# 악성 코드 분류 및 명명법에 관한 연구†

조선대학교 최준호 · 곽효승* · 공현장 · 김판구**
한국정보보호진흥원 이병권 · 오은숙

## 1. 서 론

인터넷 사용자의 증가와 개방적인 인터넷 환경으로 인해 악성코드에 대한 지식을 충분히 숙지하지 못한 인터넷 사용자들은 악성코드에 아무런 보호 없이 노출되고 있다. 최근에는 악성코드의 종류나 출현 횟수가 크게 늘고 있고, 그에 따른 피해액도 늘어나고 있는 추세이다. 현재 많은 백신 업체들이 악성 코드에 대한 탐지 및 치료에 대해 많은 성과를 이루고 있지만, 각 백신 업체들마다 각기 다른 분류 지침과 명명법을 내부적으로 마련하여 사용하기 때문에, 정확한 통계자료나 피해상황을 전체적으로 파악할 수 없는 실정이다. 또한, 전문적인 악성 코드에 대한 지식이 없는 대부분의 일반 사용자들은 비록 악성 코드를 발견하였다 할지라도 분류나 이름의 모호함으로 인해 악성 코드에 대처하기란 쉬운 일이 아니다. 이러한 문제점 해결과 악성 코드를 정확히 파악하고, 대처하기 위해서는 악성 코드에 따른 새로운 분류와 명명법이 절실히 요구된다.

각 백신업체의 분류 체계가 상이하기 때문에 한 악성 코드에 대하여 서로 다르게 분류될 수 있다. 또한, 악성 코드 이름은 대체적으로 각 백신업체에서 분류체계에 기반으로 하여 작성하기 때문에, 각 백신업체마다 명명규칙도 다르고, 악성 코드의 이름 또한 각각 다르다. 뿐만 아니라, 각 백신업체들의 분류체계와 명명법은 예전의 도스 운영체제 때부터 사용하던 분류체계와 명명법을 여전히 사용하고, 그 분류체계와 명명법에 새로운 부분을 추가하여 사용하는 게 일반적인 경향이다. 그래서, 최근에 나타나는 악성 코드에 대한 정확한 분류에는 상당한 어려움을 보이고 있다. 한 개의 악성 코드가 여러 형태의 복합적인 악성을 지내고 있을 때, 예를 들어 웜 형태의 악성 코드가 윈도우, 유닉스 등의 여러 운영체제에서 악성 행위를 할 때와 같은 악성 코드 분류는 지금 사용되어지는 분류 지침을 사용하여 분류하기란 쉽지 않다.

본 논문에서는 이러한 문제를 해결하기 위해 국내외 각 백신업체들의 분류지침과 명명규칙을 조사·분석하여 최근 추이에 적합하고, 정확한 악성 코드에 대한 분류를 할 수 있다. 또, 보다 쉽게 모든 컴퓨터 사용자들이 인지하여 대처할 수 있도록 새로운 악성 코드의 분류지침을 마련하고, 분류지침에 의거하여 악성 코드 명명규칙을 제안하고자 한다.

본 논문의 2장에서는 국내외 악성 코드 분류에 대하여 알아보고, 3장에서는 각 백신업체들의 악성 코드 명명법을 알아본다. 4장에서는 새로운 악성 코드 분류지침과 악성 코드 명명법을 제안하고, 5장에서는 결론을 맺는다.

## 2. 국내외 악성 코드 분류 지침

### 2.1 Wildlist 악성 코드 분류 지침

Wildlist는 1993년에 Joe Wells에 의해 처음 시작되었고, 안티-바이러스 소프트웨어 프로그램의 실제적인 유효성을 측정함에 있어서 그 기준으로 널리 활용되고 있다. Wildlist는 전 세계에 널리 퍼져 있는 잘 알려진 바이러스의 리스트이다. WildList에서는 특별한 분류 원칙을 마련하지 않고, 전 세계의 회원들로부터 악성 코드의 출현에 대한 보고를 받는다. 새로운 WildList는 전 세계에 퍼져있는 많은 협력자들로부터 확실히 증명되어진 악성 코드를 받아서 매달 중순경에 작성된다[4][9].

† 본 연구는 2002년도 한국정보보호진흥원 지원으로 수행되었음.
* 학생회원
** 종신회원

WildList는 WildList, Supplemental List, Frequency List로 구성되어 악성 코드 분류를 하고 있다. Wild List와 Supplemental List에서는 "+, -, *"의 심볼들을 사용하여 악성 코드에 대한 정보를 표시하고 있다. "+"는 그 달의 새로이 보고된 악성 코드의 이름 앞에 붙인다. "-"는 그 달의 WildList로부터 Supplemental List로 내려온 악성 코드 이름 앞에 붙인다. "*"는 WildList와 Supplemental List에서 자주 나타나는 악성 코드 명 앞에 붙인다.

WildList는 많은 협력자들, 최소한 두 명의 협력자에 의해 통보되고, 일정 지역이 아닌 전 지역에 걸쳐 발견된 악성 코드 명 목록이다. Supplemental List는 특정 지역에서 자주 발생하지만 다른 지역에서 발견하기 힘든 악성 코드 명이고, WildList로부터 이동한 악성 코드명 목록이다. Frequency List는 각 악성 코드에 대하여 상당히 많은 협력자들에 의해 보고·저장되어진 List이다.

### 2.2 안철수연구소 악성 코드 분류 지침

안철수연구소에서는 악성 코드의 분류를 표 1과 같이 악성 코드의 종류를 중심으로 대분류로 나눈 다음, 대분류를 각각 유형별로 소분류하여 악성 코드를 분류한다.

이러한 분류 방법은 안철수연구소가 도스 운영체제때부터 악성 코드를 분류하고, 예전의 분류에 새로운 악성 행위를 추가하는 형태의 분류를 하고 있다[5].

표 1 악성 코드 분류 - 안철수연구소

| 종 류 | 내 용 |
|---|---|
| 바이러스 | 윈도우파일, 도스파일, 부트, 부트/파일, 스크립트, 팜, 리눅스, 매크로 |
| 트로이목마 | 윈도우파일, 도스파일, 백도어, 스크립트 |
| 웜 | 윈도우파일, 도스파일, 스크립트, 매크로 |
| 가짜(Hoax) | 가짜 |
| 조 크 | 조크 |
| 기 타 | 보안상 위험 |

### 2.3 ㈜하우리 악성 코드 분류 지침

㈜하우리는 악성 코드에 대한 분류를 다음의 표 2와 같이 대분류와 대분류에 다시 악성 코드가 행하는 증상에 따른 분류를 추가하는 2단계의 악성 코드 분류 지침을 가지고 있다. ㈜하우리의 이러한 분류 방법은 비교적 세분화되어 있다[6].

표 2 악성 코드 분류 - ㈜하우리

| 종 류 | 내 용 |
|---|---|
| 유형별 분류 | 매크로, 스크립트, 도스, 가짜, 부트, 도스용 트로이 목마, 윈도우, 트로이 목마, 윈도우 파일, 윈도우조크, IIS 웜, 기타 |
| 증상에 따른 분류 | 하드포맷, 파일생성, 파일삭제, 파일감염, 메일발송, 정보유출, 화면출력, 시스템 정보 변경, 플래쉬 메모리 및 하드디스크 손상, 네트웍/시스템속도저하, 특정음 출력, 파일 손상, 홈페이지 주소 변경, 하드디스크손상, 시스템비정상작동, FAT 파괴, CMOS 삭제, 섹터 삭제, 홈페이지 변경, 내용상 기제, 레지스트리 변경, 파일 파괴, 기본 메모리 감소,특정 포트 오픈, 메시지 박스 출력, 프로세스 종료, 윈도우 종료, 메시지 전송, 파일 삭제 및 하드포맷, 감염 외 증상 없음, 확산 외 특별한 증상 없음, 특정 파일 겹쳐쓰기, 증상없음 등 |

### 2.4 Trend Micro 악성 코드 분류 지침

Trend Micro에서는 표 3에서와 같이 Payload와 Type으로 악성 코드를 분류한다. Payload는 악성 코드가 시스템에 미치는 영향 또는 악성 코드로 인한 피해를 산출하기 위한 종류이고, Type은 악성 코드의 형태이다. 즉, 운영체제 또는 파일의 종류 등에 의한 분류이다. Trend Micro에서도 2단계로 악성 코드에 대한 분류를 한다[7].

표 3 바이러스 분류 - Trend Micro

| 종 류 | 내 용 |
|---|---|
| Payload | Corrupts Hard Disk, Creates Files, Deletes Files, Displays Graphics, Displays Message, Formats Hard Disk, Generates Sound, Hangs System, Modifies Files, Others |
| Type | ActiveX Control, Backdoor, Batch File, Boot, Elf Executable, File Infector, Html, IRC Script, Java Applet, Java Script, Joke, Macro, Shell Script, Trojan, VB Script, Visio 5, VXD, Worm, Others |

## 3. 국내외 악성 코드 명명법

### 3.1 안철수연구소 악성 코드 명명법

안철수연구소의 악성 코드에 대한 명명의 기본 원칙은 제작자의 이름, 발견장소, 프로그램 길이, 내부

문자열의 특징적인 증상을 기본원칙으로 한다. 변형 악성 코드의 명명은 증상이나 실행 코드가 조금 변형되었을 때는 B, C 등을 덧붙여 사용하고, 악성 코드의 변형이 심할 경우 II, III 등을 사용한다[5].

아래의 표 4는 안철수연구소의 악성 코드 명명법을 표현한 것이다.

표 4 안철수연구소 악성 코드 명명법

| 접두어 | 이름 | 접미어 |
|---|---|---|
| Win32 | Goner | worm |

### 3.2 ㈜하우리 악성 코드 명명법

㈜하우리에서는 기존에 사용하여 오던 명명법에 대하여 보완·수정하여 새로운 바이러스 명명 규칙을 마련하여 사용하고 있다. 새로운 바이러스 명명 규칙은 "형태(Type).이름(Name).변형정도(A, B, ...).크기(Size)"의 형태로 기존의 명명 규칙에서 플랫폼(OS)과 형태(TYPE)를 같이 사용하여 형태(TYPE)로 바꾸어 표현하였다[6]. 표 5는 ㈜하우리의 악성 코드 명명법의 표현을 보여준다.

표 5 하우리 악성 코드 명명법

| 분류 | 접두어 | 이름 |
|---|---|---|
| I-Worm | Win32 | Goner |

### 3.3 시만텍 악성 코드 명명법

시만텍의 악성 코드 명명법의 기본적인 구성은 "접두사·이름·접미사"의 형태로 이루어져 있으며, 도스형 악성 코드에서는 보통 접두사를 포함하지 않는다. 접두사는 악성 코드가 복제되거나 또는 특성의 표시이며, 이름은 널리 사용되는 패밀리 네임을 의미한다. 접미사는 같은 패밀리 네임을 구별지어 주고 악성 코드의 크기를 표시하여 준다[8]. 표 6은 시만텍의 악성 코드 명명법의 한 예를 보여준다.

표 6 (주)시만텍 악성 코드 명명법

| 접두어 | 이름 | 변형 | 접미어 |
|---|---|---|---|
| W32 | Goner | A | @MM |

### 3.4 CARO 악성 코드 명명법

CARO(Computer Anti-virus Research Organization)는 유럽을 중심으로 한 안티-바이러스 단체로써, 악성 코드에 대한 명명 규칙을 1991년에 제정하였다. 이 단체에서 제시하는 전체적인 악성 코드의 구성은 Family_Name, Group_Name, Major_Varient, Minor_Varient의 4부분으로 이루어져 있으며, 각 부분은 '.'으로 구분한다. 각 부분은 대문자와 소문자, 숫자 0~9, 그리고 특수 문자들을 사용하여 명명하고, 글자수가 20글자 이상일 경우에는 가급적 짧은 이름을 정한다. 초기에 모든 백신업체에서 이 기준에 기반을 두고 분류 및 명명을 해왔다. 그런데, 그 기준은 예전의 도스에 기반을 두고 있어 현실정에는 부적절한 부분이 많다. 현재는 각 회사별로 나름대로의 분류지침 및 명명법을 만들어 사용하고 있다.

### 3.5 백신 업체의 악성코드 명명 비교

제시된 내용을 다음의 표 7에서 서로 비교하여 살펴보면, 업체들마다 같은 악성코드에 대해 서로 다른 명칭을 사용하고 있음을 쉽게 파악할 수 있다.

표 7 백신 업체의 악성 코드 이름 비교

| 안철수연구소 | ㈜하우리 | 시만텍 |
|---|---|---|
| Win32/Klez.worm.H | I-Worm.Win32.Klez.H | W32.Klez.H@mm |
| Win32/FunLove.4099 | Win32.FunLove.4099 | W32.FunLove.4099 |
| Win32/Nimda | Win32.Nimda.D | W32.Nimda.A@mm |

이런 결과로 인하여, 많은 악성코드에 대한 연구에서 악성 코드에 대한 정확한 통계를 파악하는데 많은 어려움이 따른다. 또, 악성 코드의 명칭에 대해 악성 코드에 대한 지식이 없는 대부분의 사용자는 쉽게 어떠한 악성 코드인지 알 수 없을 뿐만 아니라, 악성 코드인지 아닌지도 확인할 수 없다.

## 4. 새로운 악성 코드 분류 및 명명법 제안

본 논문에서 제안한 악성 코드의 분류는 악성 코드에 대하여 여러가지 각도에서 더 세분화하여 분류하였고, 세분화된 분류기준마다 이름을 부여하는 새로운 명명 규칙을 제안함으로써, 악성 코드에 대한 명칭을 통일시키는데 목적이 있다. 이에, 기존의 각 업체들이 사용하여 온 악성 코드 분류지침과 명명법에 기반하여, 환경 변화에 적응하고, 다양한 악성 코드의 내용을 가능한 정확하게 포함하는 새로운 악성

코드에 대한 분류지침과 분류지침에 기반 한 명명법을 제시한다.

## 4.1 악성 코드 분류지침

제시한 새로운 악성 코드에 관한 분류는 악성 프로그램 정의에 의한 분류, 운영체제에 의한 분류, 감염영역(부위)에 의한 분류, 감염 경로에 의한 분류, 악성 프로그램 증상에 의한 분류의 5개의 큰 분류와 함께 자세한 분류를 통하여 어떠한 형태의 악성 코드들도 자세히 분류될 수 있도록 분류 지침을 설계·제안한다.

### 4.1.1 악성 코드의 정의에 의한 분류

악성 코드 정의에 의한 분류는 악성 코드가 다른 프로그램에 기생하는지, 또는 다른 시스템에 자기 복제를 하는지 등을 기반으로 악성 코드를 정의하여 이를 크게 5가지로 분류하였다. 표 8은 악성 코드 정의에 의한 분류이다.

표 8 악성 코드 정의에 의한 분류

| 분 류 | 정 의 |
|---|---|
| 바이러스 | 바이러스의 가장 큰 특성은 복제와 감염이라고 말할 수 있다. |
| 웜<br>(인터넷 웜) | 인터넷을 통하여 시스템에서 시스템으로 자기 복제를 하는 프로그램을 의미한다. |
| 트로이목마<br>(백도어) | 트로이 목마 프로그램은 유틸리티 프로그램 내에 악의의 기능을 가지는 코드를 내장한다. |
| 가짜<br>(Hoax) | Hoax는 전자메일로 다른 사람에게 거짓 정보 즉 루머를 유포를 의미한다. |
| 조크<br>(Joke) | 조크는 트로이목마와 달리 악의적인 목적을 가지지 않고 사용자에게 심리적인 위협 혹은 불안을 조장하는 프로그램을 말한다. |

### 4.1.2 운영체제에 의한 분류

운영체제의 의한 분류의 기준은 악성 코드들이 어떠한 운영체제를 대상으로 하여 활발하게 활동하고 전파되는지에 대한 분류로써, 크게 도스, 윈도우, Linux, Unix, Plam, FreeBSD의 6개로 분류하고 있다. 표 9는 악성 코드의 운영체제에 의한 분류이다.

### 4.1.3 감염 영역(부위)에 의한 분류

감염영역(부위)에 의한 분류에서는 악성 코드들이 주로 어느 영역에서 활동하는지에 대하여, 파일, 부트, 부트/파일, 매크로, 스크립트의 5가지 형태로 분류하였다. 표 10은 악성 코드의 감염 영역(부위)에 따른 분류이다.

표 9 운영체제에 의한 분류

| 분 류 | 정 의 |
|---|---|
| 도스 | MS-DOS 기반에서 활동하는 일반적인 부트, 파일, 부트/파일 바이러스 또는 트로이 목마류 |
| 윈도우 | MS 윈도우 기반에서 활동하는 바이러스 및 악성 프로그램들로 다음과 같이 윈도우 버전별로 구분될 수 있다. |
| Linux | 리눅스 운영체제의 보안상 취약점을 이용하여 실행되는 악성 프로그램들 |
| Unix | 유닉스 운영체제의 보안상 취약점을 이용하여 실행되는 웜 종류들 |
| Palm | Palm 운영체제에서 활동하는 트로이 목마 또는 단순한 겁쳐쓰기 바이러스 |
| FreeBSD | 아파치 웹서버의 취약점을 이용하여 전파되는 웜 |

표 10 감염 영역(부위)에 따른 분류

| 분 류 | 정 의 |
|---|---|
| 파일<br>바이러스 | 일반적으로 실행 가능한 프로그램 파일에 감염, 윈도우에서는 다양한 형태의 실행 파일이 존재하므로 감염되는 파일 종류도 여러가지임 |
| 부트<br>바이러스 | 부트 영역(주부트 섹터, 도스 부트섹터)에 감염되는 바이러스 |
| 부트/파일<br>바이러스 | 부트 영역과 파일을 동시에 감염시키는 바이러스 |
| 매크로<br>바이러스 | MS 오피스의 매크로 기능을 이용하여 문서파일을 감염시키는 바이러스 |
| 스크립트<br>바이러스 | 자바 스크립트 및 비주얼베이직 스크립트로 작성된 웜 또는 바이러스 |

표 11 감염 경로에 의한 분류

| 분 류 | 정 의 |
|---|---|
| 파일실행 | 감염된 파일을 실행 시 감염되지 않는 다른 파일을 감염 |
| 다운로드<br>(FTP, 메신저) | 대상을 다운로드 받고 사용자가 파일을 실행하면 감염 |
| 네트워크<br>(공유폴더) | 랜덤 한 IP 대역을 스캐닝하여 읽기/쓰기 공유된 폴더나 드라이브 파일을 감염 또는 웜, 바이러스가 복사되는 방법 |
| 보안 취약성 | IIS 웹 서버 취약성과 같이 특정 응용 프로그램이나 OS의 취약성을 이용하는 방법 |
| 메일 | MAPI 또는 SMTP 기능으로 메일을 통하여 전파되는 방법 |
| 부팅 | 부팅에 의하여 기본 메모리가 바이러스에 감염 |

### 4.1.4 감염 경로에 의한 분류

감염 경로에 의한 분류는 악성 코드가 어떤 경로를 통해서 감염되었는지에 따른 분류로, 최근에는 인터넷의 보급과 네트웍의 확산에 따라 인터넷이 많은 악성 코드들의 감염 경로가 되고 있다. 특히, 파일실행, 다운로드, 네트워크(공유폴더), 보안의 취약성, 메일 그리고 부팅에 따른 6개의 경로들로 크게 나누어 분류하였다. 표 11은 악성 프로그램이 감염되는 경로에 따른 분류이다.

### 4.1.5 악성 코드 증상에 의한 분류

악성 코드들은 시간이 지남에 따라 다양한 증상을 나타낸다. 증상에 의한 분류는 많은 증상들 중에서 특히 대표적으로 나타나는 주요 증상에 대하여 분류하였다. 하드디스크에 관련된 증상, 파일에 관련된 증상, 시스템에 관련된 증상, 네트워크에 관련된 증상과 특이증상 등 크게 5가지로 분류하였다. 표 12는 악성 코드의 증상에 의한 분류이다.

표 12 악성 코드 증상에 의한 분류

| 분 류 | 정 의 |
|---|---|
| 하드디스크 관련 | 하드와 관련된 악성 프로그램의 증상을 보이는 것. |
| 파일관련 | 바이러스가 특정 파일 또는 파일에 관련된 삭제, 겹쳐쓰기, 삭제 등의 증상을 보이는 것. |
| 시스템 관련 | 레지스트리 키 값을 변경에 의한 시스템 정보를 변경, 시스템의 FAT 파괴, CMOS의 내용을 변경에 의한 부팅 시 에러, 시스템 마비 등의 악성 프로그램에 의한 시스템 관련 증상. |
| 네트워크 관련 | 메일을 발송, 정보 유출, 네트워크 속도가 저하, 다른 PC로 메시지를 전송, 특정 포트 Open. |
| 특이증상 | 메시지의 화면에 출력, 특정 음 출력 등. |

## 4.2 악성 코드의 명명 규칙

앞의 본 논문에서 제안한 악성 코드의 분류지침을 기반으로 하여 제시하는 전체적인 악성 코드의 명명규칙은 크게 5부분으로 이루어져 있으며, 각 부분은 '_'로 구분한다. 또한 두 가지 이상 항목은 '/'로 구분하여 연속 기입한다. 각각의 이름들은 쉽게 그 악성 코드의 정보를 파악할 수 있도록 길지만 정확하게 명명한다. 분류지침에 따른 정의에 의한 분류, 운영체제의 의한 분류, 감염 영역(부위)에 의한 분류, 감염 경로에 의한 분류와 증상에 의한 분류들 각각에 특징들을 표현할 수 있는 이름을 부여한다. 표 13은 앞에서 제안한 분류지침에 기반한 악성 코드의 명명규칙을 표현한 것이다.

표 13 악성 코드 명명법

| 정의에 의한 분류 | 운영체제에 의한 분류 | 감염영역에 의한 분류 | 감염경로에 의한 분류 | 증상에 의한 분류 | 악성코드 이름 |
|---|---|---|---|---|---|
| Worm | Win | File | Mail | HDisk | CHI |

## 5. 결론 및 향후 과제

본 논문에서는 기존의 악성 코드의 분류지침 및 명명법을 보완하여 마련한 통합 분류지침과 명명규칙을 악성 코드의 분류와 이름을 지을 때 사용함으로써, 누구나 악성 코드에 대하여 정확하고 쉽게 정보를 알 수 있다. 기존 백신업체들의 분류지침에서는 한 가지 형태 또는 두 가지의 형태로 악성 코드에 대한 분류지침을 마련하였기 때문에, 그 분류지침에 의해 악성 코드를 표현하기에는 많은 일치하지 않는 악성 코드들이 발생하였다. 본 연구를 통하여 마련된 분류지침을 통하여 분류하였을 때, 자세하고 정확한 악성 코드의 분류가 가능하고, 또 명명규칙을 적용하여 이름을 부여함으로써, 악성 코드에 대한 일률적인 명명을 통한 악성 코드에 대한 정확한 통계와 이를 악성 코드에 대한 연구활동에 기초 자료로 널리 활용하여, 악성 코드에 대한 빠른 대처에도 많은 성과를 기대한다. 본 논문에서 제안한 분류지침을 토대로 안티-바이러스 산업 및 악성 코드 연구 활성화 정책 수립의 기초 자료로 사용할 수 있으며, 악성 코드 정보의 체계화 · 통합화 · 표준화 등에 기여할 수 있다.

## 참고문헌

[1] F.Fernandez, "Heuristic Engine," The 11th International Virus Bulletin Conference, 2001.

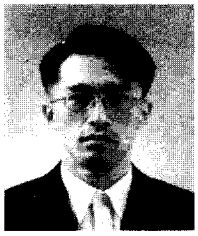
[2] T.Okamoto, "A Distributed Approach to Computer Virus Detection and Neutralization by Autonomous and Heterogeneous Agents," http://www.sys.tutkie.tut.ac.jp/~ishida, 1998.

[3] Sara Hedberg, "Combating Computer Viruses: IBM's New Computer Immune System," IEEE Parallel & Distributed Technology, 1996 Summer.

[4] http://www.thewildlist.com/

[5] http://www.ahnlab.com
[6] http://www.antivirus.com
[7] http://www.hauri.co.kr
[8] http://www.symantec.com
[9] http://www.wildlist.org/


## 최 준 호

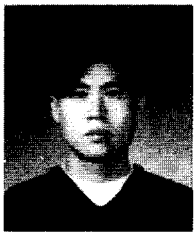


1997 호남대학교 컴퓨터공학과 졸업(공학사)
2000 조선대학교 전자계산학과(이학석사)
2001~현재 조선대학교 전자계산학과 박사과정
관심분야 : 정보보안, 침입탐지시스템, 컴퓨터 바이러스, 영상처리
E-mail : spica@hitel.net

## 곽 효 승

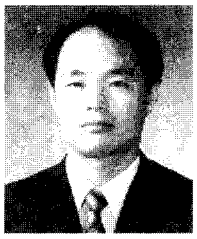
2001 조선대학교 전자계산학과 학사
2001~현재 조선대학교 전자계산학과 석사과정
관심분야 : 정보보안, 바이러스
E-mail : robotsc@mina.chosun.ac.kr

## 공 현 장



2002 조선대학교 전자계산학과 학사
2002~현재 조선대학교 전자계산학과 석사과정
관심분야 : 멀티미디어 정보검색, 컴퓨터 바이러스
E-mail : kisofire@mina.chosun.ac.kr

## 김 판 구



1988 조선대학교 컴퓨터공학과(공학사)
1990 서울대학교 컴퓨터공학과(공학석사)
1994 서울대학교 컴퓨터공학과(공학박사)
1995~현재 조선대학교 컴퓨터공학부 교수
관심분야 : 시스템 보안, 운영체제, 정보검색, 영상처리
E-mail : pkkim@mina.chosun.ac.kr

## 이 병 권

1989 전북대학교 전자계산기공학과(공학사)
1992 포항공대 전자계산학과(공학석사)
1992~1997 POSCON 기술연구소 대리
1997~2001 LG EDS 과장
2001~현재 한국정보보호진흥원 선임연구원
관심분야 : 시스템 보안, 운영체제, 병렬처리
E-mail : byungkle@cert.certcc.or.kr

## 오 은 숙

1999 순천향대학교 전산학과(공학사)
2001 순청향대학교 전산학과(공학석사)
2000~현재 한국정보보호진흥원 연구원
관심분야 : 시스템 보안, 운영체제, 인공지능
E-mail : esoh@cert.certcc.or.kr