情報保護學會論文誌
第15卷 第6號, 2005. 12

MS 윈도우즈에서 E-메일 웜-바이러스 차단 시스템의 설계 및 구현*

최 종 천,† 장 혜 영, 조 성 제‡
단국대학교

Design and Implementation of an E-mail Worm-Virus Filtering System on MS Windows*

Jong-Cheon Choi,† Hye-Young Chang, Seong-Je Cho‡
Dankook University

요 약

최근 널리 유포되고 있는 악의적인 프로그램으로 e-메일 웜-바이러스가 있다. 이들은 웜-바이러스가 포함된 e-메일이나 첨부파일을 열기 만해도 메일 주소록에 등록된 모든 사용자에게 자신을 전파하여 막대한 피해를 유발시킨다. 본 논문에서는 MS 윈도우즈 시스템에서 e-메일 웜-바이러스의 전파를 차단시켜 주는 두 가지 방법을 제안한다. 첫 번째 방법은 메일 클라이언트에서 <보내기> 버튼의 클릭과 같은 송신자 행위에 의해서만 메일을 발송하게 하였다. 두 번째 방법에서는 제안한 시스템이 정한 규칙에 따라 수신자 메일주소를 변형하는 모듈과 복원하는 모듈이 각각 송신 측에 추가되었다. 또한 새로운 e-메일 웜-바이러스 공격에 대응하기 위한 방법으로 다형성 기법도 적용한다. 두 방법 모두 e-메일 웜-바이러스에 의해 자동으로 메일이 전송되는 것을 차단하도록 설계되었으며, e-메일 수신 측에서는 추가로 실행하는 작업은 없다. 실험을 통해, 제안한 방법들이 적은 오버헤드로 e-메일 웜-바이러스를 효과적으로 차단함을 보였다.

ABSTRACT

Recently, the malicious e-mail worm-viruses have been widely spreaded over the Internet. If the recipient opens the e-mail attachment or an e-mail itself that contains the worm-virus, the worm-virus can be activated and then cause a tremendous damage to the system by propagating itself to everyone on the mailing list in the user's e-mail package. In this paper, we have designed and implemented two methods blocking e-mail worm-viruses. In the first method, each e-mail is transmitted only by sender activity such as the click of <send> button on a mail client application. In the second one, we insert the two modules into the sender side, where the one module transforms a recipient's address depending on a predefined rule only in time of pushing <send> button and the other converts the address reversely with the former module whenever an e-mail is sent. The latter method also supports a polymorphism model in order to cope with the new types of e-mail worm-virus attacks. The two methods are designed not to work for the e-mail viruses. There is no additional function on the receiver's side of the e-mail system. Experimental results show that the proposed methods can screen the e-mail worm-viruses efficiently with a low overhead.

Keywords : *e-메일 웜-바이러스(e-mail worm-virus), 사용자 행위(user activity), 다형성(polymorphism), 변형 모듈(transformation module), 복원 모듈(restoration module)*

Ⅰ. 서 론

최근 바이러스는 유무선 네트워크 기술의 발달과 인터넷 확대에 힘입어 다양한 형태로 급속히 유포되고 있다. 1999년 3월 26일 처음 등장한 Melissa

접수일: 2005년 7월 29일; 채택일: 2005년 12월 5일
* 본 연구는 2004학년도 단국대학교 대학연구비의 지원으로 연구되었습니다.
† 주저자 : godofslp@dankook.ac.kr
‡ 교신저자 : sjcho@dankook.ac.kr

웜-바이러스, 2000년에 널리 유포된 Love Letter 웜-바이러스, 최근에 출현한 Sobig, NETSKY worm 등의 e-메일 웜-바이러스(e-mail worm-virus)[1]는 VBS(Visual Basic Script) 또는 Win32 응용으로 작성되어 유포되었으며[1,2,3], 수신자가 웜-바이러스가 내장된 e-메일 첨부파일이나 e-메일 자체를 열기 만해도 활성화되어 실행된다. E-메일 웜-바이러스는 실행되면서 메일링 리스트에 등록되어 있는 모든 사용자에게 자신을 급속히 확산시킴으로써 단기간에 큰 피해를 유발시킨다. Love letter 웜-바이러스의 경우, 5시간 만에 150억 달러 정도의 막대한 피해를 입힌 적이 있다[1]. 이러한 e-메일 웜-바이러스에 대처할 수 있는 상용 백신 프로그램이나 메일 필터링 기술들이 있지만, 이들은 해당 웜-바이러스에 대한 정보나 패턴(signature)을 미리 알고 있을 때에만 효력을 발휘하며 패턴이 알려지지 않은 새로운 웜-바이러스에 대하여는 적절히 대처할 수 없다는 단점이 있다. MS 윈도우즈 상에서 VBS로 작성된 웜·바이러스가 실행되지 못하게 스크립팅(scripting)을 비활성화 시키는(disabling) 방법도 있지만, 이 방법은 정상적인 VBS도 실행되지 못하게 하는 불편을 줄 수도 있다.

본 논문에서는 MS 윈도우즈 시스템에서 e-메일 웜-바이러스에 의한 e-메일 전송을 효과적으로 차단하기 위한 두 가지 시스템, 즉 "송신자 발송 시스템"과 "메일주소 변형 및 복원 시스템"을 제안한다. 송신자 발송 시스템은 송신자의 〈보내기〉 버튼 클릭과 같은 외부 이벤트 발생 여부를 감지하여, 송신자가 직접 전송하는 e-메일만 정상 발송하고, 외부 이벤트와 무관하게 e-메일 웜-바이러스가 자동으로 e-메일을 전송하는 경우에는 e-메일을 차단한다. 영리한 e-메일 웜-바이러스의 경우 〈보내기〉 버튼 클릭과 동일한 외부 이벤트를 발생하여 e-메일을 발송함으로써, 송신자 발송 시스템을 무력화시킬지도 모른다. 이를 대비하기 위해 메일주소 변형 및 복원 시스템을 제안한다. 메일주소 변형 및 복원 시스템은 송신자 발송 시스템에 추가적으로 "변형 모듈"(transformation module) 및 "복원 모듈"(restoration module)을 송신 측에 구축하여, 웜-바이러스에 의한 e-메일 전송을 차단한다. "변형 모듈"은 송신 측의 메일클라이언트에 구현되어 송신자가 〈보내기〉 버튼을 누를 때만 수신자 메일주소를 미리 정한 규칙에 따라 변형한 후 송신 측 메일서버에 전달한다. "복원 모듈"은 송신 측의 메일서버에 구현되어, 클라이언트로부터 전달된 메일주소를 원래의 수신자 주소로 복원한 후, 인터넷을 통해 수신 측에 전송한다. "변형 모듈"은 사용자의 동작에 의해서만 실행되며, "복원 모듈"은 모든 e-메일 전송 시마다 자동적으로 수행된다. 또한 제안 기법이 새로운 e-메일 웜-바이러스에 의해 공격받을 경우를 대비하여 다형성(polymorphism) 모델로 한층 더 보완한다.

본 논문의 구성은 다음과 같다. 2장에서 e-메일 전송시스템, e-메일 웜-바이러스의 현황 및 기존 대처 방안의 단점에 대해 기술하며, 3장에서는 제안하고자 하는 시스템의 전체 구성과 설계 방안에 대해 설명한다. 4장에서는 실제 프로토타입 시스템 구현 환경 및 구현 과정에 대해 기술하고, 5장에서 구현된 시스템을 테스트하고 성능을 평가한다. 6장에서 결론 및 향후 과제로 끝을 맺는다.

II. 관련 연구 및 현황

본 장에서는 일반적인 e-메일 전송 시스템의 전체적인 구조, e-메일 웜-바이러스의 현황 및 피해 규모, 방지 방법 등에 대해 간략히 기술한다.

1. E-메일 전송 시스템

E-메일은 인터넷 응용 중의 하나로, 메일을 보낼 때와 받을 때 사용되는 일반적인 프로토콜은 각각 SMTP(Simple Mail Transfer Protocol)[4]와 POP3(Post Office Protocol Version 3)[5]이다. 일반적으로 e-메일 전송 시스템은 그림 1과 같이 MUA(Mail User Agent) 와 MTA(Mail Transfer Agent)로 구성된다. MUA는 e-메일 클라이언트 응용에 해당되며, MTA는 메일 서버에 해당된다.

사용자는 MUA로 e-메일을 작성한 다음 SMTP를 이용하여 송신측 MTA에 전송한다. 메일을 전달받은 송신측 MTA는 수신인을 확인하여 어느 MTA로 전송할지를 결정한다. 대부분의 경우 메일을 수신자의 사서함(mailbox)이 있는 수신 측 MTA로 직접 전송하며, 때에 따라서는 다른 MTA를 경유해 전송하기도 한다. 수신인의 사서함에 저장된 메일은 수신인이 자신의 MUA를 이용하여 사서함을 열어볼 때 최종적으로 배달된다. 수신측 MUA가 사서함

1) 현재는 바이러스와 웜의 경계가 모호해 졌으며, 두 가지 특성이 서로 결합된 악성 프로그램인 웜·바이러스가 작성되어 유포되기도 한다.

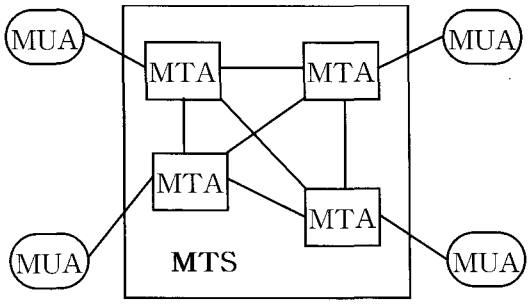


그림 1. 메일 전송 시스템

으로부터 메일을 받기 위한 프로토콜로 POP3를 사용한다. 본 논문에서는 그림 1과 같은 메일 전송 시스템 상에서 송신측이 e-메일 웜-바이러스를 차단하는 경우를 다루며, 현재 많은 메일 시스템이 그림 1과 같은 구조로 되어 있다.

2. MS 윈도우즈 e-메일 전송 시스템

MS 윈도우즈에서 e-메일 시스템은 그림 2의 구조를 갖는 MAPI (Messaging API)를 이용하여 구현된다[6]. MAPI는 메일 클라이언트가 사용하는 프로그래밍 인터페이스로 OLE 컴포넌트 객체 모델 형태로 운영체제 상위에 구축되어 있다. MAPI는 클라이언트 응용이 사용할 수 있는 Simple MAPI, CMC(Common Messaging Calls), CDO (Collaboration Data Objects) 라이브러리 등 세 개의 인터페이스를, 그 하위 단계에는 Service providers를 제공한다. MS 윈도우즈에서 주로 사

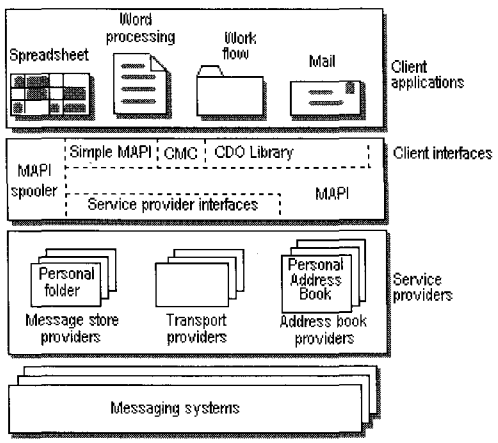


그림 2. MAPI 구조

표 1. E-메일 웜-바이러스 전파 시간 및 피해 규모

바이러스	발견 년도	유 형	전파 시간	추정 손해비용
Jerusalem, Cascade, Form	1990	.exe 파일	3 years	$50 million for all over five years
Concept	1995	Word macro	4 months	$50 million
Melissa	1999	E-mail enabled, Word macro	4 days	Up to $385 million
Love letter	2000	E-mail enabled, VBS based	5 hours	Up to $15 billion

용되는 Outlook express는 Simple MAPI 인터페이스를 이용하여 메일에 관련된 작업을 실행한다. e-메일 웜-바이러스도 이러한 MS 윈도우즈 메일 시스템을 이용하여 VBS나 실행파일 형식으로 MAPI를 통하여 Outlook의 주소록을 참조하고 메일을 전송하여 웜-바이러스를 전파시킨다.

3. E-메일 웜-바이러스 동향

E-메일 웜-바이러스는 웜-바이러스가 포함된 메일 첨부파일 또는 e-메일을 열 때 활성화된다. 활성화되면 특정한 타입의 파일을 찾아 자신의 복사본을 포함한 파일로 대체하는 작업을 수행하며, 공통적으로 Outlook express나 MS outlook을 이용하여 주소록에 등록된 모든 사람들에게 자신이 복제된 메일을 전송한다. 표 1에 대표적인 e-메일 웜-바이러스의 종류 및 이들 웜-바이러스에 의한 피해규모가 나타나 있다[1]. 표 1에 나타난 것처럼, 바이러스가 전파되는데 몇 년 또는 수개월이 걸렸던 과거에 비해 이제는 몇 시간 내에 바이러스가 급속히 전파될 수 있으며 그 피해 규모도 매우 크다. 최근에는 개별 e-메일 웜-바이러스의 피해규모를 측정하기도 쉽지 않을 정도로 많은 e-메일 웜-바이러스가 발생하고 소멸하고 있다.

KrCERT가 조사한 최근 국내 주요 바이러스 피해현황 및 동향분석[7]에 따르면 e-메일 웜-바이러스에 의한 피해가 여전히 매우 크며, 2002년 상반기부터 출현한 바이러스의 주요 특징 중 하나는, 빠른 전파력을 지닌 e-메일 웜-바이러스의 스팸 메일화이다. 초기의 e-메일 웜-바이러스는 비교적 간단한 형태의 제목, 본문, 첨부파일을 지닌 e-메일 웜-바이러스 형태였다. 그러나 클레즈웜 변종(Klez.H)과 같은 e-메일 웜-바이러스의 형태를 보면 매우 다양

한 제목과 본문, 첨부 파일명을 지니고 전파되어 일반사용자들이 실제 메일과 구별하기가 어렵고 메일 필터링 기능을 통한 예방에 한계가 있다. 한국정보보호진흥원의 '2005년 4월 해킹바이러스 통계 및 분석 월보'[2] 문서에 나타난 바이러스(악성프로그램) 현황을 보면, 웜·바이러스 형태가 전체 바이러스 감염의 절반이상을 차지하며 그중에서도 e-메일 웜-바이러스인 NETSKY는 총 피해건수가 2005년에만 4월 기준으로 3,414건에 이르러 2위인 BAGLE와 7배 이상의 차이를 보인다. 이처럼, 최근 바이러스들 중에서 e-메일 웜-바이러스에 의한 피해 규모가 가장 크며, 피해를 예방하는 것도 쉽지 않다.

4. E-메일 웜-바이러스 대처 방법

E-메일 웜-바이러스를 막는 가장 일반적인 해결책은 상용 백신(anti-virus) 제품들을 계속 갱신하거나, 바이러스 월(virus wall)과 같은 e-메일 필터링 기술을 사용하여 웜·바이러스를 포함한 메시지를 제거하거나 차단하는 것이다[3]. 제품에 따라 차이는 있지만, 현재 사용되는 백신 또는 필터링 프로그램들은 실시간으로 다양한 프로토콜(HTTP, FTP, SMTP, TELNET 등)을 감시하여 인터넷으로부터의 알려진 바이러스 유입을 차단한다. 하지만 이러한 제품들은 이미 발견된 바이러스들의 패턴들을 확보하고 검색 엔진을 통하여 바이러스의 존재 여부를 판단하여 대응하므로, 패턴이 알려지지 않은 새로운 바이러스에 대해서는 대처할 수 없다는 단점이 있다. 최근에는 이러한 단점을 보완하기 위한 방법으로 휴리스틱(Heuristic) 기능을 포함한 제품들이 연구 및 개발되고 있다[13]. 이러한 휴리스틱 기능은 악성코드에서 자주 사용하는 함수나 메소드 또는 시스템 호출에 대한 내용을 감시하다가 발견되는 변화를 통한 악성 행위를 판단하는 방법이다[14]. 또 다른 해결책은 VBS로 작성된 웜이 실행되지 못하게 WSH(Windows Scripting Host)를 비활성화 시키거나 인터넷 익스플로러의 Active scripting을 비활성화 시키는 것이다. 그러나 이 방법은 사용자가 필요로 하는 기능까지 비활성화 시킬 수 있으므로, 이의 적용에 대해서는 유의해야 한다[3].

Ⅲ. 시스템 구성

본 논문에서는 MS 윈도우즈 시스템 환경에서 사용자가 원하는 기능을 그대로 유지하면서, 새로운 e-메일 웜-바이러스도 차단할 수 있는 e-메일 송신 기법을 제안한다. 제안한 기법은 크게 두 가지 방법으로 설계·구현되었다. 구현된 첫 번째 시스템은 송신자 발송 시스템이며, 두 번째는 메일주소 변형 및 복원 시스템이다.

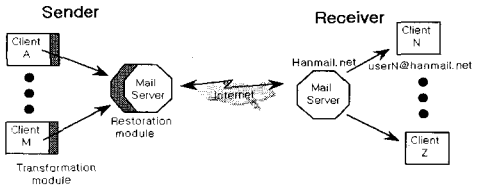


그림 3. 주소 변형 및 복원 시스템 구성도

1. 송신자 발송 시스템

송신자에 의해 요청된 메일만 정상적으로 발송하고, 송신자의 간섭 없이 e-메일 웜-바이러스에 의해 자동으로 메일이 발송되는 것을 차단하는 시스템이다. 정상적인 메일 전송의 경우는 Outlook express를 이용하여 송신자가 메일을 작성한 후 〈보내기〉 버튼을 클릭하게 된다. 이러한 점을 적용하여 메일 클라이언트가 메일을 보내기 위해 SMTP 서버에 접속을 시도할 때, 송신자의 〈보내기〉 버튼 클릭과 같은 외부 이벤트 발생 여부를 조사한다. 〈보내기〉 버튼이 눌러졌다면(즉, 외부 이벤트가 발생했다면) 정상적으로 메일을 발송하지만, 〈보내기〉 버튼이 눌려지지 않았는데 SMTP 서버로 접속 시도가 있다면 메일 전송을 거부함으로써 e-메일 웜-바이러스에 의한 자동 메일 전송을 차단한다. 이러한 방법은 매우 단순하면서도 강력하여 웜-바이러스에 의한 네트워크 트래픽의 증가도 예방할 수 있다.

2. 메일주소 변형 및 복원 시스템

영리한 e-메일 웜-바이러스는 〈보내기〉 버튼 클릭 이벤트를 흉내 내어 송신자 발송 시스템을 우회하여 웜-바이러스를 유포하게 될 지도 모른다. 따라서 웜-바이러스가 외부 이벤트 발생을 흉내 낼 가능성에 대비하여 송신자 발송 시스템의 개선책이 필요하다. 메일주소 변환 및 복원 시스템은 3.1절의 송신자 발송 시스템에 추가적으로 송신 측에만 '변형 모듈' 및 '복원 모듈'을 삽입한 것으로, 송신자 발송 시스템을

개선한 것이다. 이 시스템의 구성이 그림 3에 나타나 있다.

메일주소 변형 및 복원 시스템에서 송신 측 클라이언트에는 '변형 모듈'이, 송신 측 서버에는 '복원 모듈'이 새로이 추가되어 있다. 그림 3에서 두 모듈은 약한 음영으로 표시되어 있으며, 수신 메일주소만을 변형·복원한다. '변형 모듈'은 클라이언트가 e-메일을 보낼 때 〈보내기〉 버튼 클릭과 같은 사용자 행위에 의해 실행되어, 수신 메일주소를 정해진 규칙에 따라 변환한 후 송신 측 서버로 전달한다. '복원 모듈'은 클라이언트로부터 메일을 수신할 때마다 자동 실행되어 전달받은 메일주소를 정해진 규칙에 따라 변환한 다음 인터넷을 통해 수신 측에 전송한다. 복원 모듈이 실행하는 연산은 변형 모듈이 실행하는 연산의 역(reverse) 연산이다. '변형 모듈'은 e-메일 웜-바이러스에 의한 메일 송신 시에는 실행되지 않도록 설계되며, '복원 모듈'은 모든 메일 송신 시에 자동으로 실행되게 한다. 즉, 송신측 클라이언트의 변형 모듈은 사용자에 의해서만 실행되어 메일주소를 변형하여 전송하며, 송신측 서버의 복원 모듈은 전달받은 메일주소를 변형(또는 복원)하여 인터넷을 통해 수신자 측에 전송하게 된다. 이때 정상으로 송신되는 e-메일주소만이 변환 및 복원의 대상이 되며, 수신자 측에서는 추가로 수행할 일이 없다. 웜-바이러스에 의한 메일 전송의 경우, 변환과정이 없이 복원과정만 적용되므로, 수신 측으로 향하는 메일주소가 비정상 주소가 되어 메일 전송이 차단된다. 일반 사용자 입장에서는 기존 메일 전송 시스템에서와 동일한 인터페이스를 사용하므로 투명성이 제공된다.

2.1 다형성 모델

E-메일 주소를 변환하고 복원하는 모듈이 감추어져 구현된다하더라도 공격자가 유추하기 쉬운 방식으로 구현된다면, 새로운 e-메일 웜-바이러스에 의해 공격받을 가능성이 존재하게 된다. 따라서 웜-바이러스 공격에 대항할 수 있는 안전한 시스템을 구축하기 위해 변환 및 복원 모듈이 다양한 형태로 구현될 수 있어야 한다. 즉, 다형성 모델(polymorphous model)[2]을 지원하도록 구현된다. 예를 들어, 송신자 A가 메일주소가 *userN@mserver.net*인 수신자 *N*에게 e-메일을 보낸다고 가정하자. 이때 본 논문에서 *userN@mserver.net*를 간단히 줄여 *x*라고 표시하기로 한다. A가 메일을 작성한 후 〈보내기〉 버튼을 누르면 변형 모듈이 수행되어 *x*를 $y=f(x)$로 변환하여 복원 모듈로 전달된다. 복원 모듈은 $f(x)$를 수신하여 $f^{-1}(f(x)) = f^{-1}(y)$를 수행하여 *x*를 계산해 내어 인터넷으로 통해 메일을 전송하게 된다. 즉, 복원 모듈이 하는 일은 변환 모듈의 역함수이다. 다형성 모델에서, 변환 모듈이 수행하는 $f(x)$ 계산과 복원 모듈이 수행하는 $f^{-1}f(x)$ 계산에 대한 구체적인 예는 다음과 같다.

· 문자 첨가 및 제거
- 수신자 전자우편 주소의 임의 부분에 m개라는 특정한 수의 문자 삽입 및 제거
- 수신자 주소가 *userN@mserver.net*, $m=2$, 임의의 한 문자를 ?로 표현한다고 할 때,

① 변형 모듈은 다음의 예와 같이 수신자 주소의 제일 앞, 또는 @ 기호 앞이나 뒤, 또는 주소의 제일 뒤 등에 2개의 임의문자를 첨가한다. 이러한 조작은 단순하기 때문에 송신자의 수작업에 의해서도 가능하다.

예) *??userN@mserver.net*,
userN??@mserver.net,
userN@??mserver.net,
userN@mserver.net??

② 복원 모듈은 역으로 수신자 주소의 제일 앞, 또는 @ 기호 앞이나 뒤, 또는 주소의 제일 뒤 등에 첨가된 2개의 임의문자를 제거하여, 본래의 주소 *userN@mserver.net*를 얻는다.

· 압축 및 풀기
- 변형 모듈은 압축함수 *C*를 이용하여 *x*를 압축, 변형 모듈은 $C(x)$를 계산하여 복원 모듈에 전달
- 복원 모듈은 함수 *C*의 역함수 *D*를 이용하여 *x*를 복원, 복원 모듈은 $D(C(x))$를 계산하여 *x*를 복원

· 암호 및 복호
- 변형 모듈은 암호화 알고리즘 *E* 및 암호화 키 k_e를 이용하여 *x*를 암호화, 변형 모듈은 $E_{ke}(x)$를 계산하여 복원 모듈에 전달
- 복원 모듈은 복호화 알고리즘 *D* 및 복호화 키 k_d를 이용하여 *x*를 복원, 복원 모듈은 D_{kd}

2) 본래 바이러스 중에도 백신 프로그램으로부터 탐지를 피하기 위해 감염시마다 시그너쳐(signature) 패턴을 변형하는 다형성 바이러스가 있다. 이와 달리, 본 논문에서는 다형성이란 용어를 e-메일 웜-바이러스의 공격을 피하기 위한 한 방안으로 사용한다.

메일바이러스 방지 기법 적용	☑ 적용 ☐ 적용 않함	
변환 및 복원 방식	☐ 압축 및 풀기	(1)번 함수 (2)번 함수 (3)번 함수
	☑ 암호 및 복호	(1)번키 (2)번키 ③난수에 의한 키 생성
	☐ 문자 첨가 및 제거	○ 2문자○ 3문자 (1)제일 앞 (2)@ 앞 (3)@ 뒤 (4)제일 뒤
적용 기간	☐ 1일 ☑ 1주일 ☐ 한달	

그림 4. e-메일 웜-바이러스 차단을 위한 시스템 설정 화면 예

$(E_{ke}(x))$를 계산하여 x를 복원

- DES와 같은 대칭키(비밀키) 방식이 사용되면 k_e와 k_d는 동일 키, RSA와 같은 비대칭키(공개키) 방식이 사용되면 k_e와 k_d는 다른 키

사용자나 시스템 관리자로 하여금 위 세 가지 방식 중에 한 가지를 임의 선택하게 함으로써 다형성을 지원할 수 있으며, 주기적으로 변형 및 복원 방식을 변경함으로써 e-메일 웜-바이러스의 공격을 효과적으로 예방할 수 있다. 즉, 이러한 다형성 모델은 웜-바이러스로 하여금 주소 변환 규칙을 쉽게 파악할 수 없도록 하여 준다. 만약 웜-바이러스가 주소 변환 규칙을 파악한다하더라도 사용자마다 다른 변환 규칙을 사용한다면, 웜-바이러스의 파급효과는 미미해진다.

2.2 E-메일 웜-바이러스 차단 시스템의 설정 예

웜-바이러스에 의한 메일 전파를 차단하기 위한 송신측 클라이언트 시스템의 설정 화면 예가 그림 4에 나타나 있다. e-메일 웜-바이러스 차단 기법을 적용하겠다고 선택할 경우에는 수신자의 전자우편 주소를 변환하고 복원하는 방식과 해당 방식을 적용하는 기간을 선택해야 한다. 그림 4에서 변환 및 복원 방식으로는 크게 세 방식이 제시되어 있으며, 각 방식 별로 추가적인 선택 사항이 있어, e-메일 웜-바이러스가 변환 및 복원 방식을 유추할 수 어렵게 하였다. 또한 선택한 변환 및 복원 방식을 적용하는 기간도 짧게는 하루, 길게는 한 달로 설정하여, 만약 웜-바이러스가 현재의 변환 및 복원 방식을 추론하였다 하더라도 다음 번 적용 기간에는 의미가 없도록 하였다. 그림 4에서 선택된 방식 및 기간은 송신측 클라이언트의 변환모듈과 송신측 서버의 복원 모듈에 동시에 반영되도록 한다. 그림에서 추천하는 방식은 난수에 의해 키를 생성하는 대칭키 방식의 암호 및 복호화 방식으로 적용 기간은 1주일이다.

메일을 송신하는 사용자마다 선택 가능한 변환 및 복원 방식, 적용 기간이 다를 수 있으므로, 송신 측 메일 서버는 해당 사용자(클라이언트) 별로 각 사용자 메일 주소(계정, 사용자 ID)를 적절히 처리할 수 있도록 서로 통신하여 설정될 수 있게 한다. 만약, 모든 사용자 별로 메일주소를 변환하고 복원하는 방식이 복잡하다면, 간단히 서버가 주도적으로 변환 및 복원하는 방식, 그리고 적용 기간을 정해서, 초기화 시 또는 변경 시마다 각 사용자의 클라이언트가 실행될 때 통신하여 알려주는 방식도 고려할 수 있다.

설정 정보를 은닉하기 위한 한 방법으로, 설정 정보는 서버에 관리자만이 접근할 수 있는 특수 파일에 저장되어 관리된다. 메일 클라이언트는 실행될 때 서버와 통신하여 설정 정보를 알게 되는데, 클라이언트 측에서는 설정 정보를 위한 저장 장소가 임의 배정이 되게 하여 매번 달라지게 한다. 성능을 위해, 현재로서는 설정정보의 암·복호화는 고려하고 있지 않다.

Ⅳ. 구현

1. 구현 환경

E-메일 웜-바이러스 차단을 위한 메일주소 변환 및 복원 시스템의 전체 구성이 그림 5에 나타나 있다. 송신측 메일 클라이언트는 펜티엄 III 700MHz, MS 윈도우즈 2000, Outlook express 6.0 상에, 송신측 메일 서버는 펜티엄 III 800MHz, 와우 리눅스, Sendmail 8.11 상에 구축되었다. 즉, 변형 모듈은 MS 윈도우즈에 구현되었으며 복원 모듈은 리눅스 상에서 구현되었다. MS Outlook은 오피스를 설치해야 사용할 수 있는 반면, Outlook ex-

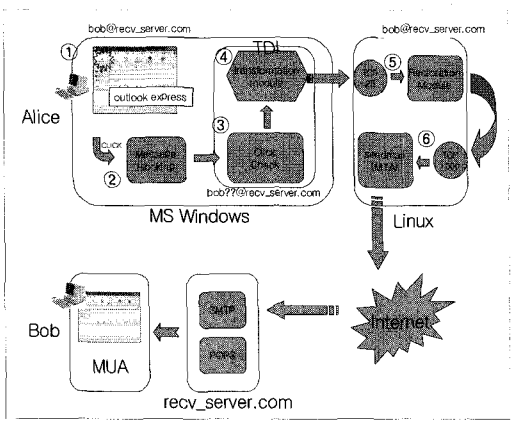


그림 5. 시스템 구성도

press는 MS 윈도우즈에 기본적으로 내장되어 있기 때문에 송신측 대상 응용으로 Outlook express를 선택하였다. 수신측의 시스템은 변경 없이 그대로 사용된다.

2. 사용자 행위 감지

MS 윈도우즈 운영체제는 메시지 구동 구조(message-driven architecture)를 갖는다. 여기서 메시지는 키보드를 입력, 마우스 이동 또는 클릭, 내부적인 운영체제 활동 등에 의해 발생되는 이벤트를 나타내며, 메시지 구동 구조란 메시지를 감지하여 해당 메시지에 해당하는 윈도우 프로시저를 호출하는 구조를 의미한다. MS 윈도우즈는 외부의 다양한 이벤트들을 감지하여 해당 응용에게 관련 메시지를 통지할 수 있다.

메일 송신자가 Outlook express에서 메일을 작성한 후 그림 5의 ①과 같이 〈보내기〉 버튼 클릭 이벤트를 발생하면, 이 이벤트 관련 메시지가 장치 드라이버에 전달되는데, 이때 메시지를 가로채는 것이 가능하다. 본 시스템은 전역 메시지 후킹(global message hooking)을 위한 DLL(Dynamic Link Library) 함수를 작성하여, 〈보내기〉 버튼 이벤트가 발생할 때 이에 대한 메시지를 가로채어 〈보내기〉 버튼이 눌려졌음을 알리는 *발송플래그*를 설정하고 메일 주소에 대한 변형을 수행한다(그림 5의 ①). 발송플래그의 초기 값은 0이며, 〈보내기〉 버튼 클릭 이벤트가 발생하면 1로 설정된다. 변형된 주소의 메일이 Outlook express에 전달되는데, 본 시스템에서는 Outlook express로 하여금 Simple MAPI 인터페이스와 연계된 TDI(Transport Driver Interface) 디바이스 드라이버에서 발송플래그를 검사함으로써 〈보내기〉 버튼 클릭이라는 사용자 행위를 확인할 수 있도록 구현하였다(그림 5의 ②). 즉, 전송할 메일이 있고 또한 발송플래그가 1이면 그 메일을 SMTP 서버로 전송한다(그림 5의 ③). 이때, 이벤트 큐를 관리(큐 길이 포함)하는 방식으로는 MS 윈도우즈 시스템에서 사용되는 고유 방식을 적용한다. 일부 영리한 공격자가 〈보내기〉 버튼이 눌려진 직후에 이어 붙여 웜-바이러스 전송 공격을 시도할 수 있다. 문제가 되는 경우는 웜-바이러스가 정상 메일의 첨부파일로 인식되는 경우이다. 만약, 웜-바이러스가 메시지 큐에 첨부된다 하더라도 SMTP 특성상 첨부파일이 존재하면 MIME (Multipurpose Internet Mail Extensions) 타입으로 변형되어야 하며 이러한 경우 다양한 형태의 인코딩이 필요하게 되며, 결국 원본 메일을 재작성하지 않고 추가적인 첨부 파일로 웜-바이러스를 전파하는 것은 매우 힘들다. 이러한 특성으로 인해, 메일 클라이언트가 아닌 메시지 큐 해킹을 통한 파일 첨부 공격은 아주 어렵다고 판단된다[15].

MS 윈도우즈의 TDI 구조가 그림 6에 나타나 있다. TDI는 트랜스포트 프로토콜 스택(transport protocol stack)의 상위 부에서 사용할 수 있는 커널 모드 네트워크 인터페이스로, 특정 용도에 맞는 트랜스포트 드라이버를 쉽게 개발할 수 있게 해준다[8]. 그림 6에서 TDI 하단에 TCP/IP 등의 프로토콜 계층이 있는데, 이 계층은 임의 특정 모듈을 삽입할 수 있도록 NDIS (Network Driver Interface Specification) 라이브러리 구조를 지원한다. NDIS는 여러 유형의 네트워크 드라이버를 지원하며, 계층적인 네트워크 드라이버들 사이에 표준 인터페이스를 기술함으로써 네트워크 트랜스포트와 같은 상위 수준 드라이버에 하위 수준 드라이버를 추상화시켜 준다. 이러한 구조를 이용하여 사용자가 필요로 하는 목적에 따라 원하는 계층에 특정 모듈을 삽입하는 것이 가능하다. 또한 이러한 구조에 의한 모듈 삽입은 계층간에 표준 전송방식에 따른 큐(Queue)를 사용하므로 별도의 자료구조에 대한 설정 없이 구현할 수 있다. 본 논문에서는 *Transport providers* 계층에 가상장치 드라이버(*Hook Drv*)를 삽입하여 필요한 메시지를 가로채게 하였다. 즉 〈보내기〉 버튼이 눌려 졌을 때 *Hook Drv*로 관련 이벤트가 통보되고, 이에 대한 처리를

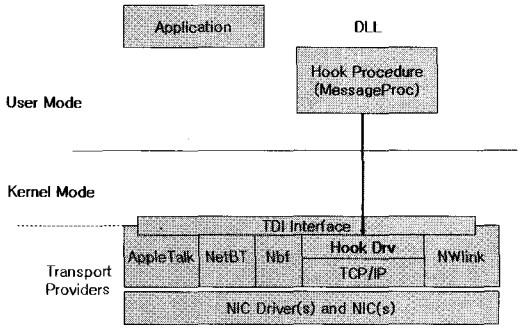


그림 6. TDI 구조

DLL로 작성된 *Hook procedure*가 담당하도록 구현하였다.

3. 송신자 발송 시스템의 구현

메일 전송 시 TCP 프로토콜을 필터링하는 장치 드라이버에서 메일 클라이언트 창의 〈보내기〉 버튼이 눌러졌는지를 조사하여, 눌러졌으면 SMTP 서버로의 접속을 허용하지만 그렇지 않은 경우 접속을 원천적으로 차단한다. 이 시스템이 구현된 구조는 그림 6과 같으며, 버튼 클릭 여부에 따라 SMTP 서버 접속이 TDI 드라이버 및 DLL에 의해 제한된다.

4. 주소변형 및 복원 시스템의 구현

본 논문에서는 3.2.1에서 제안한 다형성 모델 중 문자 추가 및 제거 기법에 대한 프로토타입 시스템을 구현하였다. 메일주소의 어느 부분에 문자를 삽입할 것인지는 그림 7의 사용자 인터페이스를 이용하여 송신자가 설정하도록 하였다. 그림에서 변형모듈은 수신자 메일주소의 특정 부분에 2개 문자를 삽입하고 복원모듈은 삽입된 2개의 문자를 제거하는 방식으로 구현되었다. 수신자 주소가 Account@Domain일 경우, 송신자가 그림 7과 같이 설정하였다면 메일전송 시 변형모듈은 주소를 Account??@Domain로 변형하여 자신의 SMTP 서버에 보낸다. 서버의 복원모듈은 주소의 @ 기호 앞에 첨가된 2개의 문자를 제거하여 본래의 주소 Account@Domain을 복원한 다음, 수신 측으로 전송한다. 그림 7의 'Sender E-mail Address' 칸에는 서버의 복원모듈이 복원한 정상 메일 주소가 표시되게 하여 송신자가 확인할 수 있게 하였다. 이러한 다형성 모

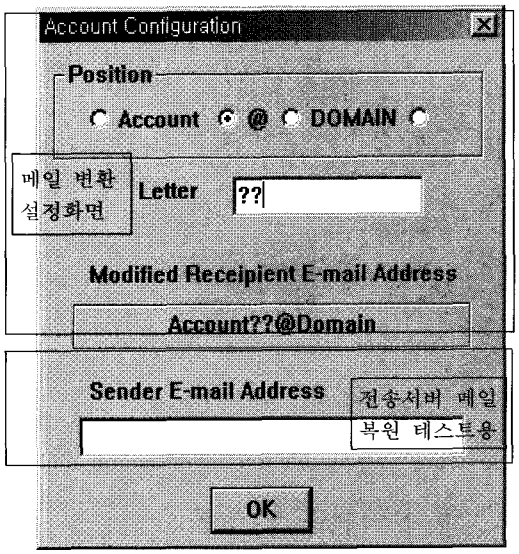


그림 7. 사용자 설정 화면의 예

델을 적용함으로서 웜-바이러스가 특정 패턴의 주소 변환을 감지하기 어렵게 만들어 변형모듈에 대한 안정성을 높였다. 주소 변환규칙은 서버의 복원모듈과 연동되어 구동되어야 하는데, 본 논문에서는 이러한 기능을 TCP 1000번 포트를 사용하여 구현하였다.

4.1 변형 모듈

송신자가 e-메일 작성 후 메일전송을 위해 Outlook express에서 〈보내기〉 버튼을 누를 때만 변환모듈이 실행되도록 하였다(그림 5의 ④). TDI의 메시지 가로채기 함수를 구현하여, 통지된 메시지가 〈보내기〉 버튼에 대한 것인지를 확인하여 *Hook* 드라이버에게 사용자 행위에 의해 송신되는 메일임을 알린다. 즉, TDI 드라이버는 메일 클라이언트에서 〈보내기〉 버튼이 눌러졌다면 수신자 메일주소를 변형하지만, 그렇지 않다면 그대로 통과시킨다.

4.2 복원 모듈

송신 측 SMTP 서버는 기본적으로 TCP 25번 포트를 이용하여 메일 클라이언트의 요청에 응답한다[4]. 즉, 메일전송 시 별도 지정이 없으면 클라이언트는 e-메일을 SMTP 서버의 25번 포트로 전달한다. 제안한 방법을 구현하기 위해, 그림 8와 같이 SMTP 서버 에 대응된 포트번호를 TCP 1000번으로 변경하고, TCP 25번 포트 부분에는 복원모듈을 설정하여 복원모듈로 하여금 메일 클라이언트의 요

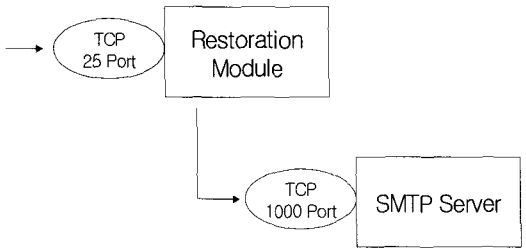


그림 8. 복원모듈의 구조

청을 받아들여 수신자의 주소를 정한 규칙에 따라 변환(또는 복구)하게 하였다(그림 5의 ⑤). 그 다음, 그림 5의 ⑥에서와 같이 전송하고자 하는 메일을 1000번 포트를 이용하여 SMTP 서버로 중계하였다. 이러한 작업은 리눅스 운영체제에서 Sendmail에 관련된 환경설정 파일(sendmail.cf)을 수정하여 구현하였다[9].

V. 실험 및 평가

프로토타입을 구현한 다음, 제안한 방법이 e-메일 웜-바이러스를 차단할 수 있는지 확인하기 위해 간단한 메일 자동발송 프로그램을 작성하여 실험하였다. 그림 9에 송신자 발송 시스템의 실행결과 화면이 나타나 있으며, 주소 변환 및 복원 시스템의 실행결과도 동일하다. 그림에서 "*net start tdi_ft*"는 TDI *Hook* 드라이버를 수동으로 활성화시키는 명령이며, "mail_test"는 MAPI를 이용하여 메일을 자동으로 발송하는 프로그램으로 e-메일 웜-바이러스와 동작 방식이 동일하도록 작성되었다. e-메일 웜-바이러스는 주로 VBS 또는 실행파일을 통하여 주소록을 바탕으로 메일을 전송하며, VBS와 실행파일은 기본적으로 MAPI를 호출하여 원하는 목적을 달성한다[10]. 그림 9를 보면 "mail_test"에 의해 자동 발송되는 e-메일이 차단됨을 확인할 수 있다. 여기서 주목할 만한 점은 '송신자 발송 시스템'에 의해 메일 웜-바이러스가 차단될 경우에는 구현 및 네트워크 트래픽 면에서 가장 효과적이며, '메일주소 변환 및 복원 시스템'에 의해 웜-바이러스가 차단될 경우에는 설정한 규칙에 의존하여 비정상 메일주소로 메일이 전송 후 되돌아 올 수도 있으므로 네트워크 트래픽 면에서 크게 효과적이지는 않을 수 있다는 것이다.

다음으로 구현된 시스템의 성능을 측정하여 보았다. 송신자 발송 시스템의 경우에는 성능 저하(메일 전송 지연)가 거의 없었으나, 메일주소 변환 및 복원 시스템의 경우는 일부 성능저하가 있었다. 후자 시스템의 성능평가는 변형모듈과 복원모듈이 추가됨으로써 메일전송 시 발생되는 지연시간을 측정하여 수행하였다. 성능평가 시 외부상황에 의해 발생할 수 있는 지연 요소를 제거하기 위해 메일 클라이언트와 서버를 로컬 네트워크로 접속하였으며, 서버에서는 네트워크 모니터링 도구인 "*tcpdump*" 명령을 사용하여 각 패킷이 캡쳐되는 시간을 측정하였다. 또한 메일을 외부 네트워크 서버로 중계하는 경우에는 그에 따른 지연시간을 측정하는 것이 어렵기 때문에, 메일 수신자가 복원모듈이 설치되어 있는 서버에 있는 상황을 가정하였으며, 모든 메일 사용자가 동일한 주소 변환 규칙을 설정하였다고 가정하여 메일이 전송되는 시간 지연을 측정하였다. E-메일의 크기를 변경해 가면서 메일 전송 지연시간을 측정한 결과가 표 2에 나타나 있다. 표의 지연시간은 각 메일 크기에 대해 전송시간을 5번씩 측정하여 평균한 값이며, 제안기법으로 인한 전송지연 시간은 약 1.4~1.6ms이다. 결과를 분석해 보면 메일 크기는 전송 시간에 큰 영향을 미치지 않으며, 송신 측 SMTP 서버에서 메일을 25번 포트로 전달받아 주

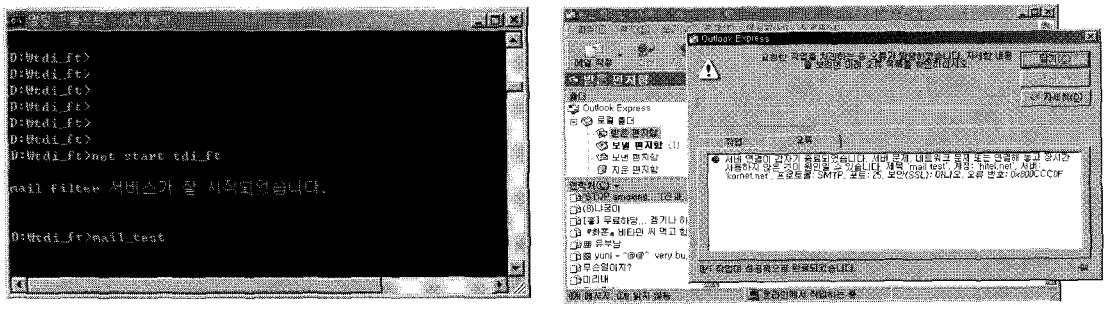


그림 9. 실험 결과 화면

표 2. 변형 및 복원 시스템에서 평균 메일전송 지연시간

메일 크기	5 바이트	50 바이트	500 바이트	5K 바이트
기존 시스템	0.064 ms	0.140 ms	0.142 ms	0.148 ms
제안한 시스템	1.526 ms	1.568 ms	1.716 ms	1.770 ms

소 복원 후 1000번 포트로 릴레이 하는 과정에서 일부 지연시간이 유발된다. 이는 다형성 모델 및 새로운 e-메일 웜-바이러스에 대비하기 위한 것으로 보안을 강화하기 위해서는 피할 수 없는 오버헤드라고 생각된다.

Ⅵ. 결론 및 향후 과제

본 논문에서는 e-메일 웜-바이러스의 전파를 차단할 수 있는 두 가지 방법을 설계하고 구현하였다. 첫 번째 방법은 송신자 발송 시스템으로, 메일 송신자가 〈보내기〉 버튼을 누를 때만 메일이 발송되는 시스템으로 사용자 행위 없이 자동으로 발송되는 메일을 차단한다. 두 번째는 메일주소 변형 및 복원 시스템으로 송신자 발송 시스템을 개선한 방법이다. 이 방법에서 메일 클라이언트는 사용자의 행위에 의해서만 수행되는 변형모듈을 이용하여 수신자 메일 주소를 변환해서 송신 측 서버로 전송하며, 송신 측 서버는 메일전송 시 항상 수행되는 복원모듈을 이용하여 전송 받은 메일의 주소를 정한 규칙에 따라 변환(또는 복구)시킨 다음 수신자 측에게 전달하게 된다. 또한 주소변환 규칙에 다형성 모델을 지원하여 새로운 웜-바이러스 공격도 감내하게 하였으며, 실험을 통해 유용성을 보였다. 향후, 다형성 정보를 은닉하는 방법을 개선하고, 최근 SMTP 서비스에서 도입되고 있는 SSL(Secure Sockets Layer)에 대해 고려하는 것이 필요하다. 송신자 발송 시스템은 SSL 기반에서도 그대로 적용될 수 있으나, 메일 주소 변형 및 복원 시스템은 일부 보완이 필요하다.

현재, 송신 측 메일서버의 변경 없이 클라이언트의 변경만으로 e-메일 웜-바이러스를 예방하고 차단하는 기법 및 변환에 대한 서버와 클라이언트의 약속 부분의 보안 메커니즘에 대해 연구할 계획이다. 또한 제안한 기법을 확장하여 로봇 에이전트가 전자게시판에 스팸 메시지를 자동으로 게시하는 것을 방지하는 방법을 연구하고자 한다.

참 고 문 헌

[1] William Stallings, *"Operating Systems"*, 4th Edition, Chapter 15, Prentice Hall, 2000

[2] KrCERT, 인터넷 침해사고 동향 및 분석 월보, 한국정보보호진흥원, 2005. 5

[3] http://www.cert.org/advisories/CA-2000-04.html

[4] Jonathan B. Postel, "Simple Mail Transfer Protocol", RFC 821, 1982

[5] M. Rose, "Post Office Protocol - Version 3", RFC 1081, 1988

[6] "MS Windows Plaform SDK", Document, Microsoft, 2002

[7] http://www.krcert.or.kr/

[8] "MS Windows DDK Document", Microsoft, 2001

[9] Eric Allman, "Sendmail Installation and Operation Guide", No. 8, Sendmail. Inc, 2001

[10] Roger A. Grimes, *"Malicious Mobile Code"*, O'reilly, 2001

[11] David Wood, *"Programming Internet Email"*, O'reilly, 1999

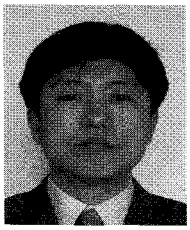
[12] 이현우, 백원민, 하도윤, 김상철, "메일필터링을 통한 E-mail 보안", 한국정보보호진흥원, 2001

[13] http://info.ahnlab.com/securityinfo/info_view.jsp?seq=5968&category=02

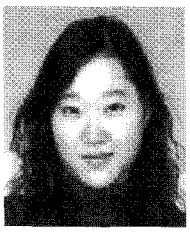
[14] 이성욱, "정적 분석과 코드 변환을 이용한 적극적인 악성 스크립트 대응", 아주대학교 박사학위논문, 2002

[15] N. Borenstein, Bellcore, N. Freed, Innosoft, RFC1521-MIME(Multipurpose Internet Mail Extensions), IETF, 1993

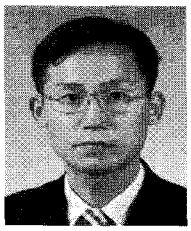
〈著者紹介〉



최 종 천 (Jong-cheon Choi) 학생회원
1995년 강릉대학교 통계학과 졸업 (학사)
2005년 단국대학교 컴퓨터과학 이학석사
2005년 3월~현재: 단국대학교 컴퓨터과학 박사과정
〈관심분야〉 컴퓨터 보안, DRM, 임베디드 시스템, 센서 네트워크



장 혜 영 (Hye-young Chang) 학생회원
2003년 단국대학교 전산통계학과 졸업 (학사)
2005년 단국대학교 컴퓨터과학 이학석사
2005년 3월~현재: 단국대학교 컴퓨터과학 박사과정
〈관심분야〉 정보보호, DRM, SW 취약점 분석, RFID



조 성 제 (Seong-je Cho) 정회원
1989년 서울대학교 컴퓨터공학과 졸업 (학사)
1991년 서울대학교 컴퓨터공학과 공학석사
1996년 서울대학교 컴퓨터공학과 공학박사
1996년~1997년 서울대학교 컴퓨터신기술연구소 연구원
2001년~2002년 미국 University of California, Irvine 객원연구원
1997년 3월~현재: 단국대학교 정보컴퓨터학부 부교수
〈관심분야〉 컴퓨터 보안, 시스템 소프트웨어, 실시간 시스템, 임베디드 시스템 등