An Agent Based Dynamic Information Security Model in Information Warfare

M. H. Kuo

Graduate School of National Defense Information,
National Defense University
P. O. Box 90046-15, Chung-Ho, Taipei, Taiwan, R.O.C.
Email:mhk@rs590.ndmc.edu.tw

Abstract

In this paper we proposed an agent-based Dynamic Information Security Decision Model (DISDM) that automatically integrates all distributed information security systems to achieve efficient IW-D. The proposed DISDM can 24-hour automatically drive the intrusion inspection by dynamically executing sound security strategies (firewalls, high assurance guards, authentication, intrusion detection, encryption, and security management etc.). Also, it can recover the damages when systems have been under attack. Thus, it is expecting to yield IW-D cost-effective solutions.

Keywords Information Warfare, Information Security, Software Agent, Multi-Agents

1. Introduction

In today's electronically interconnected world, information moves at the speed of light, is intangible, and is of immense value. Today's information is the equivalent of yesterday's warfare weapon. A revolution in military affairs is a major change in the nature of warfare brought about by the innovate application of new technologies which, combined with dramatic change in military doctrine and operational and organizational concepts fundamentally alters the character and conduct of military operations - Information Warfare (IW).

Basically, the Information Warfare functional capabilities include offensive information warfare (IW-O) and defense information warfare (IW-D) (Figure 1). The IW-O weapons have follows: Computer Virus, Worms, Trojan Horses, Logic Bombs, Trap Doors, Denial-of-Service, Spooling, Nano Machines, Electric jamming, HERF-gun, and EMP-bomb etc. The IW-D methods have follows: Intrusion Detection System (IDS), Firewall and System Guard etc. [1]. However, these methods are with many deficiencies.

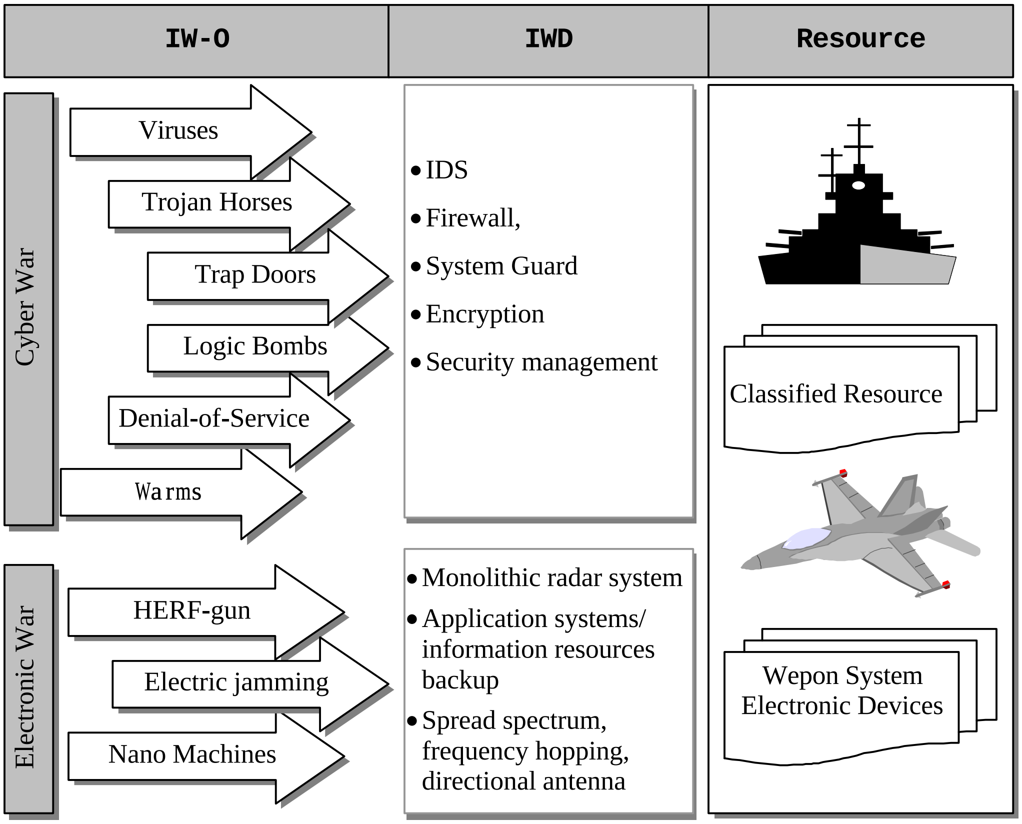

Figure 1. Information Warfare functional capabilities

First, IDS is an IT technique that can detect network intruders on its own initiative [2]. Many current intrusion detection systems are based on Denning's intrusion detection model [3], where audit records, network packets, or any other observable activity serves as the basis for detecting abnormalities in the system or checking them with traces and signals of known intrusion patterns. S. Kumar divided the methodology into two categories [4]:

(1) Misuse Intrusion Detection
Misuse intrusion detection uses well-defined intrusion patterns to check if a user were hacker or cracker. These intrusion patterns can be installed into the system in advance.

(2) Anomaly Intrusion Detection
Anomaly intrusion detection builds statistical profiles of the activities from network users' activity records. It regards activities that differ remarkably from normal use as intrusions, i.e. the user could be a hacker or cracker.

Each method has its own drawbacks. The misuse intrusion detection will not work against new or unknown forms of attack. The anomaly intrusion detection is ineffective in detecting insider attacks. Thus, any intrusion detection system that employs only one of these methods will have a limited range of intrusions it can detect. Intrusion detection system that intends to avoid these handicaps usually involves parallel employment of both techniques. Unfortunately, the system will become very large and operate slowly [5]. In addition, IDS is usually installed in an intrusion server. To protect the whole network (intranet), it needs momentarily to check, analyze and store the activities of network users alone. The so-called "Client-Server protection mechanism" will become a big burden of the network because it generates much extra communication information between IDS and application systems.

Second, firewalls or system guards are unlike IDS that takes the initiative in detecting intrusion, they passively use pr-defined filtering rules to blockade illegal network access. Usually firewalls do not check the contents of information packets. If a valid application with viruses or worms they will not be able to detect the attacks. Furthermore, firewalls or system guards are unable to stop hackers using denial-of-service techniques to attack network. Besides previous two drawbacks, firewalls or system guards are ineffective in detecting insider attacks [6].

From previous discussion, traditional IW-D methods are “static”, with many drawbacks. In this paper, we propose an agent-based Dynamic Information Security Decision Model (DISDM) that dynamically integrates distributed information security systems to achieve efficient IW-D. The core of DISDM is an intelligent agent and several distributed mobile agents work together with it to perform network security. Details of DISDM components and how it works will be discussed in following sections.

2. Software agents

The so-called “agents” are software entities that assist people and act on their behalf [7-13]. Usually a software agent system is composed of several small agents that may have different structures and remain highly dynamic when working a job together. It has been considered as a real good technique that can perform searching or retrieving from the completely dispersed data sets distributed all over the whole world. Recently software agents are considered as the major technique of solving network problems and have been widely applied. Plenty of investigations have been reported, such as: information filtering and gathering, electronic commerce, telecommunication systems, process control, entertainment and medical care [14,15]. Nevertheless, attention must be paid to a software agent, instead of a new style of programming, which is a way of handling tedious/difficult jobs not handled by a single program.

The benefits of using agent are as follows [16]:

- *Distributed Parallel Processing* - An agent uses to search among distributed information or to solve problems that a user cannot solve. Using an agent provides an optimum way and is a real distributed parallel process without limiting to a local region. A task is divided into several components. Each component can be executed separately. Communications are performed when they are needed. These components can be, for example, storing/retrieving data from a digital library, data mining, electronic business, and searching/filtering over WAN.
- *Reduction of the network load* - In a distributed client/server system, tasks can be executed only if some effective communication protocol is followed. It releases that implies that multiple interactions exists on the network, since all the temporary data during the execution must be transmitted between the client and the server. A software agent can reduce the load and solve unsteadiness on the network based on its abilities such as pro-activity and portability. It means that operations on the network can be more efficient and stable. For instance, when searching information coming from a remote server, a client may have to answer several questions one after the other depending on the results of executing the previous orders. Under the client/server structure, all the information that the client asks for will be sent to the client and the client is ready to make the next decision in order, say to filter the information. In other words, all the temporary information during the execution of a whole task must be transmitted to the client. If such a filtering of information can be processed at the server completely, the load on the

network and the searching time can be greatly reduced. The advantage is a balance between requesters and providers. Moreover, the discontinuous connection makes the transmission lines available for other users.

- *Scalability due to dynamic development* - In addition to reducing the network load, a software agent can also check doubly transmitted information from various resources through the network and check whether the information fits the users' requirement. Moreover, an agent can take charge of the whole process of retrieving information so as to save the searching time and in turn enhance the user's ability of making correct decisions.
- *Asynchronous operation* - This is also one of the motivations of using an agent. The user can turn off the computer equipment after assigning the job and checking the result whenever the user wants to. The agent works completely independently after receiving the task. This builds the structure or function of "not in need now but useful in the future."
- *Software distribution on demand* - The advantage can be illuminated by the use of applet of Java. Software can be downloaded through an agent without warring about the version of software or installing the software manually. On the other hand, the user can assign personal tasks to other agents or write new functions for special tasks. Software can be developed asynchronously, unlike the present client/server structure in which re-setup is required at both the client and server computers.
- *Agreement on a Language or Protocol* - The present solution to continuously emerging applications and requirements is to establish more and more protocols. However a software agent only requests a consistent transport protocol and no new protocol is needed for new applications and requirements. An open digital information system can thus be built on heterogeneous computers. The obvious flexibility also enables the development and extension of the system in the future.

Based on agent capability, agent technologies are divided into three types: Intelligent agent, Mobile agent and Collaborating agent [17]. Each has its own properties as follows:

(1) *Intelligence agent* - An agent can observe and learn users' habits through its past experiences. Therefore, an agent possesses the abilities of supervising, reasoning, and explaining.

(2) *Mobile agent* - An agent can be transported from a server to another template of a different system structure.

(3) *Collaborating agent* - An agent handles applications through cooperation that a single program cannot handle, such as job scheduling, mail managing, data mining, data searching, shopping on line, digital library, process managing, and so on.

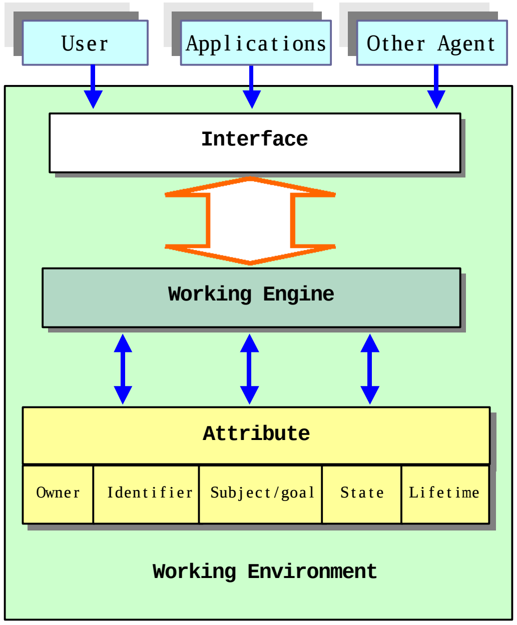

Figure 2: The inner components of a DISDM agent.

Technically, the agents in the proposed DISDM system includes the following several components (Figure 2)

- **Interface** - An agent should provide an interface that the user or other agents can communicate with. This interface is usually developed using a kind of software communication language, such as KQML (Knowledge Query and Manipulation Language). The tasks of the interface are client's authority check, security checking, subject/goal validation and agent language translation. Through it, user can commit the agent to achieve his work or other agents can seek the agent's help. Similarly, when the agent wants to seek outside support, it uses the interface to communicate with other agents.
- **Attributes** - While an agent executes a mission it must carry necessary information. The information is called "agent-attribute". Usually, number of agent-attribute depends on the agent requirement. Different agent has different number of attributes. Follows are five attributes of a procurement agent

  (1) **Owner** - Owner attribute is to record an agent owner's information, such as the owner's server name, IP address, password, email account, … and so on. If an agent is cloned from other agent, then the owner attribute will record its parent agent's ID number. This attribute is very useful for information backtracking and error debug. Moreover, the final result of agent's work is sent back to the owner according to this information.

  (2) **Identifier** - Each agent must have an ID number. This is to prove that it is a legal agent but not a virus. Then the host server is willing to provide its service. If the service needs to pay, the host records agent's ID for further charge.

  (3) **Subject/Goal** - When an agent (especially for a mobile agent) wants to seek service it needs a subject/goal to let the host server understanding its purpose. For example, an agent expects the central unit of DISDM to alert all agencies to increase awareness activities. Then, the statement "Alert all agencies to increase awareness activities" is the agent subject for this mission.

  (4) **State** - The state attribute records an agent's snapshot of its execution. When an agent travels, it transports its state with it. This is necessary for the agent to resume execution at the destination host. For most programming languages we can partition

the agent's state into its execution state that is its runtime state, and its object state that is the value of the instance variables in the object.

(5) **Lifetime** - Each agent must have a lifetime to determine the period of its existence. Different agent has different lifetime. Some static agent has log lifetime. When its resident host machine does not shut down, it always exists. However, some are short life, especially for most mobile agents. It needs to know; a dispatched mobile agent without lifetime could become an "agent orphan" roaming in network after it done its mission. This will cause network disastrous.

- **Working Engine** - A working engine is the core component of an agent. It serves as the workhorse for the agent while it processes its job. More precisely, working engine provides the agent with job scheduling, state extraction, subject execution, linkage to the underlying network and other resources provided by the host. We can judge whether an agent is " intelligent" or not by examining how smart its working engine is.
- **Environment** - Environment is sometimes called working place or working background. It is the place where agent processes its job (executes the owner's command). In practical system, it could be a server's operation system or a specific place (for example, a Java virtual machine) to host a client agent.

3. Dynamic Information Security Decision Model (DISDM)

In this section, we proposed an agent-based Dynamic Information Security Decision Model (DISDM) that automatically integrates all distributed information security systems to achieve efficient IW-D. Basically DISDM is divided into 5 kinds of agents. They are Inspection Agent, Evaluation Agent, Messaging Agent, Command Agent, and Defense Agent (see figure 3). The functions of each agent are briefly described as follows:

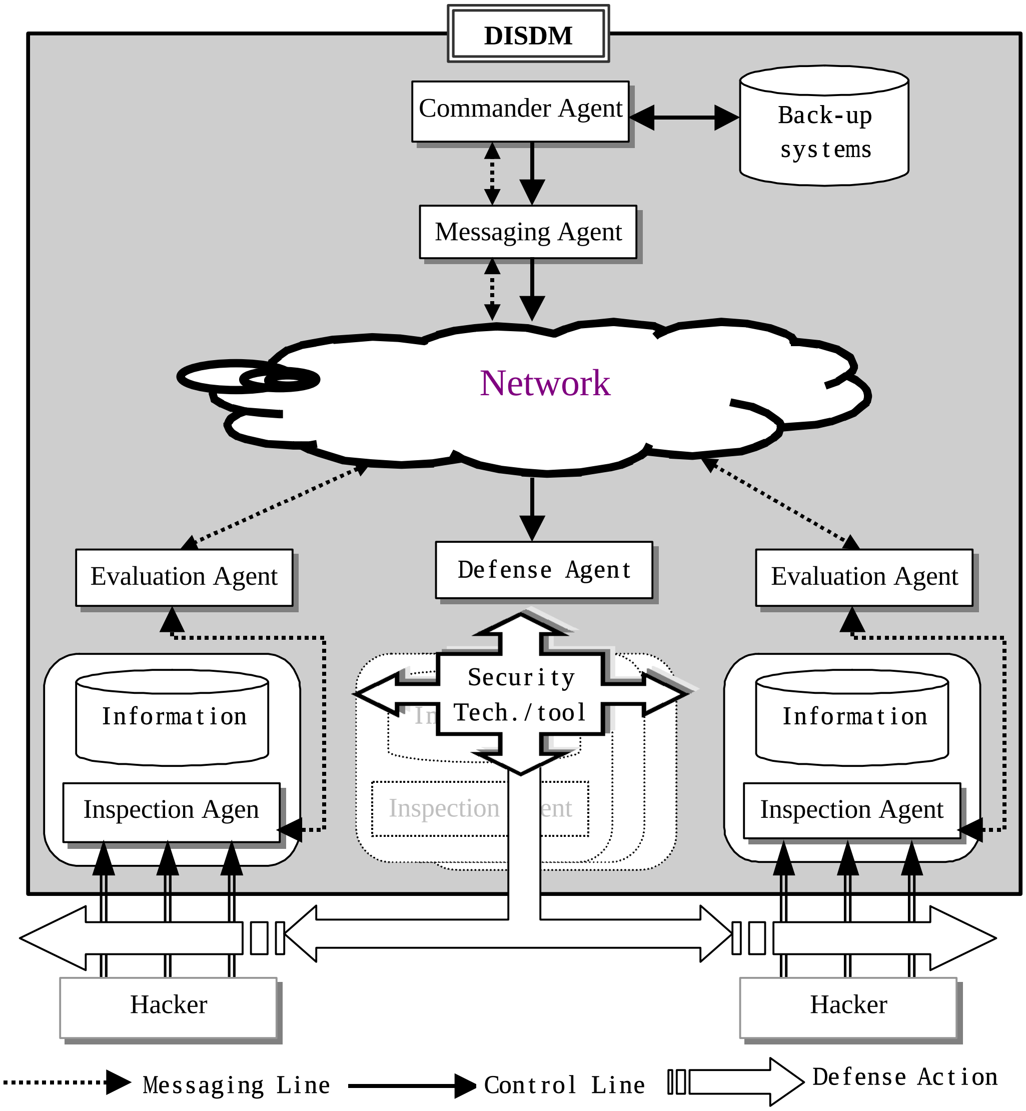

Figure 3. The architecture of DISDM

- *Inspection Agent*: installed in a local system, always monitors possible information attacks.
- *Evaluation Agent*: according to inspection agent's information attack reports, evaluates the threat condition (5 classes [19]) and decides whether the attack is minor taking normal actions is quite enough, or could be a widespread attack that needs to report to central unit (command agent) to alert all agencies to increase awareness activities.
- *Messaging Agent*: installed between command agent and evaluation agent that charged in integrating all inspection agents' attack information and mandatory reporting to central command agent.
- *Defense Agent*: plays the role to implement IW-D. It executes sound security strategies such as authentication, encryption, security management, repair/reload damage software, … etc. to defense all possible information attacks.
- *Command Agent*: is the core role of DISDM that declares state of emergency, disconnects all unnecessary network connections, turns on real-time audit for critical systems and implements damage repair when information systems were under attacked.

Based on the DISDM architecture, we outline its workflow as follows (Figure 4):
(1) Each inspection agent (installed in local system) 24-hour monitors possible intrusion or

information attack. It employs a new IDS technique called "*A*utonomous Agents For Intrusion Detection (AAFID)" that could detect almost all intrusions precisely [18]. When a possible system intrusion was detected it reports the situation to local evaluation agent.

(2) Evaluation agent charges with analyzing inspection agent's information attack reports and evaluates the threat conditions. If the attack is minor (below class 3) it could ask local system to disconnect all unnecessary network connections, increases in incident monitoring for patterns across a wide range of variables and expects command agent to alert all agencies to increase awareness activities. The criteria of threat condition are according to Kuo's evaluation model [19].

(3) Command agent accepts all agencies' threat reports (through messaging agent), analyzes these reports and decides the threat levels. There are five levels of threats in DISDM. When no information attacks reported from distributed agent system it is level 1 threat. DISDM takes normal actions. But when command agent analyzes all agencies' threat reports and finds a level 5 information attacks happed it immediately declares full-scale information warfare defense, asks all agencies to disconnect all network connections and stop application systems, implements information system damage assessment and damage systems repair/reload etc. Other attack threat levels and corresponding response are given in table 1. It is worth to note that the criterion of threat level is determined as follows:

Let L be the threat value of a MISDM network system. It is defined by following equation:

$$L = \frac{\sum_{i=1}^{n} w_i \times \tau_i}{n} \quad (1)$$

where n is the number of application systems in a MISDM network.

$0 < w_i \leq 1$ is the weight of information resource in i^{th} application system.

$1 \leq \tau_i \leq 5$ is the threat condition of i^{th} application system.

Using equation (1), we can determine the threat level as follows:

$$Level = \begin{cases} 1 & if\ 0 < L \leq 1 \\ 2 & if\ 1 < L \leq 2 \\ 3 & if\ 2 < L \leq 3 \\ 4 & if\ 3 < L \leq 4 \\ 5 & if\ 4 < L \leq 5 \end{cases} \quad (2)$$

Notes that (1) is a heuristic equation. DISDM manager could adjust the weight w_i according to his/her own experience, or a MISDM network committee could discuss together to decide the weight of each application system.

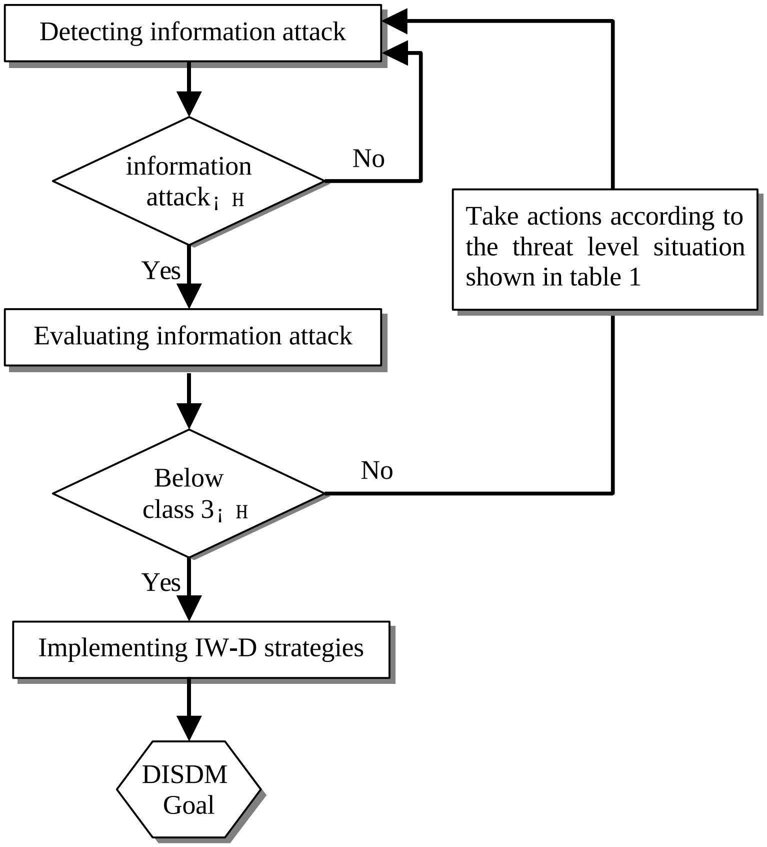
Detecting information attack
information attack¡ H
No
Yes
Evaluating information attack
Take actions according to the threat level situation shown in table 1
Below class 3¡ H
No
Yes
Implementing IW-D strategies
DISDM Goal

Figure 4. The workflow of DISDM

Table 1. DISDM threat-level situation and response

CONDITION	SITUATION	RESPONSE
Level-1 (Normal)	• Normal threat-crime /incompetents	• Normal actions and requirements.
Level-2 (Pertubation)	• About 10% increase in incident reports, regional or functionally base.	• Increase in incident monitoring and look for patterns across a wide range of variables. • Alert all agencies to increase awareness activities. • Begin selective monitoring of critical elements.
Level-3 (Heightened Defense Posture)	• About 20% increase in all incident reports	• Disconnect all unnecessary network connections • Turn on real-time inspection systems (Local inspection agent always monitors possible information attack and shares information with other inspection agents) • Inspection agent informs the authorities (evaluation agent) and urges it to evaluate the IW-D condition.

Level-4 (Serious)	• Major regional of functional events that seriously undermine national interests	• Immediately disconnect all network connections and implement alternate routing. • Command agent begins "aggressive" forensic investigations and asks first-line agents (messaging agent) frequently reporting to it. • Defense units (defense agent) uses IW-D techniques/tools to defense all possible information attack.
Level-5 (Brink of War)	• Widespread incidents that undermine	• Immediately disconnect all network connections and stop application systems. • Command agent declares full-scale information warfare defense, and implements information system damage assessment. • Damage systems repair/reload.

4. Conclusion and the future work

According to U.S Joint-War-Fighter S&T plan, IW is defined as: *the action taken to achieve information superiority by affecting adversary information, information processes, information systems and computer-based networks, while defending ones own information, information-based process, information systems and computer-based networks* [1]. Therefore, protecting critical information resources becomes the main target in IW. Traditional information security methods usually use passive mechanisms such as security management, firewall, system guards, etc. to protect information resources. These methods are “static” that cannot detect possible information attacks before these really happen. Intrusion Detection System (IDS) can takes the initiative in detecting hackers or crackers. However, IDS is a client-server protection mechanism that will cause the network traffic burden.

To solve the mentioned problems, we proposed an agent-based Dynamic Information Security Decision Model (DISDM) that dynamically integrates all distributed information security systems to achieve efficient IW-D. Basically, the DISDM includes five kinds of agents (inspection agent, evaluation agent, messaging agent, command agent, and defense agent.). Among them, command agent is the core of DISDM that is an intelligent agent. Others are mobile agents. They work together to perform effective network security. In concluding the benefits of the DISDM are follows:

- It can 24-hour automatically drive the intrusion inspection by dynamically executing sound security strategies and recover the damages when systems have been under attack. Thus, it is expecting to yield IW-D cost-effective solutions.
- Since it is a distributed system it dramatically reduces the load and solve unsteadiness on the network based on its abilities such as pro-activity and portability.
- Because of operation distributed, hackers are unlikely to wreck the whole system.

The DISDM project is currently in the implementation phase. Agents in DISDM are written in IBM Aglets [16]. The future work is to install the system and evaluate its performance.

5. Reference

[1] Gowar Kuo-Hua His, “Military decision-making process for Information Warfare”, NDMC’99, 1999

[2] "FAQ:Network Intrusion Detection System", http://www.robertgraham.com/pubs/network-intrusion-detection.html

[3] D. E. Denning, “An Intrusion-Detection Model”, *IEEE Transactions on Software Engineering*, Vol. SE-13. No. 2, pp.222-232, 1987

[4] S. Kumar, Classification and Detection of Computer Intrusions, Purdue University Ph.D. Disserstion, 1995

[5] "A Method of Tracing Intruders by Use Mobile Agents," http://power2.nsysu.edu.tw/conference/INET2000/inet99/4k/4k_2.html

[6] Kenny Hung, Network System Intrusion and Protection, Unalis Publish, 2000

[7] "Agents FAQ" http://www.cs.umbc.edu/agentslist

[8] FTP Software Inc., Introduction to Intelligent Agents - White Paper, Andover, MA, http://www.ftp.com/Cyber-Agents

[9] Foner, L.N. “What’s An Agent, Anyway? : A Sociological Case Study,” http://www.cs.umbc.edu/kqml/kqmlspec/spec.html

[10] Kalakota & Whinston, *Frontiers of Electronic Commerce*, Addison-Wesley, 1997.

[11] J. M. Bradshaw, *Software Agents*, AAAI Press/The MIT Press, 1997.

[12] Tung Bui & Jintae Lee, "An agent based frame for building decision support systems", *Design Support Systems*, 25, pp.225-237, 1999.

[13] Nicholas R. Jennings and Michael J. Wooldridge, *Agent Technology Foundations, Applications, and Markets*, Springer-Verlag, 1998.

[14] Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge, "A Roadmap of Agent Research and Development," *Autonomous Agents and Multi-Agent Systems*, 1, pp.275-306, 1998.

[15] Vu Anh Pham and Ahmed Karmouch, "Mobile Software Agents: An Overview", *IEEE Communication Magazine*, July, 1998

[16] Chih-Lin Hu and Wen-Shyen E. Chen, "Mobile Agents Collaboration for Information Gathering," *Workshop on Distributed System Technologies & Applications*, pp. 537-546, May 14-15 1998, NCKU, R.O.C

[17] Danny B. Lange , Mitsuru Oshima ,*Programming and Deploying Java Mobile Agents with Aglets*, Addison Wesley Longman Inc,1998.

[18] Eugene H. Spafford and Diego Zamboni, “Intrusion detection using autonomous agents”, *Computer Networks*, 34(4), pp.547-570, October 2000.

[19] Kuo Chung-Hsin, “Dynamically Controlling Mechanism for Multilevel Collaborative Detection in Network Security”, Master Thesis, Graduate School of National Defense Information, National Defense University, Taiwan, 2002.