Efficient Method for Detecting Worm Virus based Bloom-like Connection Behavior

Jangwon Choi[1], Jaewook Lee[2], Jahwan Koo[2], Byungyeon Park[1], Wonhyuk Lee[1], and Seongjin Ahn[3]

[2] Korea Institute of Science and Technology Information,
Eoeun-dong 52, Yuseong-gu, Daejeon city, Korea, 305-806
{jwchoi, bypark}@ksc.re.kr, livezone@kisti.re.kr
[2] Dept. of Computer Engineering, Sungkyunkwan University,
300 Chunchun-Dong, Jangan-Gu, Suwon, Korea, 440-746
{jwlee, jhkoo}@songgang.skku.ac.kr
[3] Dept. of Computer Education, Sungkyunkwan University,
53 Myungryun-Dong, Jongro-Gu, Seoul, Korea, 110-745
sjahn@comedu.skku.ac.kr

Abstract. The effort required for detecting worm viruses, that threaten the reliability and stability of network resources, is in the process of advancing, demanding increasingly sophisticated resources. Pattern-based worm virus detection systems use detection methods, which focus on pattern analysis for specific worm viruses. In the event of a different attack method, or a new attack occurs, current systems suffer from the problem of being unable to detect the worm virus quickly. This paper proposes a worm virus detection system that focuses on a common feature of worm viruses, which attempt many connections in a scanning process. The central contribution of the proposed system is to decrease the response time of an attack.

1 Introduction

The effort required for detecting worm viruses that threaten the reliability and stability of network resources is in the process of advancing, demanding increasingly sophisticated resources. Examples of these include, [7], [8], [9].

A worm is defined as a malicious code (standalone or file-infecting) that propagates over a network, with or without human intervention [1]. The scanning process is required to search and find the next attack target in the propagated worm virus [2]. The host infected by the worm virus is characterized by accesses from a large number of IP address, occurring rapidly over a short duration.

Many systems are available for worm virus detection. Pattern-based worm virus detection systems use a detection method, which focuses on the pattern analysis regarding a specific worm virus. When the attack form changes, or a new attack occurs, the systems suffer from the problem of being unable to detect worm viruses quickly. In this paper, a worm virus detection system, which analyses the characteristics of the scanning process, is proposed [5].

This paper is organized as follows. Section 2 describes other related research. Section 3 introduces the worm virus detection algorithm. Section 4 discusses implementation of the worm virus detection system. In Subsection 4.1, the architecture of the worm virus detection system is described. Subsection 4.2 discusses the construction of a manager system. Subsection 4.3 discusses the construction of an agent system. The test result is addressed in Subsection 4.4. Finally, a short conclusion, including a summary of future work, is presented in Section 5.

2 Related works

2.1 Characteristic of Worm Virus

Worms are typically considered to be one category of malicious code, actively propagating over a network, with or without minimal user intervention. In contrast to traditional viruses, which require user intervention to spread from one machine to another, worms are sometimes thought of as requiring little or no human assistance in order to spread. Representative worm viruses include Code Red, and Nimda.

Generally speaking, worms can be classified into two types, direct and indirect, depending on the attack method [2]. Worms classified as a direct, always obtain power illegally through communicating with remote computers, in particular, the lack of security, and the bugs of a program are focused on. In an actual environment, many direct worms obtain control via the well known buffer overflow technique. Due to this characteristic, direct worms obtain control in a very short time, and therefore are characterized by higher mobility and transfer speed. Indirect worms always obtain control using indirect methods. Indirect worms generally combine with other malicious code, in order to control other computer systems – for example, computer viruses, and Trojan horses.

The operations of direct worms can be divided into the following four steps.

1. Scanning: search and find the next attack target.
2. Obtain control: obtain control through weaknesses in the target computer.
3. Hooking back: transfer the malicious worm and necessary program to target computer.
4. Install and run: install the main program and obtain control.

Worm viruses constantly search for the next attack target. Target selection is based on a random selection of IP addresses within a specified range [5]. The random generator is seeded with the creation time of the instance of the worm, thus ensuring each new instance will exert unique random target selection behavior. The infected host approaches the attacking target of the selected IP address. The worm analyzes the vulnerability ports of the attack target. The infected host, copies the virus program using the vulnerable port. The new infected target executes the worm virus program, which was copied from the original infected host. The worm virus code now automatically executes, when appropriate. The newly infected host repeats the process by which it was infected, infecting a different machine.

2.2 Network Scanning Method

The effective method for retrieving TCP network services is scanning the attack target networks TCP ports, from an external source. The TCP port scanning technique, used in worm attacks, is used to classify events as worm related behavior [3].

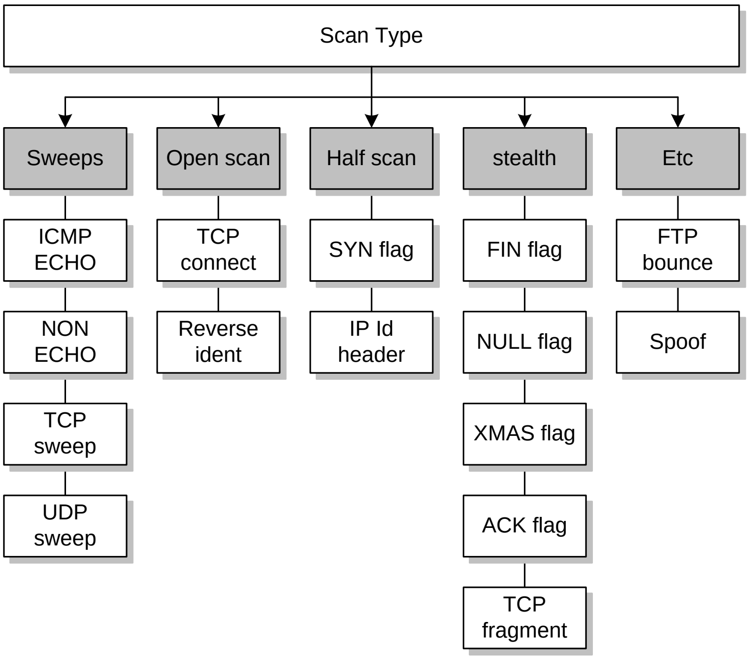

Fig. 1. Type of Network Scanning

Each scanning technique remotely researches the attack target network. This is achieved, order to identify the ports, which are opened, closed or filtered. In addition, the effect network topology and security mechanisms applied to the attack target host and network, are received. Network scanning is simple in terms of logging and detection, but it may be challenging to receive an accurate result regarding the ports which are opened, closed, or filtered. To overcome this challenge, for example, the stealth scan method sets the flags of packets, in order to bypass the security tools. This method either results in a success or failure, depending on the network environment and security policy of the target.

2.3 Intrusion Detection System

An Intrusion Detection System (IDS) [6] is a software/hardware tool used to detect unauthorized access to a computer system or network. This may take the form of detecting attacks by skilled hackers, or beginner hackers, using automated tools.

An IDS is required to detect all types of malicious network traffic and computer usage. This includes network attacks regarding vulnerable services, data driven attacks on applications, host based attacks such as privilege escalation, unauthorized logins, access to sensitive files, and malware (viruses, Trojan horses, and worms).

An IDS is composed of several components, sensors which generate security events, a console to monitor events, alerts to control the sensors, and a central engine that records events logged by the sensors in a database and uses a system of rules to generate alerts from security events received. There are several ways to categories an IDS, depending on the type and location of the sensors and the methodology used by the engine to generate alerts. In many simple IDS implementations, all three components are combined in a single device or appliance.

3 Algorithm of Worm Virus Detection

There are many types of worm viruses and the number of active patterns is considerable. Worm virus detection must detect common activity pattern based on the worm virus. In this paper, worm virus detection uses the feature of the scanning process in the activity method of the worm virus. The worm virus exists in the scanning process, searching for the next target automatically. Infected hosts use random IP addresses for searching the next attack target.

The worm will attempt a connection with the IP address, which is not used, because the IP addresses are generated randomly. In addition, the worm virus searches numerous IP addresses in a short duration, to promote rapid propagation. When this feature is used, it will be able to detect worm viruses with a communicating IP address, for a basis inside unit time. This time will be able to consider the method used to install an agent system in all hosts. However, in this paper, a method of installing an agent system at a point where all traffic passes, is proposed. All traffic is analyzed from this point. The agent system analyzes the collected packets, and compares the source IP ad-dress with the destination IP address. The system investigates each IP address and the number of IP addresses used in communication.

In this paper, in order to obtain the maximum amount of effort, the packets of three types are monitored. First, the Address Resolution Protocol (ARP) [4] packet is analyzed. This investigates cases of approaching a different host. Investigation items are not the number of ARP request packets, but source and destination IP addresses of the ARP request packet. When requesting the MAC address of the destination IP address, where the same source IP address is numerous, the host will be able to detect scanning from the network inside. Specially, some viruses use this method and are able to detect the worm virus easily. This method has characteristics of the following. When the IP address is a.b.c.d, a.b.c is fixed, d is created randomly and the virus searches the next destination IP address. In addition, when a MAC address is obtained as already known, considering the point of which an ARP request packet is not transmitted. This investigates the source and destination IP addresses of the each packet. If the result is the same network, it is a method, which investigates the host number used in the process of connecting. This method does not investigate other fields, except source and destination IP address field, in order to provide maximum detection efficiency. The last method is used in the case of scanning, initiated from the external network. This previously contains two methods, considering a point of which most worm viruses use similar vulnerability. Most worm virus attack well-known ports used by the Windows operating system. For example, 135(RPC), 445(LSASS draw-back), 80(WebDAV), 139(NetBIOS), etc. Consequently, the method uses analyses of specific ports from received packets. If the port of the received packet is contained in the port scan target port, the source and destination IP addresses are compared and regarded as a management target IP address from the connection host number recorded.

```
packet ← New Captured Packet
if (packet = ARP)
  then Database.arp_mark[packet.source_ip][packet.target_ip] = checked
else if (packet = IP)
  then if (packet.source ip ∈ Managed IP List and packet.target_ip ∈ Managed IP List)
    then Database.ip[packet.source_ip][packet.target_ip] = checked
  if (packet = TCP) or (packet = UDP)
    then if (packet.target_port = WeakPort List)
      then if (packet.target_ip ∉ Database.port[packet.source_ip] List)
        add packet.target_ip in Database.port[packet.source_ip] List
```

Fig. 2. Algorithm of Worm Virus Detection (Packet Monitoring)

This system collects information using packet monitoring, presented above. The connection host threshold value of the ARP, IP and port are both set. The system compares the management target IP address list with collected data. If it is larger than the threshold value, a host infected by to the worm virus is recognized.

```
ɑ ← ARP threshold value, β ← IP threshold value, ɣ ← Port threshold value
for ip in Managed IP addresses List
  arp_count ← count checked database.ARP[ip]
    if (arp_count > ɑ )
      then notify ip is infected.
    else
      initalize database.ARP[ip]
  ip_count ← count checked database.IP[ip]
    if (ip_count > β )
      then notify ip is infected.
    else
      initalize database.IP[ip]
```

```
port_count ← count checked database.PORT[ip]
  if (port_count > $\gamma$ )
    then notify ip is infected.
  else
    initalize database.PORT[ip]
```

Fig. 3. Algorithm of Worm Virus Detection (Counting Connections)

The system will be able to consider methods of studying network traffic, for accurate worm virus detection. The patterns of normal network traffic are analyzed periodically a normal use pattern is studied through database construction. This function can be used to analyze suitability in different network environments. In addition, if the number of connecting hosts increases suddenly, based on the records in the database, the machine is infected by a worm virus.

4 Worm Virus Detection System

4.1 System Architecture

The entire system consists of management system and agent system, with the former managing an agent system and the latter installed on each broadcast domain, in order to capture and analyze packets.

This system is separated by a manager, and an agent for agent integrated management. This structure distributes the system load and improves efficiency. The entire system operation manages the agent system where the manager system exists in each network, and the manager system instructs management policies of the agent system by which the agent system enforces. The agent system exists in each broadcast domain and manages the corresponding broadcast domain.

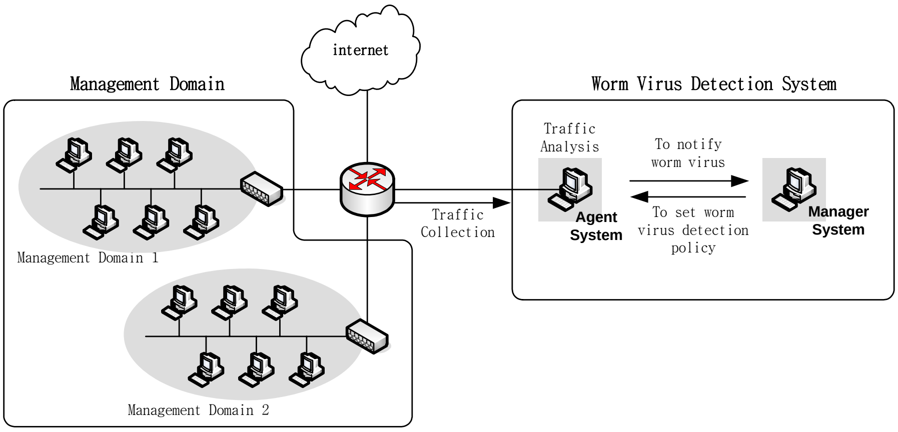

Fig. 4. System architecture

When the initial agent creates the network, the manager registers in the manager system, and the agent system collects network information of its network. The collection of network information accomplished periodically. This minimizes the packet creation in order for collection of network information. The agent system transmits the collected information to the manager through network monitoring. They wait for the policy accomplishment message from the management system. The management system is able to monitor information regarding the management network in real-time. This information is recorded in a database, where the particulars of the network are specified.

4.2 Construction of Manager System

The management system provides monitoring functions, which are used in the present agent conditions in the current network. In using this system, the manager will be able to confirm traffic information, collected in real-time from the agent system. The agent system will establish the policy which follows various situations. Figure 5 is the module composition of manager system. Each module function of the manager system becomes modulation in the center and consists of a centralization management module, worm virus detection module and report module.

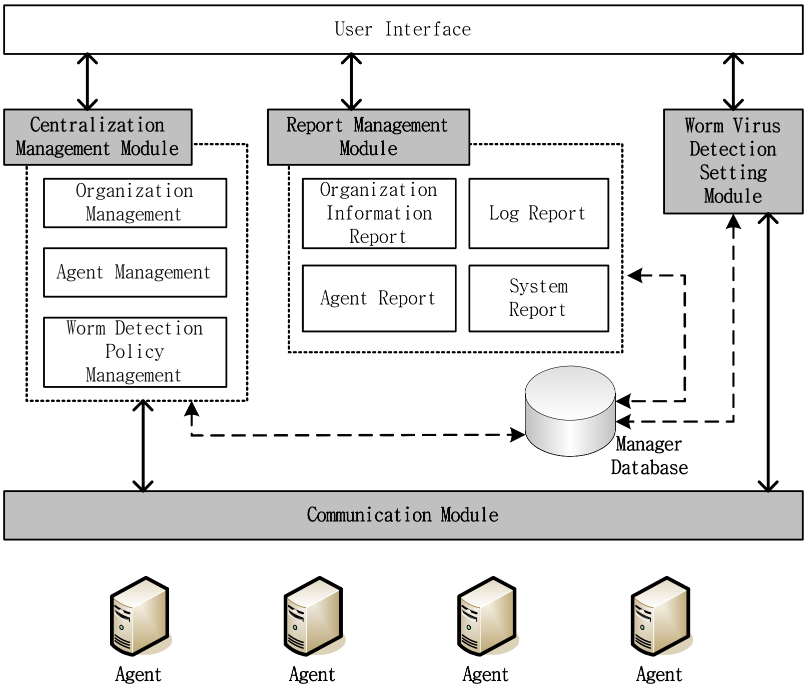

Fig. 5. Modules of Manager System

The centralization management module is composed of functions which include organization management in order to manage distributed agent systems and agent system management, manages the worm virus detection policy which it sets from the agent system. In addition, the worm virus detection setting module sets the threshold value for worm virus detection regarding each detection method. This selects the vulnerability ports for port connection investigation. This will also be able to set directly, as necessary. A set value is used for worm virus detection in the agent system, through the worm virus detection setting module. The report management module is composed of organization information, agent information, log, and system report functions.

4.3 Construction of Agent System

The agent system is installed on each broadcast domain in order to capture and analyze packets. It notifies the manager system and sets worm virus detection to follow the policy of the management system. In addition, it plays a role in preventing the spread of the worm virus through worm virus detection functions. The agent system is composed of agent management, packet monitoring, packet creation, and study and judgment modules.

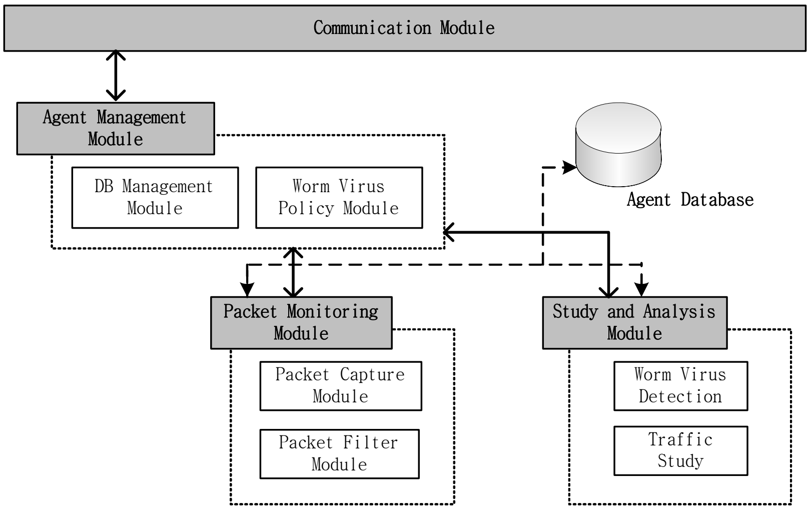

Fig. 6. Module of Agent System

The agent management module is composed of functions which manage the agent database and worm virus policy, received from the management system. The packet monitoring module is the function which collects all resources automatically. For example, IP address, MAC address, system name, and user name. Packet monitoring is accomplished in a real-time and the collected data is transmitted periodically to the management system. This provides assistance in policy decisions. In addition, it collects and analyzes packets and collects information regarding the worm virus through the packet monitoring module. The monitoring module separates the ARP packet, IP address of communicating in same network, TCP and UDP packets of specific ports. In addition, the IP address of a separate packet is stored in a database.

For accurate detection of the worm virus, the method of studying the normal use pattern of the management network is important. In the case of ARP records, a destination IP address per source IP address, and investigation of the MAC address which it requests an address for a certain time, are important. IP packet investigation is only deployed in cases of communicating over the same network. In addition, each IP address of the management area investigates the number of IP addresses, which are transmitted for unit time. This includes, the cases of analyzing TCP packets and comparing the source IP address when each IP address in the management area attempt a connection at a specific port. Periodically, it counts the number of the connected hosts for each IP address. In the event this value exceeds a threshold value or increases the number of connected hosts, the agent system detects the worm virus. In addition, the previous stored database data is used to understand the connection number of the host, from comparison of the effect in each hosts natural environment.

4.4 Test and Analysis

In this paper, worm virus detection is tested, by setting the threshold value of the environment as shown in Figure 4. The one host of the management target host is used to connect to the rest of the management target host. The host is setup as an infected host, and the system is tested continuously. An experiment of when the host connects the threshold value over a unit time, increasing the connection number more than the connection number of previously, is shown.

Table 1. The result of worm virus detection test

No.	Connetio n threshold	Increasement threshold	Connecte d host	Detectio n
1	50	8	66	Yes
2	50	8	8	No
3	50	8	30	No
4	50	8	6	No
5	50	8	48	Yes

The 1st test confirmed a worm virus because the connection number of the host is 66 and the thresh-old value is 50. From the 2nd test to 4th test, the threshold value is not exceeded. The 5th test confirms the worm virus because the value increases in octuple, over more than the value of the 4th test but the value does not exceed the threshold value.

As shown above, this system detects the worm virus when the connection number of the present is greater than connection number of the previous connection, or it surpasses the threshold value, where the manager sets are based on a connection number of the host. The common feature of the worm virus is used, and the system therefore has the merit of being able to detect new worm viruses, and new strains of current worm viruses.

5 Conclusion

In this paper, a method of detecting the worm virus quickly through the functions, which detect the host infected with the worm virus, is proposed. . In addition, the proposed method, which detects abnormal situations based on the value obtained as the system is studied periodically, and compared to when the system acts in normal situations, this method decides situations of the worm virus depending on the absolute threshold value.

In the proposed method, the agent and management nodes are separated, therefore providing the capability of managing many integrated networks. In addition, the data collected, is used to decide on a specific policy, the agent reflects the desired policy, and each agent manages the corresponding network. In order to study detection of worm viruses, the abnormal behavior detection method is applied, this is based on the number of connecting hosts. For study and detection of a worm virus, abnormal behavior detection methods are applied, based on the number of connecting hosts.

In the future, through the simulation, the system, will be executed in a real situation, this will be used to research how efficiency can be obtained. In addition, the problem points occurring from simulation situations will be discovered, through this, the system can be continuously improved. In the future, research is planned not only for worm virus detection, but also for interception mechanisms for dealing with worm viruses quickly.

References

[1] Darrell M. Kienzle and Matthew C. Elder, "Recent worms: a survey and trends," in Proceedings of the 2003 ACM workshop on Rapid Malcode, 2003

[2] Jason C. Hung, Kuan-Cheng Lin, Anthony Y. Chang, Nigel H. Lin and Louis H. Lin, "A bahavior-based anti-worm system," In Proceedings on AINA'03, China, 2003

[3] Matta Security Limited, "IP network scanning & reconnaissance," 2002

[4] David C. Plummer, "An ethernet address resolution protocol," RFC 826, 1982

[5] Vincent Berk and George Bakos, "Designing a framework for active worm detection on global networks," in Proceedings on IWIA'03, 2003

[6] Wagner D. and Dean R., "Intrusion detection via static analysis," in Proceedings of 2001 IEEE Symposium on Security and Privacy, 2001

[7] Jahwan Koo, Seongjin Ahn, Jinwook Chung, "Network blocking algorithm and architecture for network resource and security management," in Proceedings of International Scientific-Practical Conference "Problems of Operation of Information Networks", 2004

[8] Wonwoo Choi, Hyuncheol Kim, Seongjin Ahn and Jinwook Chung, "A network access control system using on address spoofing and VLAN filtering," The 4th Asia Pacific International Symposium on Information Technology, 2005

[9] Kyohyeok Kwon, Seongjin Ahn and Jinwook Chung, "Network security management using ARP spoofing," in Proceedings of ICCSA 2004, 2004

Biography

▲ Name: Jangwon Choi
Address: Korea Institute of Science and Technology Information, Eoeun-dong 52, Yuseong-gu, Daejeon city, Korea, 305-806
Education & Work experience:
1998. 8: received M.S. degree in Electrical Engineering from Hongik University
2005. 8: finished Ph.D. course in Computer Engineering from Korea University
1998. 8~Current: Researcher in Korea Institute of Science and Technology Infromation
Tel: +82-42-828-5132
E-mail: jwchoi@ksc.re.kr

Other information:

▲ Name: Jaewook Lee
Address: Dept. of Computer Engineering, Sungkyunkwan University, 300 Chunchun-dong, Jangan-gu, Suwon, Korea, 440-746
Education & Work experience:
2005. 2: received B.S. degree in information and communication engineering from Sungkyunkwan University
2004 .3~Current: working towards the M.S. degree in computer engineering with the School of Information and Communications Engineering, Sungkyunkwan University
Tel: +82-31-290-7212
E-mail: jwlee@songgang.skku.ac.kr
Other information:

▲ Name: Jahwan Koo
Address: Dept. of Computer Engineering, Sungkyunkwan University, 300 Chunchun-dong, Jangan-gu, Suwon, Korea, 440-746
Education & Work experience:
1995. 2: received B.S. degree in Information Engineering from Sungkyunkwan University
1997. 2: received M.S. degree in Information Engineering from Sungkyunkwan University
2002. 3~Current: working towards the Ph.D. degree in Electrical Electronics and Computer Engineering from Sungkyunkwan University
Tel: +82-31-290-7212
E-mail: jhkoo@songgang.skku.ac.kr
Other information:

▲ Name: Byungyeon Park
Address: Korea Institute of Science and Technology Information, Eoeun-dong 52, Yuseong-gu, Daejeon city, Korea, 305-806
Education & Work experience:
2004. 8: received M.S. degree in Education Information from Kongju National University
1990. 5~Current: Researcher in Korea Institute of Science and Technology Infromation
Tel: +82-42-869-0525
E-mail: bypark@ksc.re.kr
Other information:

▲ Name: Wonhyuk Lee
Address: Korea Institute of Science and Technology Information, Eoeun-dong 52, Yuseong-gu, Daejeon city, Korea, 305-806
Education & Work experience:
2001. 2: received B.S. degree in Computer Engineering from Sungkyunkwan University
2003. 2: received M.S. degree in Computer Engineering form Sungkyunkwan University
2003. 3~Current: Researcher in Korea Institute of Science and Technology Infromation
Tel: +82-42-869-0648
E-mail: livezone@kisti.re.kr
Other information:

▲ Name: Seongjin Ahn
Address: Dept. of Computer Education, Sungkyunkwan University, 53 Myungryun-dong, Jongro-gu, Seoul, Korea, 110-745
Education & Work experience:
Received the B.S., M.S., and Ph.D. degrees in information and communication engineering from Sungkyunkwan University in 1988, 1990, and 1998
1990. 2~1995. 5: Researcher in Electronics and Telecommunications Research Insititute (ETRI)
Current~: Associate professor Department of computer education, Sungkyunkwan University
Tel: +82-31-290-7212
E-mail: sjahn@comedu.skku.ac.kr
Other information: