효율적인 E-메일 바이러스 차단 기법

장혜영[0], 남강운, 최종천, 조성제, 우진운
단국대 정보컴퓨터학부
cheni199@hanmail.net[0], skywonny@hitel.net, godofslp@freechal.com ,sjcho@dankook.ac.kr, jwwoo@dankook.ac.kr

An Efficient Method for Blocking E-mail Virus

Hye-Young Chang[0], Kangwoon Nham, Jongchen Choi, Seongje Cho, Jinwoon Woo
Division of Information and Computer Science, Dankook University

요 약

최근 널리 유포되고 있는 악의적인 프로그램으로 E-메일 바이러스가 있는데, 이들은 바이러스가 포함된 E-메일이나 첨부파일을 열기 만 해도 메일 주소록에 등록된 모든 사용자에게 자신을 전파하여 막대한 피해를 유발시킨다. 본 논문에서는 메일주소 변환 및 복원 방식을 이용하여 E-메일 바이러스의 전파를 차단시켜주는 방법을 제안한다. 또한 다형성의 기법을 사용하여 새로운 E-메일 바이러스 공격에도 대응하였다. 이 방법들은 E-메일 바이러스에 의해서는 작동되지 않도록 설계되었으며, E-메일 서버쪽이나 수신자 측에서는 추가로 하는 작업이 전혀 없다.

1. 서론

E-메일 바이러스는 VBS(Visual Basic Script)또는 Win32응용으로 작성되며, 바이러스가 내장된 E-메일 패키지의 메일링 리스트에 있는 모든 사용자에게 자신을 급속히 확산시킴으로써 단기간에 큰 피해를 유발한다. E-메일 바이러스에 대처할 수 있는 상용 백신 프로그램이나 메일 필터링 기술들이 있지만, 이들은 해당 바이러스에 대한 정보나 패턴(signature)을 미리 알고 있을 때에만 효력을 발휘하며 새로운 바이러스(패턴이 알려지지 않은 바이러스)에 대해서는 적절히 대처할 수 없다는 단점이 있다.

본 논문에서는 E-메일 바이러스를 간단하고 효과적으로 차단하기 위해 메일 주소 변형 및 복원 시스템을 제안한다. E-메일 바이러스에게 감추어진 "변형 모듈"(transformation module) 및 "복원 모듈"(restoration module)을 송신자 시스템에서 구축하여 바이러스는 E-메일을 전송할 수 없게 하고 일반 사용자는 정상적으로 메일을 전송할 수 있게 한다.

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

2. 관련 연구 및 현황

2.1 E-메일 바이러스

E-메일 바이러스는 바이러스가 포함된 메일 첨부파일 또는 E-메일을 열 때 활성화된다. 활성화되면 특정한 타입의 파일을 찾아 자신의 복사본을 포함한 파일로 대체하는 작업을 수행하며, 공통적으로 아웃룩익스프레스나 MS아웃룩을 이용하여 주소록에 등록된 모든 사람들에게 자신이 복제된 메일을 전송한다. 2002년 상반기에 출현한 바이러스의 주요 특징 중 하나는, 빠른 전파력을 지닌 E-메일 바이러스의 스팸메일화이다. 클레즈웜 변종(Klez.H)과 같은 E-메일 바이러스의 형태를 보면 매우 다양한 제목과 본문, 첨부 파일명을 지니고 전파되어 일반사용자들이 실제 메일과 구별하기가 어렵고 메일 필터링 기능을 통한 예방에 한계가 있다.

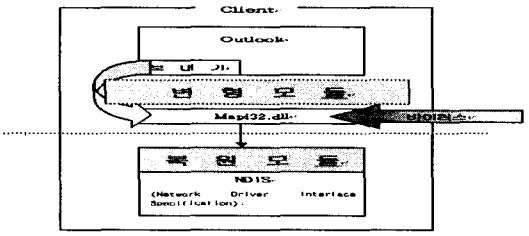
2.2 E-메일 바이러스 대처 방법

E-메일 바이러스를 막는 가장 일반적인 해결책은 상용 백신(anti-virus)제품들을 계속 갱신하거나, E-메일 필터링 기술을 사용하여 웜바이러스를 포함한 메시지를 제거하거나 차단하는 것이다[10]. 제품에 따라 차이는 있지만, 현재 사용되는 백신 또는 필터링 프로그램들은 실시간으로 다양한 프로토콜을 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. 또 다른 해결책은 VBS로 작성된 웜이 실행되지 못하게 WSH(Windows Scripting Host)를

불능화시키거나 인터넷 익스플로러의 Active scripting을 불능화시키는 것이다. 그러나 이 방법은 사용자가 필요로 하는 기능까지 불능화시킬 수 있으므로, 여러 적용에 대해서는 유의해야 한다.[10].

3. 시스템 구성

3.1 메일 주소 변환 및 복원 시스템



(그림 1) 주소 변형 및 복원

시스템의 구성은 (그림 1)에 나타나 있다. 기존 메일 전송 시스템의 송신자측 User Mode에 "변형 모듈"이, Kernel Mode에 "복원 모듈"이 새로이 추가되었다. 변형 모듈은 클라이언트가 수신자에게 E-메일을 보내기 직전에 <보내기>버튼 클릭 등의 사용자 행위에 의해 수행되어 수신자의 메일 주소를 정해진 규칙에 따라 새로운 형태로 변환한 후 MAPI(Messaging Application Program Interface)로 전달한다. 복원 모듈은 사용자 행위가 발생하거나 하지 않거나 상관없이 메일이 보내질 때마다 무조건 실행이 된다. E-메일 바이러스는 대부분 주소록에 등록된 모든 사람들에게 자신이 복제된 메일을 전송하는데 이때 바이러스는 MAPI로 바로 가서 실행하기 때문에 변환된 과정없이 복원만 된다. 복원만 실행된 E-메일 주소는 E-메일 주소의 형식을 벗어나게 되어 더 이상의 E-메일 바이러스 전파를 막을 수 있다. 일반 사용자 입장에서는 기존 메일 전송 시스템에서와 동일한 인터페이스를 사용하므로 투명성이 제공된다.

3.2 다형성 모델

E-메일 주소를 변환하고 복원하는 모듈이 감추어져 있다하더라도 유추하기 쉬운 방식으로 구현된다면, 새로운 E-메일 바이러스에 의해 공격받을 가능성이 존재하게 된다. 따라서 바이러스 공격에 대항할 수 있는 안전한 시스템을 구축하기 위해 변환 및 복원 모듈이 다양한 형태로 구현될 수 있어야 한다. 설명을 위해, 클라이언트 A가 메일주소가 userN@test.net인 클라이언트 N에게 E-메일을 보낸다고 가정하자. 또한 본 논문에서 userN@test.net는 간단히 줄여 x라고 표시하기도 한다. A가 전자우편을 작성한 후 <보내기> 버튼을 누르면 변형 모듈이 수행되어 x를 $y=f(x)$로 변환하여 복원 모듈로 전달된다. 복원 모듈은 $f(x)$를 수신하여 $f^{-1}f(x) = f^{-1}(y)$를 수행하여 x를 계산해 내어 인터넷으로 통해 그 전자우편을 전송하게 된다. 즉, 복원 모듈이 하는 일은 변환 모듈의 역함수이다. 변환 모듈이 수행하는 $f(x)$ 계산 및 복원 모듈이 수행하는 $f^{-1}f(x)$ 계산에 대한 구체적인 예는 다음과 같다.

- 수신자 전자우편 주소의 @를 임의의 특수문자로 대체한다. 예를들면 !, +, W 등의 E-메일 주소에서 사용 못하는 특수문자로 대체한다.
- 수신자 주소가 userN@test.net, !는 임의의 특수 문자를 표현한다고 할 때,

① 변형 모듈은 @를 대신하여 !로 대체하여 userN!test.net의 주소를 가지게 된다.

② 복원 모듈은 역으로 수신자 주소의 !를 @로바꾸어 원래의 주소 userN@test.net으로 전송되어 진다.

③ 변형 모듈을 거치지 않고 복원 모듈만 거쳤을 때는 주소에 @기호가 들어가 있게 되는데 복원 모듈에서는 @기호를 빈칸(Space, 0x20)으로 바꿔주어 더 이상의 바이러스의 전파를 막는다.

4. 구현

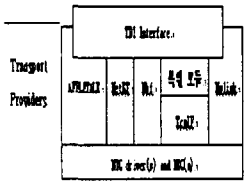
MS 아웃룩은 오피스를 설치해야 사용할 수 있는 반면, 아웃룩 익스프레스는 MS 윈도우와 같이 기본적으로 설치되기 때문에 송신측 대상 응용으로 아웃룩 익스프레스를 선택하였다.

4.1 변형 모듈

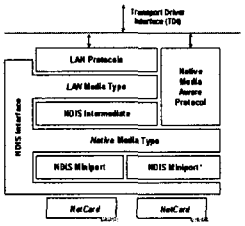
MS 윈도우 운영체제는 메시지 구동 구조(message-driven architecture)를 갖는다. 여기서 메시지는 키보드를 입력, 마우스 이동 또는 클릭, 내부적인 운영체제 활동 등에 의해 발생되는 이벤트를 나타내며, 메시지 구동 구조란 메시지를 검색하여 해당 메시지에 해당하는 윈도우 프로시저를 호출하는 구조를 의미한다. MS 윈도우는 외부의 다양한 이벤트들을 감지하여 해당 응용에게 관련 메시지를 통지할 수 있다. 본 논문에서는 윈도우 클래스 구조 중 메시지를 받아 처리하는 기능을 가지고 이벤트 callback을 처리하는 WNDCLASSA.lpfnWndProc라는 함수

포인터를 후킹한다. 후킹한 함수 안에서 <보내기>버튼을 눌렀을 때에만 그 포인터를 변환 모듈의 포인터로 삽입하고 그 외의 메시지는 원래의 함수로 복원하여 정상처리 되도록 한다.

4.2 복원 모듈



(그림 2) TDI구조

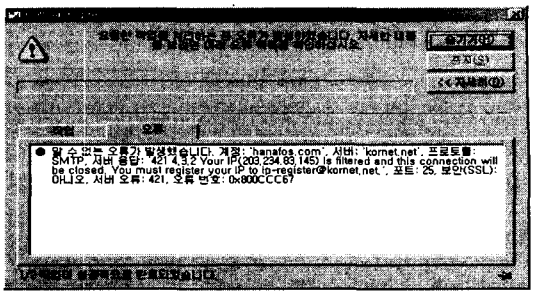


(그림 3) NDIS Architecture

MS 윈도우는 TDI 구조가 (그림 2)에 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[7]. (그림 2)에서 TDI 하단에 TCP/IP 등의 프로토콜 계층이 있는데, 이들 계층에 특정 모듈을 삽입할 수 있도록 NDIS(Network Driver Interface Specification) 라이브러리 구조도 지원된다. (그림 3)에 나타나 있는 NDIS는 여러 유형의 네트워크 드라이버를 지원하며, 계층적인 네트워크 드라이버들 사이에 표준 인터페이스를 기술함으로써 네트워크 트랜스포트와 같은 상위 수준 드라이버에 하위 수준 드라이버를 추상화시켜 준다. 따라서, 이들 구조를 이용하여 사용자가 필요로 하는 목적에 따라 원하는 계층에 특정 모듈을 삽입하는 것이 가능하다. 본 논문에서는 *Transport providers* 계층에 복원 모듈을 삽입하여 발송하는 E-메일의 주소를 무조건 복원하게 하였다.

5. 실험

MAPI를 이용하여 메일을 자동으로 발송하는 프로그램을 만들어 E-메일 바이러스와 동작 방식이 유사하도록 하여 실행시켰다. @기호가 공백으로 치환된 주소가 빠져나감으로써 제대로 된 E-메일 주소 형식을 갖추지 못하여 오류 메시지만 출력되었다. 정상적인 전송이 이루어지지 않음이 (그림 4)에 나타나 있다.



(그림 4) 복원만 실행됐을 때의 오류 메시지

6. 결론 및 향후 과제

본 논문에서는 E-메일 바이러스의 전파를 차단하기 위하여 간단하면서도 효율적인 메일주소 변형 및 복원 방법을 사용하였다. 사용자의 행위에 의해서만 수행되는 변형 모듈을 이용하여 수신자 메일주소를 변환해서 MAPI로 보내면 커널에 설치되어 있는 복원 모듈에서 본래의 주소로 복구시키게 된다. 이때 사용자의 행위없이 복원 모듈만 거쳤을 경우 인터넷에서 사용하는 주소의 형식을 갖추지 못하게 되어 E-메일을 전송하지 못하게 된다. 또한 주소 변환 규칙에 다형성 모델을 지원하여 새로운 바이러스공격에도 안전하다. 향후 악의적인 코드에 의한 전송인지 아닌지를 탐지하는 것에 대한 연구가 필요하다.

참고 문헌

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

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

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

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

[5] 이 현우, 백 원민, 하 도윤, 김 상철, "메일필터링을 통한 *E-mail* 보안", 한국정보보호진흥원, 2001

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

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

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

[9] http://www.certcc.or.kr/{statistics/virus/virus.htm, statistics/2003/0306_statistics.pdf}

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