can be used to prevent agent from malicious hosts by encapsulate the result that carried by the agent. Jansen and Karygiannis (National Institute of Standards and Technology) propose three kinds of approach on this technique:

1. Give an agent facility to encapsulate its information
2. Give a host a means to encapsulate the visiting agent information
3. Coordinate with other trusted third party to timestamp the agent result

If the agent has the ability to encapsulate the information, when it visits a host it can encrypt the information using its public key and decrypt it using private key that is maintained in the visited host.

The second approach that can be done is by creating a chain that encapsulates the result from previous hosts and the other hosts that will be visited next. Each host signs the result using its private key and keep the hash function to bind the results.

The last approach in partial result encapsulation is using trusted third party that will sign the result of an agent when it visits a host.

4. 2. Mutual Itinerary Recording

The Mutual Itinerary Recording involves two agents at least. The first agent is the one that moves from one host to another host and the second agent is the cooperating agent that will track and record the itinerary of the first agent. The first agent sends its itinerary (past, current and next) through an authenticated channel to the second agent (Jansen and Karygiannis National Institute of Standards and Technology). When there is inconsistency itinerary detected from the first agent, the second agent will take an appropriate action to tell the first agent.

4. 3. Itinerary Recording with Replication and Voting

Fred Schneider in Bierman and Cloete (2002) proposes this technique to prevent malicious host attack. The idea is to use multiple copies to do computation instead only one copy to perform computation. On each stage of the computation, the host makes sure that agent carries out valid information. The host will know the previous hosts involved at a certain stage of computation. Therefore, the host will propagate onto the next stage with the replica agents it considers valid.

4. 4. Execution Tracing

In Prabhu (Oregon State University), “execution tracing is a technique for detecting unauthorized modifications of an agent through the faithful recording of the agent's behavior during its execution on each agent platform”. The host needs to create a log that store the operations performed by the agent while it stays there. Then the log will be stamped with the cryptographic hash function to make it readable only for trusted party.

4. 5. Environmental Key Generation

Riordan and Schneier (1998) introduce this technique to overcome the traditional cryptographic systems that "do not depend upon temporal, spatial, or operational conditions". The environmental key generation has the same idea with ephemeral keys, the keys that are created when the task starts and destroyed soon the task finishes. The agent when arrives on an environment condition (e.g., string match in search), a key is generated. This key will be used to decipher the cipher text, therefore this technique ensures that the host cannot directly read and execute the code carried by an agent.

4. 6. Computing with Encrypted Functions

The Encrypted Function method is a way to prohibit the executing host to find out about the sensitive information carried by an agent. Sander & Tschudin (1998a) propose the method that works like this:

(1). Alice encrypts f.
(2). Alice creates a program P(E(f)) which implements E(f).
(3). Alice sends P(E(f)) to Bob.
(4). Bob executes P(E(f)) at x.
(5). Bob sends P(E(f)) (x) to Alice.
(6). Alice decrypts P(E(f)) (x) and obtains f(x).

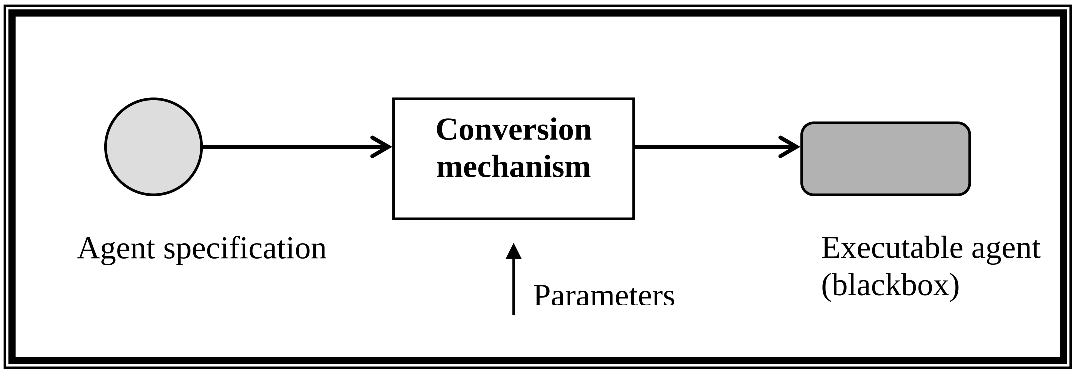
In the mobile agent environment, the host can execute the program that is encrypted using a function without knowing the original function that builds the encrypted program.

4. 7. Time Limited Blackbox

Blackbox in this approach means an agent that is already converted into a different structure of agent but still perform the same task with the original agent. Blackbox can be used to protect agent from the malicious host attack (read and manipulation) because the original code has been scrambled into an executable code. According to Hohl (1998), an agent is called to be a "blackbox", if it meets the following criteria: code and data of the agent specification cannot be read, be modified at anytime.

There is a "conversion mechanism" in converting an agent into an executable agent (blackbox). The conversion uses parameter so the different executable code can be generated for the same agent system. The Figure 2 below shows the agent conversion:

Figure 2. An agent conversion into blackbox



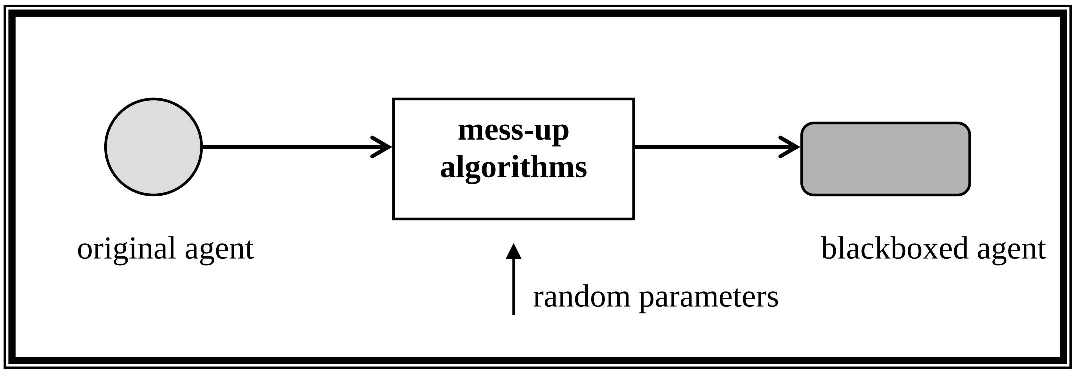
Hohl (1998) proposes an algorithm that can be used to fulfill the blackbox approach, Mobile Cryptography, which is introduced by Sander & Tschudin (1998b). The Mobile Cryptography is "the study of mathematical techniques related to aspects of information security of mobile executable code in a network" (Sander & Tschudin 1998b). This algorithm implements encrypted program, which encrypts data using set of operations (program) in order to make it harder for the attacker to break the code.

The restriction of this algorithm is only can be applied to agent and trusted hosts' communication. Therefore Hohl (1998) redefines the blackbox property definition in a way that the protection of the blackbox is not forever (anytime) but only for a limited amount of time. This method is called Time Limited Blackbox Security. The idea is the blackbox carries its expiration date, which is stated the time interval for an agent to be considered valid. The time interval is formulated so that it is impossible for the attacker to break the blackbox and manipulate the agent.

The redefining of blackbox protection (Hohl 1998), holds the following properties: code and data of the agent specification cannot be read, be modified *for a certain amount of time* and after the protection agent can be attacked but the attack will not harm the blackbox.

The conversion algorithm that will be used to create a time-limited blackbox in this approach is called obfuscating or mess-up algorithm. This algorithm is built in a way that it is hard to be analyzed by the host in a reasonable amount of time. Another way to reach the time-limited blackbox protection is by not allowing the attacker to create a mental model of the visiting blackbox's code. The mental model is the "understanding" of semantic and flows of the program. Figure 3 below shows this approach.

Figure 3. Agent conversion into time limited blackbox



The blackbox that is expired can have the new expiration date from trusted party (Hohl 1998). This model allows blackbox to finish the task.

There are costs that should be taken into consideration before applying an agent to use this method. The four classes of costs that are identified are (Hohl 1998):

1. Cost at creation time, the cost of converting the original agent into a new executable agent.
2. Cost at transmission time, the agent will carry all its libraries to perform the task; therefore it will be quite a big size agent.
3. Cost at execution, the mess-up algorithm will need some time to perform the calculation.
4. “Cost” by not using efficiency enhancing method, the approach is not a modular method since it needs new code to mess-up agent even for the same functionality agent.

5. CONCLUSIONS AND FUTURE RESEARCH

The growth of Internet applications that implement mobile agents make the security is one of the crucial issues. The security of agent can be divided into four categories, which are: agent to host, agent to agent, host to agent and other to host attack. This paper describes the host to agent security, where one of the possible attacks to the agent is called malicious hosts attack. One of the solutions to prevent the malicious hosts attack is Time Limited Blackbox.

The main characteristic of agent, mobility, allows agent to move from one host to another host to perform its task. This activity puts the agent on a risk of malicious host attack when it arrives on untrusted host. The idea of protecting an agent from the malicious host has been discussed that has resulted the main point which is to keep the agent code secure without the ability of the visited host to modify or manipulate the code. However, the visited host of the agent is naturally be able to access the code since the host is responsible to control the execution of the agent's code when the agent requests for services provided. Therefore, further researches on limiting the host for accessing agent's code should be applied. The host should only be able to read the code related to task that would be performed on that host and the remaining code should be locked with encryption so that only the authorized host that can access it.

The Time Limited Blackbox approach is mainly proposed only for agent that transports cash value or secret key. It is because the creation, transmission and execution costs that should be considered to apply this model. Some other points that should be clearly defined are the appropriate length of time interval, the protection of expiration date carried by the blackbox and also the further research on kinds of code that should be implemented on the agent's code so that the attacker will not find it easy to break the code by generating mental code in a short of time. The protection of expiration time should be applied for example by using public key cryptography system where the blackbox encrypts it using its public key and the visiting host will decrypt it using its private key. The protection of expiration time is important so that the malicious host cannot modify it. Once the malicious host modifies the expiration time, it can set it into specified time that will support the attack to the blackbox.

REFERENCES

Baumann, J., Hohl, F., Rothermel, K., and Straßer, M. 1997. Mole - Concepts of a Mobile Agent System. IPVR (Institute for Parallel and Distributed High-Performance Computers). Fakultät Informatik, Universität Stuttgart. Available: http://www.informatik.uni-stuttgart.de/ncstrl-documents/TR-1997-15/TR-1997-15.pdf (Accessed: September 29, 2002)

Bierman, E., and Cloete, E. 2002. "Classification of Malicious Host Threats in Mobile Agent Computing". SAICSIT'2002 proceedings. Available: http://osprey.unisa.ac.za/~elsabe/download/MAsecurity7.pdf (Accessed: October 5, 2002)

Brenner, W., Zarnekow, R., and Wittig, H. 1998. "Fundamental Concepts of Intelligent Agents", *"Intelligent Software Agents: Foundations and Applications"*. Springer-Verlag, Berlin, pp.19-34.

Franklin, S and Graesser, A. 1996. Is it an Agent, or just a Program? : A Taxonomy for Autonomous Agents. Proceedings of Third International Workshop on Agent Theories, Architecture, and Languages, Springer-Verlag.
Available: http://www.msci.memphis.edu/~franklin/AgentProg.html (Accessed: August 7, 2002)

Gray, R. S., Kotz, D., Cybenko, G., and Rus, D. 1998. "D'Agents : Security in a Multiple-Language, Mobile-Agent System", " *Mobile Agents and Security"*. (Ed.) Giovanni Vigna. Lecture Notes in Computer Science. Springer-Verlag, Berlin, pp. 154-187.

Hohl, F. 1997. An Approach to Solve the Problem of Malicious Hosts. IPVR (Institute for Parallel and Distributed High-Performance Computers). Fakultät Informatik, Universität Stuttgart. Available: http://ncstrl.informatik.uni-stuttgart.de/Dienst/Repository/2.0/Body/ncstrl.ustuttgart_fi/TR-1997-03/pdf (Accessed: September 27, 2002)

Hohl, F. 1998. "Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts", " *Mobile Agents and Security"*. (Ed.) Giovanni Vigna. Lecture Notes in Computer Science. Springer-Verlag, Berlin, pp. 92-113.

Jansen, W., and Karygiannis, T. NIST Special Publication 800-19 – Mobile Agent Security. National Institute of Standards and Technology. Computer Security Division. Available: http://csrc.nist.gov/publications/nistpubs/800-19/sp800-19.pdf (Accessed: September 24, 2002)

Lange, D. B., and Oshima, M. 1998. "Aglet Security", *"Programming and Deploying Java Mobile Agents with Aglets"*. Addison Wesley Longman Inc, Reading, pp. 171-185.

Maes, P. 1994. 'Agents that Reduce Work and Information Overload', *Communication of the ACM,* 37 (3): 31-40.Available: http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p1.html (Accessed: August 6, 2002)

Murch, R. and Johnson, T. 1999. "What are agents? ", *"Intelligent Software Agents"*. Prentice Hall, New Jersey, pp. 5-17.

Nwana, H. S., and Ndumu, D. T. 1997. "An Introduction to Agent Technology", *" Software Agents and Soft Computing: Towards Enhancing Machine Intelligence"*. (Eds.) Nwana and Azarmi. Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, pp. 3-26.

Prabhu, R. "Mobile Agent Security Issues". Department Of Computer Science, Oregon State University. Available: http://islab.oregonstate.edu/koc/ece478/proj/2000RP/P.pdf (Accessed: October 5, 2002)

Riordan, J., and Schneier, B. 1998. "Environmental Key generation Towards Clueless Agents", *" Mobile Agents and Security"*. (Ed.) Giovanni Vigna. Lecture Notes in Computer Science. Springer-Verlag, Berlin, pp. 15-24.

Sander, T., and Tschudin, C. F. 1998a. "Protecting Mobile Agents Against Malicious Hosts", *" Mobile Agents and Security"*. (Ed.) Giovanni Vigna. Lecture Notes in Computer Science. Springer-Verlag, Berlin, pp. 44-60.

Sander, T., and Tschudin, C. F. 1998b. "Towards Mobile Cryptography". Proceedings of Security&Privacy'98, 3-6 May 1998 in Oakland, California. Available: http://www.cse.cuhk.edu.hk/~cmyu/research/Towards%20Mobile%20Cryptography.pdf (Accessed: September 27, 2002)