CERIAS Tech Report 2001-36

Design Issues for Information Assurance With Agents: Coordination Protocols and Role Combination in Agents

Thomas Bellocci, Chwee Beng Ang,
Parbati Ray, Shimon Y. Nof
Center for Education and Research in
Information Assurance and Security
&
School of Industrial Engineering, No. 01-07
Purdue University West Lafayette, IN 47907

Design Issues for Information Assurance with Agents: Coordination Protocols and Role Combination in Agents[1]

ChweeBeng Ang and Shimon Y. Nof

ABSTRACT

This report is a consolidation of work that has been done to fulfill the goals defined in the CERIAS Research Proposal, "Active Protocols and Agents for Information Assurance in Networked Enterprises ". According to the proposal, the development of an agent system for information assurance will follow two stages:
Design of active, combined task and assurance protocols
Development of active, secure task autonomous agents

Part I of the report deals with the issues involved in the design of a protocol for an agent system. An agent protocol is viewed as a coordination structure between agents whose design will affect the effectiveness and efficiency of the assurance system. Subsequently, three techniques of coordination are studied: organizational structures, meta-level information exchange and multi-agent planning. It was noted in the study that the different coordination structures affect factors such as communication costs, adaptation of problem solving ability and the length of the planning horizon in the agent system. Three important criteria in the design of an agent protocol were also identified: communication overhead, flexibility and scalability of the system.

Part II of the report introduces a new model of information assurance that is based on the integration of assurance functions within agents. It was hoped that such an approach would introduce greater confidence in the level of assurance of the information as it would be checked before the actual processing begins. It was also hoped that such an approach would also allows graceful degradation of the assurance functions when security needs at a certain time are determined to be non-critical.

Issues that affect the successful implementation of the model are identified as processing time, effects on the flexibility/mobility of the system, robustness of the system against subversion, and scalability of the system. The effects of the proposed model on the four issues are described and possible solutions are suggested to overcome the shortcomings of the model.

[1] This work was supported by sponsors of the Center for Education and Research in Information Assurance, Purdue University

TABLE OF CONTENTS

1. The Agent Protocol as Defined in Terms of Coordination

1.1. Introduction

In this part, the topic of coordination structures in agent systems is introduced. This topic is important to our research on information assurance as it provides an overview of the various structures that could be implemented in our proposed system and examines the merits and demerits of each structure. In addition, it also helps us to further define the various criteria that are relevant to the design of our protocol and to encourage us to actively incorporate these features into our design.

1.2. Definition of Coordination

In an agent system, coordination is defined by Jennings [12] as the process in which an agent reasons about its local actions and the (anticipated) actions of others to try and ensure the community acts in a coherent way. Recent research on coordination protocols and production agents in the PRISM Lab includes Huang (1999), Rajan and Nof (1999), Huang and Nof (2000), and Huang et al. (2000).

Coordination is important to ensure that all the necessary portions of the overall problem are included in the activities of at least one agent, that agents interact in a manner which permits their activities to be developed and integrated into an overall solution. Coordination also ensures that team members act in a purposeful and consistent manner and that all these objectives are achievable within the available and computational and resource limitations.

The main reasons for the need for coordination between multiple agents can be summarized as below [12]:
Because of dependencies between agent' actions
Because of the need to meet global constraints
Because no individual agent has sufficient competence, resources or information to solve the entire problem
Coordination structures in agent systems can be mainly classified as below 0:

1.3. Techniques for Coordination in Agent Systems

a) Organizational Structures

One method of coordination among agents is through the organizational structure of the agent network. In the context of Distributed Artificial Intelligence systems, an organizational structure can be viewed as a pattern of information and control relationships between individuals. These control structures are responsible for designating the relative authority of the agents and for shaping the types of social interaction that can occur. Hence, they can provide overall coordination of the agents by specifying which actions an agent will undertake and how redundancies of tasks undertaken by different agents could be avoided. The relationships specified by the organizational structures provide general, long-term information about the agents and the communities as a whole.

b) Meta-Level Information Exchange

Meta-level information exchange involves agents sending each other control level information about their current priorities and focus. For example, in the Distributed Vehicle Monitoring Testbed by Corkill 0, a network of problem solving nodes attempt to identify, locate and track patterns of vehicles moving through a two-dimensional space using signals detected by acoustic signals. A node constantly transmits its goals and hypotheses to other nodes, which it deems to be interested in the information it has to provide. It also receives the goals and hypotheses of other nodes. The local problem solving behavior of a node is influenced by the information it receives from other nodes. Hence, the problem solving ability of a node is balanced by its own perceptions of appropriate solving ability with activities deemed important by other nodes.

c) Multi-Agent Planning

In the multi-agent approach to coordination, agents usually form a plan that specifies all their future actions and interactions with respect to achieving a particular objective. It details, before actual execution, the areas of search space that will be traversed and the route each agent should take at each decision point in the activity. Multi-agent plans are typically built to avoid inconsistent or conflicting actions, particularly with respect to the consumption of scarce resources.

For example, in the air traffic control problem taken by Cammarata [2] each aircraft (agent) sends the coordinator information about its intended actions. The coordinator then builds a plan that specifies all the actions that the other aircraft or itself should take to avoid collisions.

1.4. Comparison of Advantages and Disadvantages of Coordination Techniques

The advantages and disadvantages that are associated with the implementation of the various coordination techniques can be summarized in Table1.

Table 1: Comparison of Different Protocols

Protocol	Advantages	Disadvantages
Organizational Structures	Lower communication costs Provides a control framework that reduces the amount of control uncertainty present in an agent as a result of incomplete or erroneous local control information Increases the possibility of coherence in the behavior of the agents by providing a general and global strategy for network problem solving	Lower flexibility in response to a changing task and hardware environment Fixed problem solving ability of agent
Meta-Level Information Exchange	Ability to strike a balance between the costs of communication and computation in optimally determining a solution and the disadvantages associated with solving the problem locally Ability of the agent to adapt its problem solving ability according to information and hypotheses transmitted by other agents	Susceptible to coordination errors due to receipt of incorrect information from other agents
Multi-Agent Planning	High reconfigurability and adaptability in rapidly changing task environments Easy extensibility of network to incorporate other agents Enhanced reliability and fault tolerance	More communicational and computational resources required than other two methods Short time horizon Higher possibility of generating coordination solutions that are locally optimal

1.5. Relevance to Present Research

According to the CERIAS Research Proposal [16], the development of the agent system will be done in two stages:

- Design of active, combined task and assurance protocols
- Development of active, secure task autonomous agents

Task 1 can be viewed as essentially designing the coordination structure in which the agents will operate. The coordination structure of the agent system will affect the effectiveness and efficiency of the assurance system. Specifically, we have to consider the following issues:

a) Communication Overhead

Since the assurance system will be added to the normal functions of a computer system, it is essential that the communication overhead from the agent system be as low as possible so that the performance of the system will not be affected by the running of the assurance system. Hence, it would be expected that a coordination structure in the form of an organizational structure would have a lower communication overhead than a multi-agent coordination system. In addition, incorporation of measures to reduce the amount of data transmitted could also reduce communication overhead. For example, in the agent-based intrusion detection system of [1], data reduction is carried out so that the amount of data that is transferred from the agent to the transceiver is reduced.

b) Flexibility

Flexibility is interpreted as the ability of the system to adapt its problem solving ability in response to different situations. While more flexibility in a system is usually better, under certain circumstances, flexibility may not be an important criterion.

For example, in the agent-based intrusion detection system of [1], the agents are basically programs that monitor for interesting events that happen in the host. They then report their findings to a transceiver that is at a level higher than the agents in the hierarchy. The transceivers reduce the data they receive from the agents and either distributes the data to other agents or to a higher level in the hierarchy for further process. Through such an arrangement, the individual agents do not have local autonomy and the transceivers, only limited autonomy. However, in this case, through the cascading of tasks such as data reduction, the relevant data is consolidated in monitors at the highest level of the hierarchy where pattern matching is carried out. The pattern matching procedure may be a set procedure. However, by bringing the relevant data to a higher level, the flexibility requirements of the system is reduced as compared to a system where the data is analyzed at the lowest level.

In conclusion, it should be emphasized that this observation may not apply to all the functions within the arena of information assurance and further investigations are necessary when choosing the agent protocol.

c) Scalability

Scalability in the system is the ability of the system to accommodate new nodes as they are added to the system. Addition of new nodes to the networked enterprise is always possible due to increases in personnel or to the increase in the operations of the business. Hence, the coordination structure must easily accommodate new additions without requiring change to the whole coordination structure. Whenever possible, modifications should only be done at the interface between the present coordination structure and the new addition with other modifications in settings done centrally at a specific module/location.

1.6. Future Tasks

Understanding the merits and demerits of different coordination structures in agent systems allows us to design the active task and administration protocols in our assurance system more effectively. It also highlighted 3 issues we have to consider in designing the protocols, namely communication overhead, flexibility and scalability. We can further define our tasks as below:

1. To determine the areas of information assurance we wish to concentrate on and to define the protocol by focusing on the above three criteria
2. To further understand how the concept of an active protocol may further improve our model
3. To determine whether the protocol should be implemented as a separate layer or whether the assurance functions could be incorporated into the present task administration protocol developed in PRISM [7,8,9, 16, 17].

2. A Role Combination Model for Information Assurance in Agents

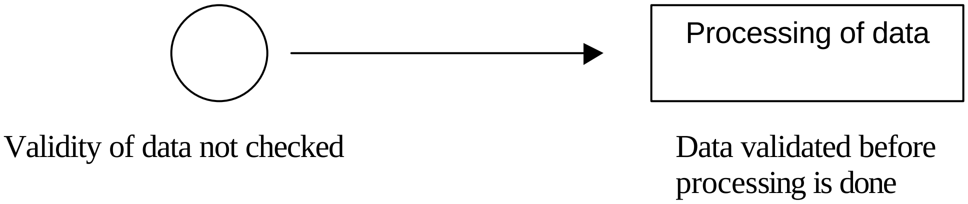
2.1. Model of Information Processing

In the CERIAS project proposal, we propose the possible integration of assurance functions into agents following the TQM approach [16]. This was done in comparison with traditional approaches to implementing assurance agents in a separate layer such as with dedicated agents for intrusion detection, authentication and so on.

The hypothesis is that the integration of assurance functions in agents would make them more autonomous in their actions. This is due to their ability to combine their data handling functions (within the context of an information system) with security functions.

In the literature on the applications of security agent architectures in [1], [7], [14], information security is implemented through constant monitoring of predetermined areas such as for signatures of intrusions [1] or vulnerabilities in the system [7]. However in our envisaged model of data processing as shown in Figure 1, the agent checks the assurance of the data (i.e. whether the data obtained is accurate and has been secured etc) before combining the data together. This model is similar to the one in [19], where authentication functions are incorporated into the agent, together with code to perform the specific task. Checks are done on the agents to ensure the required security levels are satisfied before processing is done.

Figure 1: Possible model of agent system with integrated assurance functions


Processing of data
Validity of data not checked
Data validated before processing is done

The rationale for this model is that compared to a model where assurance functions are implemented periodically such as a virus scanner that is set to execute once in a month, it offers greater confidence in the data as the quality is assured because they are carried out just before actual information processing takes place. This is especially important in mission-critical task processing where the quality of the data to be processed has to be guaranteed. Implementation of this model also allows graceful degradation of the assurance functions when security needs at a certain time are determined to be non-critical.

2.2. Scenarios where Role Combination is Justified/Unjustified

The combination of assurance functions with the normal task processing functions of the agent is justified when:

1. Assurance functions involve security and integrity checks that are to be done on the data that is to be processed (that is for data that is localized). This includes virus scan, checks for the completeness of the data and so on.
2. Data used is of very critical nature, sensitive or is constantly varying, e.g. stock prices. In this case, the validity of the data that has to be checked before the information processing begins
3. Data was received from unreliable sources
4. Autonomy and flexibility in the implementation of a security policy is important. Since the agent has control over the range of methods to carry out assurance functions, it is able to flexibly implement the security policy such as deciding which functions to implement based on say, the nature of the processing that is to be done. If it was assessed that the security requirements for a certain task 1 is not as high as task 2, then task 1 may have a smaller range of assurance functions that needs to be carried out

It is unjustified when:

1. The assurance functions are not limited to the data that is to be processed. It may for example involve the analysis of audit trails for the number of failed logins, user profiles etc over a long period of time
2. The assurance functions to be done are time consuming or may involve too high a processing overhead when carrying out the checks. In this case, it may be better to conduct the checks before the actual processing of the data
3. When there are many assurance functions to be carried out. In such a case, it is better to have a distributed form of checking rather than a serialized form of information assurance done by a single agent

Hence, there is a need to assign certain assurance functions to agents that carry out specialized tasks. These agents may be used in the collection of data in intrusion detection schemes such as AAFID [1]. Other assurance functions may be incorporated into the agents themselves and done just before information processing begins. Whether the agents carry out the assurance task or not depends on the risks associated with carrying out the task at that time.

2.3. Considerations in Agents with Combined Functions

The merits and demerits of an assurance system incorporating agents with combined functions could be further investigated in the below areas:

1. Processing time
2. Flexibility/Mobility
3. Robustness against Subversion
4. Scalability of the System

2.4. Processing Time

The time that is required for the real-time assurance of data used in processing is a very important factor in the suggested model. Users do not want their system to slow down significantly because of added assurance factors due to both the decrease in efficiency and ergonomic factors (Irritation, decrease in real-time effect in telecommunications etc). Hence, there is a need to be able to incorporate adequate responsiveness to the system when running the suggested model.

The following three steps could be carried out:

1. Assess the confidence level to be accorded to the assurance of the data
2. Based on the confidence level determined, decide on the level of assurance required
3. Activate the required assurance functions

To assess the confidence level, risk assessment as described below could be utilized.

2.5. Risk Assessment

Risk analysis could be done on the data before the processing takes place to determine the level of assurance that is required and hence the assurance functions that have to be implemented. From [13], [15] and [18], the following useful terms could be defined:

Risk Analysis – The process of identifying security risks, determining their magnitude and identifying areas needing safeguards

Threat – Any circumstance or event with the potential to cause harm to a system in the form of destruction, disclosure, modification of data and/or denial of service

Vulnerability – A weakness in the physical layout, organization, procedures, personnel, management, administration, hardware or software that could be exploited to cause harm to an ADP system or to the enterprise in which it resides

From risk analysis, threats and vulnerabilities are identified together with the risks that they pose to the system. In [15], threats and vulnerabilities serve as the inputs for the system. The risk assessment is based on the IP header component of incoming datagrams, the sub-components that have an impact on security issues are identified, of which one of them is the IP source address. Typical characteristics of the IP source address are:

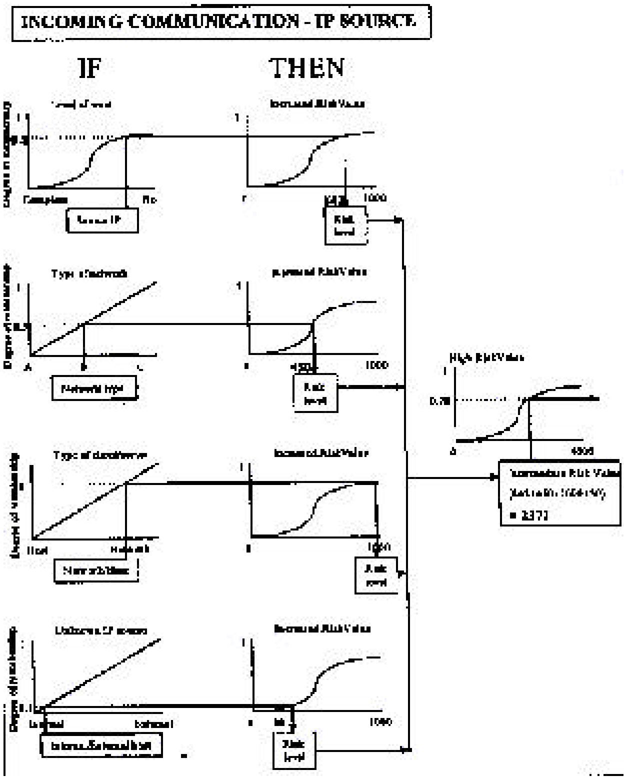
- Level of trust (trusted/untrusted)
- Size of Network (small/medium/large)
- Type of source (network/host)
- Location (internal/external)

The vagueness of the characteristics is modeled in fuzzy logic with the use of fuzzy sets. Each characteristic is then analyzed to determine how it could impact the subcomponent. An example of the fuzzy rules might be as below:

If Source.IP is trusted, then RiskValue is decreased
If Source.IP is trusted, then RiskValue is increased
If Source.IP is Type_A, then RiskValue is increased
If Source.IP is Type_B, then RiskValue is constant
If Source.IP is Type_A, then RiskValue is decreased

Inputs are mapped from the membership value to a common scale as shown in Fig. 2. The intermediate risk values for each characteristic calculated above is then consolidated mathematically to obtain a risk value for the specific sub-component. Risk values of each of the sub-components are then consolidated mathematically into a Global Risk Value (GRV) for the particular scenario.

Figure 2: Risk Value Calculation



Using similar approaches to that in [15], it is expected that the appropriate assurance level can be determined from the GRV and the appropriate responses activated in the agent. In the example of [15], the risk values are divided into low, medium and high and they respectively activate the appropriate baseline, control and analytical countermeasures.

2.6. Flexibility/Mobility

Flexibility/Mobility in the present context refers to the ease in which the agent can migrate from host to host to carry out its tasks. Such a design paradigm is referred to as mobile agent architecture.

In [6], the term “mobile agent” is used by the distributed system community to define a software component that is able to move among different execution environments. This is usually used in the artificial intelligence community in corporation with the view of an intelligent agent that is able to achieve a goal by performing actions and reacting to events in a dynamic environment [6].

Some of the advantages of achieving mobility through a mobile agent paradigm can be listed as below:

a) Overcoming Network Latency

Although a central controller can send messages to the nodes within the network and issue instructions on how to respond to certain scenarios, such an approach may become problematic when it has to respond to a certain number of events in addition to its normal processing load or when communication links are unreliable, leading to unacceptable delays. For a mobile agent paradigm, the response to a situation could be autonomously determined and executed by the agent, hence reducing such delays.

b) Reducing Amount of Data Transferred

Since mobile agents possess the methods required to process a task, they can filter the data from the host in which it is resident and perform the computations on the host instead of the home platform. Hence, there will be no need for the transfer of large amounts of data across a network for local processing. The situation is especially advantageous when the agent to be transferred is smaller in size than the data to be transferred.

c) Asynchronous Execution and Autonomy

A property of a mobile agent network is that it can continue to function even in the event of a failure of the central controller or communication links. This is due to the ability of the agent to operate autonomously after it is launched from a home platform. Consequently, the agent can continue to fulfill its task processing functions in the event of an attack on the central controller on when communication links fail.

d) Adapting Dynamically

Mobile agents provide a versatile and adaptive computing paradigm as they can be retracted, dispatched, cloned or put to sleep as network and host conditions change. In addition, they can also sense their execution environment and autonomously react to changes. Mobile agents can for example, sense the computational load on a host and if it is too high, move to another host with a lower utility. They can also distribute among the hosts in the network in such a way as to maintain the optimal configuration for solving a problem.

e) Robust and Fault Tolerant Behavior

The ability of mobile agents to react dynamically to unfavorable situations makes it easier to build robust distributed systems. In addition, the support provided for disconnected operations and distributed design paradigms eliminates single point of failure problems and allows them to provide fault-tolerant characteristics.

However, with the incorporation of assurance functions into the agents, the code size of the agents is expected to increase due to extra code for assurance functions. Hence, transmission of the agent to another host would require a longer time. In addition, the transfer would also result in greater computing and network resources. Mobility of the agents will hence be restricted.

A possible solution is to have all nodes in the network install an agent platform that will host the assurance functions that are required by the mobile agents as they migrate from one host to another. In such a case, the migrating agents will only carry the minimum code that decides the type of assurance that will be carried out at the host based on a risk assessment of the data and the security scene at that moment. The relevant functions are then invoked from the agent platform at the target host. The disadvantages of such an approach are:

1. If the agent platform centralizes all or a large part of the assurance functions that are required by the agent, it defeats the purpose of the distributing assurance in agents to reduce an attack on the assurance component of the information system (as discussed in Robustness against Subversion)
2. If different agent platforms are implemented on different networks, there is a difficulty to ensure that the assurance functions that are provided by a different network can fulfill the degree of assurance that is required by the initiator
3. If the same agent platform is to be implemented in all networks, there would be difficulties in getting the security community to agree on common standards

2.7. Robustness against Subversion

With the proposed model, it is expected that the robustness of the system against malicious attacks will be increased. This is because in the proposed model, there is no clear target that could be subverted in the event of an attack on the system. To illustrate the point, a comparison is made with the AAFID intrusion detection system described in [1]. In AAFID, the monitor is a single point of failure as it is the entity that receives the data from the tranceivers and agents beneath it and does high-level analysis of the data. Furthermore, existence of reliable communication paths between the monitor and the entities it controls is also important for the proper functioning of the intrusion detection system. Hence an attack on the monitor(s) or the communication links would effectively decapitate the entire intrusion detection system.

However, in the proposed model, there is no centralized target for attack as the assurance functions are distributed among all agents. In addition, for the case of mobile agents that are residing on another host, it may also be programmed to take an alternative route back to the home platform when a communication link is made unavailable. Hence subversion of the system becomes more difficult.

Furthermore, the design of the agents employed in task processing may also vary from network to network. There may be differences in the way the risk of accepting the data is assessed, the types of assurance functions deployed and the working of particular assurance functions. Hence, the variability in the agents also serves as a deterrent to attackers and hampers the creation of an action plan to subvert the system.

2.8. The Scalability of the System

Scalability of the system refers to the ease of adding new nodes to the system. A centralized approach to security such as a security server that handles all requests for virus scans on data suffers from an inability to meet rising demands for service as the number of nodes in the network increases and or when periodic increases in workload is experienced. Hence, there is a limit to the number of nodes that could be added to the system because each new addition would increase the computational load on the server. An incorporation of assurance functions into the agents would distribute the computational load.

In addition, if the agents are mobile as well, execution of the information assurance could be carried out in the target host, hence reducing the load on the home platform. Alternatively, the agent could be designed to detect the utility of the target host at the moment of its residence. If it is determined that the utility of the host is too high, then the agent might migrate to another trusted host with the data that that it has obtained. The assurance functions are then executed on the new host using its computational resources and the results transferred back to the home platform. (Fig. 3)

2.9. Future Tasks

From the discussions above, it can be seen that the proposed model has certain shortcomings such as the limitations in the mobility of the agents due to its increased code size. However, the model also promises a higher level of information assurance through checking just before actual information processing begins.

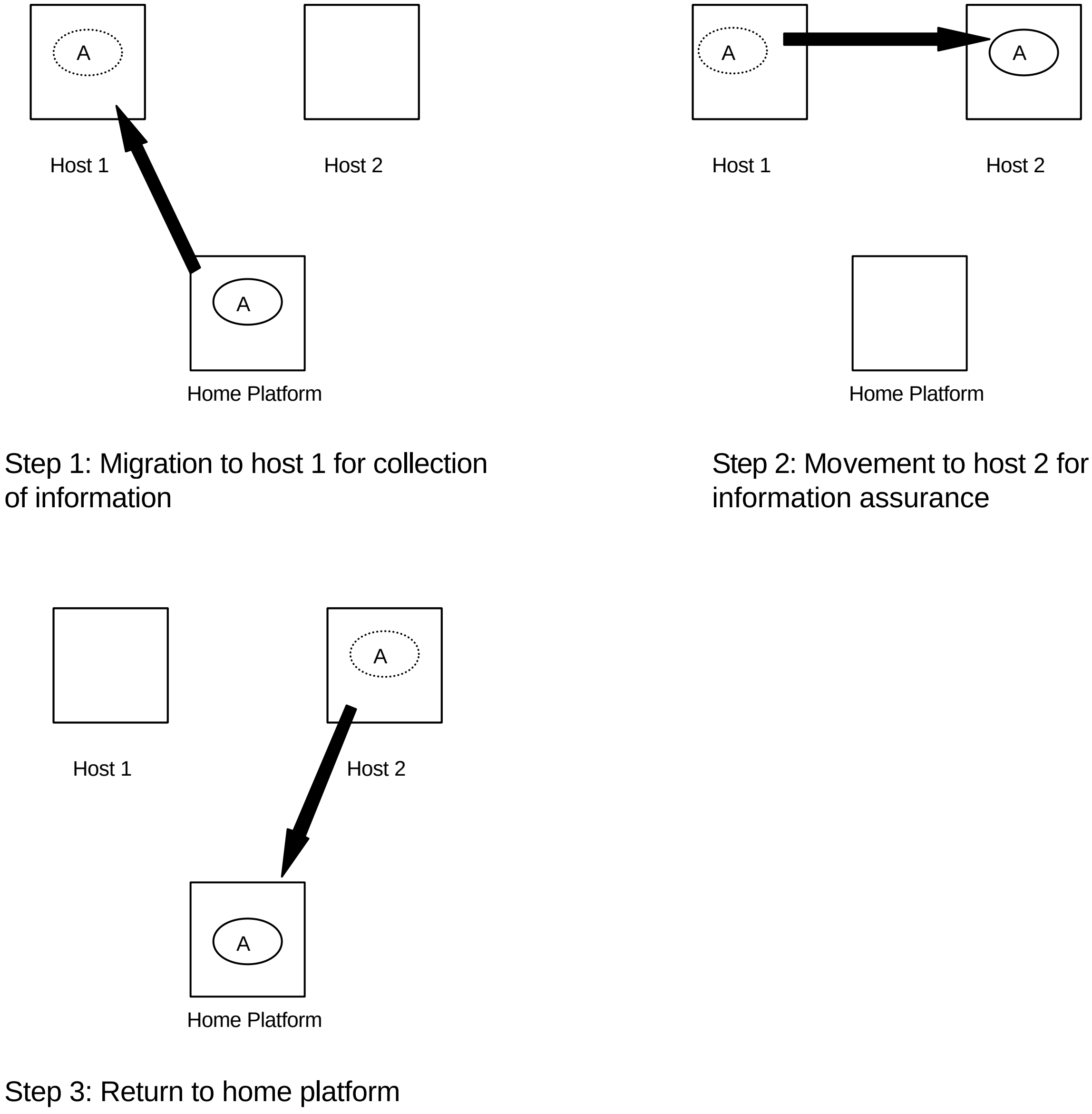
Future tasks include:

a) An investigation of the assurance functions to determine which functions are suitable for incorporation into the agents and which functions are better carried out by specialized units in the system
b) Actual design of the agent system using the proposed model. The design would include the determinatio n of the protocol that coordinates the actions of the agents in the system,

specification of communication methods between agents and the various assurance functions that are to be incorporated. In addition, if mobile agents are to be used, then appropriate measures of security has to be devised for the agents as they migrate from host to host.

c) Implementation of the agent system on the PRISM lab's parallel computer, Team Integration Evaluator. Experience accumulated during previous work done in the lab on autonomous agents for manufacturing [7, 8, 9] would be utilized at this stage.

Figure 3: Movement of agent from home platform to host 1 and 2 for data collection and information assurance



Step 1: Migration to host 1 for collection of information

Step 2: Movement to host 2 for information assurance

Step 3: Return to home platform

REFERENCES

[1] Balasubramaniya J, Garcia-Fernandez, Isacoff, Zamboni D, Spafford, *An Architecture for Intrusion Detection using Autonomous Agents*, Department of Computer Sciences,Purdue University; Coast TR 98-05; 1998, http://www.cerias.purdue.edu/coast/coast-library.html

[2] Cammarata S. MacArthur D. Steeb R., *Strategies of Cooperation in Distributed Problem Solving*, Proc. Int. Joint Conf. On AI, Karlsruhe, Germany, pp 767-770.

[3] Cockburn D., Jennings N.R., *ARCHON: A Distributed Artificial Intelligence System for Industrial Applications*, Foundations of Distributed Artificial Intelligence (eds. G. M. P. O'Hare and N. R.Jennings) Wiley, 1996, 319-344.

[4] Corkill D.D., Lesser V.R., *The Use of Meta-level Control for Coordination in a Distributed Problem Solving Network*, Proc. Int. Joint Conf. on AI, Karlsruhe, Germany, pp 748-756.

[5] Crosbie Mark, Spafford Gene, *Active Defense of a Computer System using Autonomous Agents*, Department of Computer Sciences, Purdue University; COAST; TR 95-008 http://www.cerias.purdue.edu/coast/coast-library.html

[6] A.Fuggetta, G.P. Picco, and G. Vigna, *Understanding Code Mobility*, IEEE Transactions on Software Engineering, 24(5), May 1998, pp. 342-361.

[7] Chin-yin Huang, *Autonomy and Viability in Agent-based Manufacturing Systems*, Phd Thesis, August 1999, School of Industrial Engineering, Purdue University

[8] Huang, C.Y., and Nof, S.Y., “Formation of Autonomous Agent Networks for Manufacturing Systems,” *International Journal of Production Research*, Vol.38, No.3, 2000, pp. 607-24.

[9] Huang, C.Y., Ceroni, J.A., and Nof, S.Y., “Agility of Networked Enterprises - Parallelism, Error Recovery and Conflict Resolution,” *Computers in Industry*, Vol.42, No.2-3, 2000, pp. 275-87.

[10] Humpries W. Jeffrey, Craver Jr. Curtis A., Pooch Udo W., *Secure Agents for Network Vulnerability Scanning*, Proceedings of the 2000 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 6-7 June 2000 http://www.itoc.usma.edu/marin/Papers2000/TA1_4.pdf

[11] Jennings N.R., *A Roadmap of Agent Research and Development*, Int. Journal of Autonomous Agents and Multi-Agent Systems 1 (1) 7-38, 1998

[12] Jennings N.R., *Coordination Techniques for Distributed Artificial Intelligence*, Foundations of Distributed Artificial Intelligence (eds. G. M. P. O'Hare and N. R. Jennings), Wiley, 1996, pp187-210.

[13] Jelen F. G,Williams J.R, *A Practical Approach to Measuring Assurance*, 4th Annual Comp. Security Applications Conference, Dec 7-11, 1998, Pheonix, Arizona http://www.acsac.org/1998/abstracts/fri-b-1030-jelen.html

[14] Jose Duarte Queiroz, Luiz Fernando Rust da Costa Carmo, Luci Pirmez, *Micael: An Autonomous Mobile Agent System to Protect New Generation Networked Applications*, Recent Advances in Intrusion Detection (RAID) Workshop 1999, http://www.raid-symposium.org/raid99/

[15] L. Labuschagne and J.H.P Eloff, *The Use of Real-Time Risk Analysis to Enable Dynamic Activation of Countermeasures*, Computers and Security Vol. 17, No 4, pp. 347-357, 1998

[16] Nof S.Y., *Active Protocols and Agents for Information Assurance in Networked Enterprises – CERIAS Research Proposal*

[17] Rajan, V.N. and Nof, S.Y., Computation, AI, and Multiagent Techniques for Planning Robotic Operations, Chapter 31 in *Handbook of Industrial Robotics*, 2nd Edition, S.Y. Nof, Ed., John Wiley & Sons, 1999

[18] W.G. de Ru and J.H.P. Eloff, *Risk Analysis Modeling with the use of Fuzzy Logic*, Computers and Security Vol. 15, No 3, pp. 239-248, 1996

Vijay Varadharajan, Nikhir Kumar and Yi Mu, *Security Agent Based Distributed Authorization: An Approach*, 21st NISSC Proceedings, October 6-9, 1998, Crystal City, Virginia http://csrc.nist.gov/nissc/1998/papers.html

[19] Wayne Jensen, Peter Mell, Tom Karygiannis, Don Marks, *Applying Mobile Agents to Intrusion Detection and Response*, National Institute of Standards and Technology Computer Security Interim Report (IR) – 6416, Oct 1999 http://www.itl.nist.gov/div/893/staff/mell/response.pdf

Appendix - A Summary of the Application of Agents in Computer Security

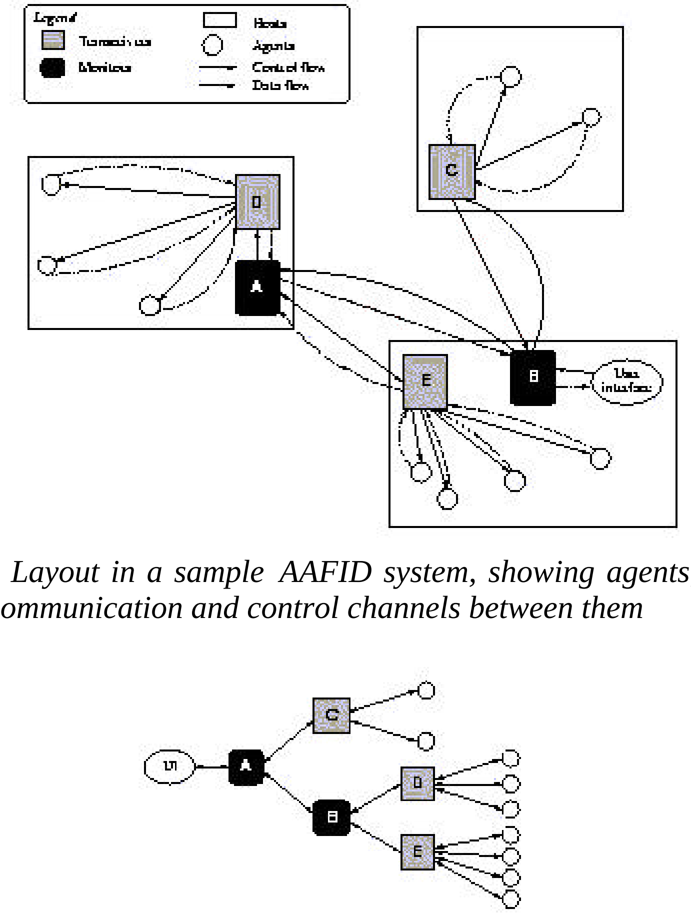
1. Introduction

The application of agent technology in computer security is becoming more pronounced in the last few years. The objective of this summary is to give the reader an idea of how agent technology was applied in various aspects of computer security. The role of agents in security range from the detection of intrusions in a computer system [1], [14] to an authorization architecture in [19].

2. AAFID – A Agent-based Intrusion Detection System [1]

AAFID or Autonomous Agents for Intrusion Detection [1] is an architecture for building intrusion detection systems (IDS) that uses agents as their lowest-level element for data collection and analysis. It employs a hierarchical structure of agents, tranceivers and monitors that may be distributed over any number of hosts in a network. The network is shown in Fig 4.

Fig 4: AAFID system architecture



(a) Physical Layout in a sample AAFID system, showing agents, tranceivers and monitors, as well as the communication and control channels between them

(b) Logic organization of the AAFID system

Agents monitor for interesting events occurring in a host and report their observations to a single tranceiver. The tranceiver overseas the operation of all agents under their charge in a host and have the ability to start, stop or send configuration commands to the agents. They do data reduction on the data received from the agents and report their results to one or more monitors. Monitors uses data from the entire network to perform higher level correlation and detect intrusions that involve several hosts.

One of the advantages of AAFID is that it introduces modularity into the system. This is because agents are independent-running entities, which may be added to or removed from the system without affecting other components. It also limits the effects of a defective agent as the damage would be only limited to only one agent or a group of agents if it stops working. The use of a hierarchical structure in the IDS also enables data to be reduced and reported to upper layers, hence enhancing the scalability to the system.

3. Secure Agents for Network Vulnerability Scanning [7]

A system for improving vulnerability assessment process was proposed through the use of mobile agents. A mobile agent is a program that represents a user in a computer network and can migrate autonomously from node to node to perform some task on behalf on the user. It combines the advantages of both host-based and network-based scanning tools with the benefits of fast customization for detecting newly discovered vulnerabilities.

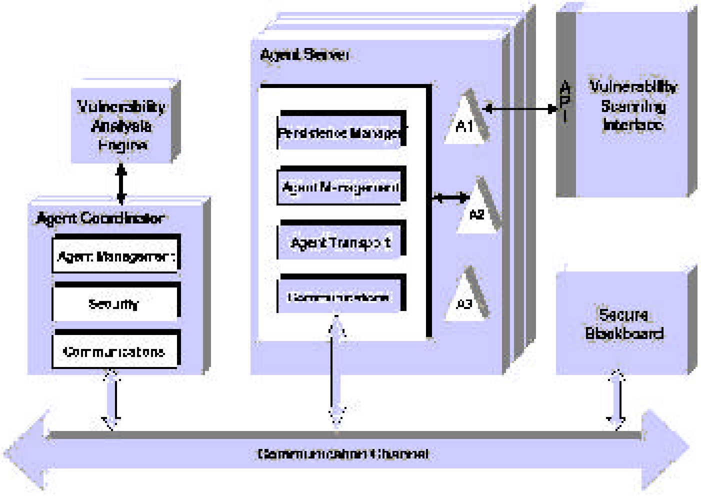


Figure 5: System overview of network

The architecture of the system is shown in Fig. 5 and consists of agents, agent servers and agent coordinators. The agents are mobile detectors that migrate from host to host to detect vulnerabilities. Mobile agent servers provide the actual runtime environment on a host for visiting mobile agents. The agent coordinator is a centralized controller responsible for the creation and removal of mobile agents from the system. It also works in conjunction with individual agent servers to detect breaches in security in its agents.

Vulnerability checks are done through the roaming agents, which follow a pre-determined itinerary that is established by the agent coordinator. The agent also carries a code that does the check and that is executed at the hosts it visits. Results obtained from the execution of the code on a host is saved as payload by the agent and passed to the agent coordinator for analysis on return. Feedback is then given to the system administrator through a graphical user interface.

In the proposed system, security measures must be taken to ensure the safety of both the mobile agent and the host in which the agents execute their code. The security issues include:

- Protection of the confidentiality of an agent's data and code
- Ensuring the integrity of the agent i.e. that the agent is not tampered with as it migrates from place to place
- Ensuring that agents are always available to do vulnerability checks
- Authentication of both agent server and agent to ensure that the right agent is visiting the right host

4. Micael [14]

The Micael system is an intrusion detection system built upon AAFID. However, unlike AAFID where the agents are static, in Micael, the agents are mobile. The architecture of Micael is shown in Fig. 6 and consists of a Headquarter, Sentinels, Detachments, Auditors and Special agents.

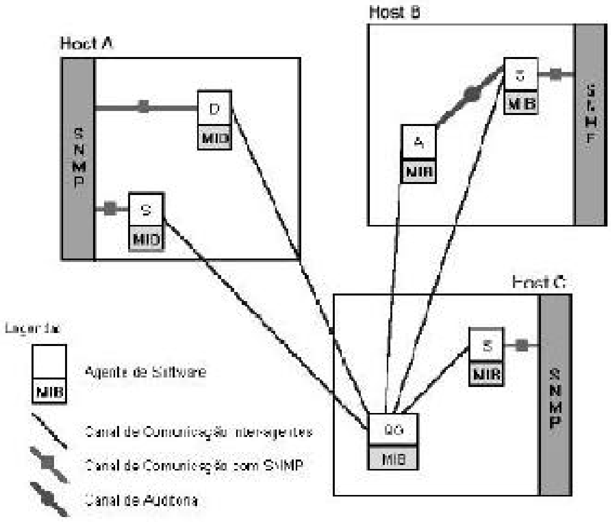


Fig 6: An example of Micael system for a network composed of three hosts. Each host runs a Sentinel Agent (S); Host A runs also a Detachment Agent (D); Host B runs an Auditor Agent; Host C also runs the Headquarter Agent (QG)

The Headquarter (QG) is a special agent that centralizes the system's control function. It is responsible for the creation of other agents and hence, maintaining a database of the agents' executable code. Sentinels are agents that remain resident in each of the target network hosts and are responsible for collecting relevant information for the QG. When the Sentinels detect any anomalies in the hosts, it requests the creation of a Detachment from the QG. Detachments are agents that are specialized to deal with a particular anomaly and can take defense and counter-attack measures against the hazard, if it is confirmed.

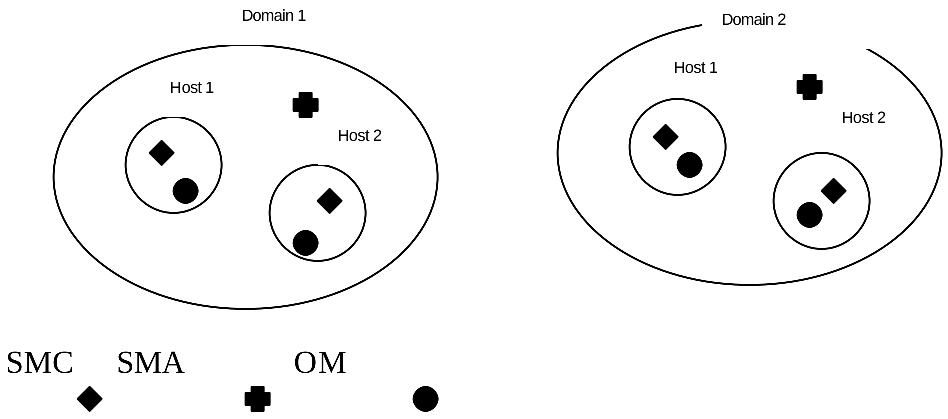
Auditor agents are agents that are created by the QG to check the integrity of the active agents. If it detects that a Sentinel is missing, it requests the QG to recreate the appropriate Sentinel. In addition, the Auditor agent also has the ability to recreate the QG if for some reason, the QG was aborted.

Micael utilizes the mobility of the agents to reduce the amount of resources required for ordinary operations but is able to concentrate the maximum amount of resources at the required place and time. For example, the ability to dispatch an Auditor agent to check the integrity of the agents make the allocation of an Auditory module on each Sentinel unnecessary and hence represents a savings in resources. In addition, in situations where a Sentinel detects an anomaly, extra resources such as mobile Detachments may also be sent to the place where it is required.

5. Security Agent Based Distributed Authorization [19]

The proposed system considers a security agent based approach to authorization in a distributed environment. A security agent (SA) is used to capture the privileges and part of the security policy on distributed authorization. Agent enabled hosts has a security management component (SMC) which is concerned with the security of the host and its execution environments. In addition, the collection of hosts that obey the same security policy are grouped together in a domain controlled by a Security Management Authority (SMA). The architecture is shown in Fig 7.

Fig 7: Architecture of agent based authorization model



Client principals that wish to access certain hosts can insert code within the SA to perform the required tasks. The SAs are created at the client principal's own SMC and has several elements containing information about the privileges of the principal, the validity of the privilege information as well as identity. It would also contain other information which it collects as it passes through other hosts.

The request from the client principal is passed with the SA to the target. An Object Management (OM) element at the target interacts with the SMC to verify the client principal, the SA and to determine whether the request is to be granted or not. Since the SA is a full-fledged object (program and data) and has the ability to gather information relevant to its requests as it moves from host to host, it can use the collected information to make dynamic decisions on the behalf of the client principals.

Several issues that are to be considered with regard to SAs are:

- That the SA should be unforgeable
- The SA should only have the capability to make those decision which it has been allowed to do and should not make any unauthorized decisions and requests without being detected
- Methods of checking the integrity of the SA should be available to the target
- Methods to protect the agent from threats from the target should be available

-

Table1: Comparison of Applications of Agents in Computer Security

Projects	AAFID [1]	Secure Agents for Vulnerability Scanning	Micael [14]	Security Agent Based Distributed Authorization [19]
Tasks				
Agents	Monitoring of events in system	Detection of vulnerabilities on host	Collection of relevant information (sentinels) Countermeasures against unauthorized use (detachments) Checking on integrity of agents (Auditors)	To carry means of authentication To execute code it carries To transfer collected data at target principal
Overall	Intrusion detection	Determination of system problems	Intrusion detection and countermeasures against intrusion and unauthorized use	Implementation of tasks on remote hosts
Coordination Scheme	Organizational Structure (Hierarchy)	Organizational Structure	Organizational Structure (Hierarchy)	Organizational Structure
Mobility of Agents	Immobile	Mobile	Mobile	Mobile
Creation of Agents	Pre-created	Created on demand	Created on demand	Created on demand
Measures of Effectiveness	Ability to detect anomalies	Whether agents or data are susceptible to compromise Ability to customize agents for new vulnerability scans	Ability to detect anomalies Effectiveness of reactions to intrusion and unauthorized use	Effectiveness of authorization system Contribution towards dynamic decision making
Costs	Monitors are single points of failure	Provision of adequate security for mobile agents Costs in performance due to implementation of security features in agent	Provision of adequate security for mobile agents	Provision of adequate security for mobile agents
Benefits	System Modularity Scalability of system	Fast customization of agents for detection of new vulnerabilities Scalability of system	Minimum use of resources due to specialization of agents Ability to dispatch the appropriate agents to handle hazards Easy reconfiguration of agents Scalability of system	Supports the ability of agent to make dynamic decisions Allows the delegation and revocation of duties and privileges