특집

인터넷 웜과 바이러스의 진화와 전망

정 관 진 [1], 이 희 조 [2]

목 차

1. 서 론

오늘날의 IT 인프라는 과거와 비교하여 볼 때 그 규모가 몇 배 또는 몇 십 배 아니 그 이상으로 커졌으며 지금 이 시간에도 빠른 속도로 범위를 넓혀 나가고 있다. 이러한 IT 인프라 발전의 밑바탕에는 전세계를 하나로 묶은 인터넷이 있었으며, 많은 기업 및 개인들은 계속 이러한 구성원으로 참여해 가고 있다. 여기서의 '보안' 이라는 것은 그 중요성이 더해가고 있으며, 과거의 보안형태와 지금을 비교해 보면 큰 차이를 발견할 수 있다. 바로 현재 삶의 변화에 큰 영향을 주고 있는 IT 인프라의 발전이 역으로 보안과 개인 정보보호에 더 큰 위협요소가 되어버린 것이다. 시스템의 디스크를 파괴하는 바이러스부터 매크로바이러스 그리고 웜의 형태에 이르기까지 바이러스 또한 현재의 추세에 맞춰서 변화를 해 나가고 있는 것이다. 또한, 서비스거부공격(DoS)은 서버와 네트워크 전역에 장애를 유발하며 정상적인 서비스를 방해하고, 2001년의 마이크로소프트사의 IIS 웹 서버 소프트웨어의 취약점을 이용한 CodeRed 웜, 그리고 2003년 1월 10분내에 전세계 인터넷으로 확산이 된 SQL_Overflow (Slammer 또는 Sapphire등의 이름으로도 불리우며 본 고에서는 SQL_Overflow로 명명함) 등의 웜이 나타나 여러가지 간접효과를 일으키며 큰 피해를 유발하였다. 이번 SQL_Overflow 웜의 사례를 통해 악성코드들의 동향과 향후 전망에 대해 알아보도록 한다.

2. 악성코드의 동향

2.1 악성코드의 정의

악성코드는 크게 이론적 정의와 실질적 정의로 나누어 볼 수가 있다. 이론적 정의는 다른 사람에게 피해를 주기 위한 목적으로 제작된 모든 컴퓨터 프로그램 또는 실행 가능한 부분을 의미하며, 실질적 정의는 다른 사람에게 심리적, 실질적인 피해를 입히는 컴퓨터 프로그램 또는 실행 가능한 부분으로 제작자(사) 실수로 포함된 버그는 제외되나 광범위한 피해가 예상되는 경우를 포함하게 된다.

1) Apache.Kr.net 운영
2) 안철수연구소 기술기획실장

2.2 악성코드의 분류

2.2.1 컴퓨터바이러스

바이러스는 1984년 Fred Cohen의 학회 논문, "Computer Viruses - Theory and Experiments"에서 새로운 보안 위협으로 소개가 되었다. 바이러스는 감염 대상 프로그램(코드)에 자신의 코드 및 변형 코드를 감염시키며, 네트워크 및 컴퓨터 시스템에서 확산된다. 대표적으로 1988년 국내에 처음 발견된 Brain, 1998년 많은 피해를 준 CIH(Win95/CIH), 2000년 메일을 통해 널리 확산되었던 러브레터(VBS/Love Letter)등이 있다.

2.2.2 트로이목마

포괄적인 의미에서 악의의 기능을 포함하고 있는 프로그램이나 악의의 목적에 적극적으로 활용되는 프로그램 등을 의미한다. 국내의 인터넷 게임방과 같은 공유 PC환경에서 많은 피해가 우려되며, 대표적인 트로이목마로는 시스템을 원격에서 제어할 수 있는 백오리피스2000(Win-Trojan/Back Orifice 2000)과 넷버스(Win-Trojan/Netbus)등이 있다.

2.2.3 웜

기억장소에 코드형태로 존재하거나 혹은 실행파일로 존재하며 작동시 파일이나 코드자체가 다른 시스템으로 감염된다. 1988년 널리 퍼져 피해를 준 모리스웜을 시작으로 많은 웜들이 만들어져 큰 피해를 주고 있다. 대표적으로 1999년 국내외에 처음으로 윈도우기반의 인터넷 웜으로 발견 확산된 I-Worm/Happy99, 중요문서 유출이라는 치명적인 부작용을 가지며 널리 확산된 Win32/Sircam.wom 등이 있다.

2.2.4 혹스(Hoax)

혹스는 주로 이메일을 통하여 다른 사람에게 거짓 정보 즉, 루머를 유포하는 것으로 사용자에게 심리적인 위협이나 불안감을 조장한다.

2.2.5 조크

조크는 심리적인 위협이나 불안감을 조장하는 프로그램으로서 물질적인 피해는 없으나 백신에서 진단/삭제한다. 대표적으로 하드디스크를 포맷하는 화면을 보여주는 Win-Joke/Format Game이나 공포스런 얼굴을 작업 중에 갑자기 나타나게 하여 놀라게 하는 고스트(Win-Joke/Ghost) 등이 있다.

2.2.6 드로퍼

드로퍼 자체로는 감염 및 파괴활동을 하지 않지만, 실행되면 악성코드를 시스템에 설치하게 되며, 내부에는 악성코드의 실행파일, 설정파일, 라이브러리파일 등이 저장되어 있다.

컴퓨터 바이러스도 악성 코드의 일종이며, 〈표 1〉과 같이 컴퓨터 바이러스, 트로이목마 그리고 웜에 대한 비교를 통해 구분을 지어볼 수가 있다.

〈표 1〉 악성코드의 특성분류

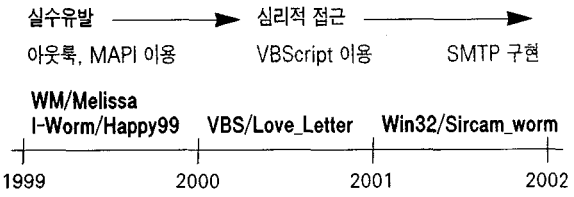
종류 \ 특성	자기복제	감염대상	형태	복구방법
컴퓨터 바이러스	O	O	기생/겹침	치료
트로이목마	X	X	독립	삭제
웜	O	X	독립	삭제

2.3 악성코드의 현황

최근의 악성코드의 동향을 보면 웜과 백도어가 결합된 형태나 네트워크 전파와 같이 복합적인 형태를 가진 바이러스가 많이 나타나고 있으며, 일반적으로 웜이나 트로이목마 제작에 고급언어를 사용하여 바이러스 보다 제작이 쉬운 것도 확산이 유중의 하나이다. 또한 구 종의 바이러스들이 여전히 상위권에 존재할 만큼 많이 활동하고 있으

며, VBScript 와 매크로바이러스는 급감되는 추세다. 기존의 악성코드 분류로 보면 과거의 바이러스는 점점 줄어들고 있으며, 웜과 트로이목마 증가가 뚜렷이 나타나고 있다. 이것을 증명하듯이 작년 하반기부터 Win32 기반의 트로이목마가 급증하고 있으며 악성 IRC 관련 트로이목마도 많이 발견되고 있다. 트로이목마의 형태가 파일삭제에서 정보유출 그리고 DoS(Denial Of Service) 또는 DDoS(Distributed Denial Of Service) 와 같이 다른 시스템을 공격하기 위한 도구로 활용되는 형태로 진화되어 가고 있다는 점이다. 더불어 자가복제기능을 가지고 전파가 되는 웜은 전파속도가 고속화 되고 있으며, 피해 규모가 광범해지고 있어 더욱 큰 위험을 주고 있다. 현재 웜의 변화를 보게 되면 다음과 같은 특징들이 나타나고 있다.

- 인스턴트 메시지(Instant Message), IRC 및 P2P웜 출현의 가속화
- FriendGreet, SysEntry와 같은 스팸(Spam) 메일러의 등장
- 주소록 이외의 파일에서 메일주소 추출기능을 가짐
- 공유폴더를 이용한 웜 증가
- 메일, 네트워크, 공유폴더와 같이 1개 이상의 전파경로사용
- 보안관련 소프트웨어 프로세스의 강제종료



(그림 1) 악성코드의 변화 추세

(그림 1)과 같이 최근에 광범위한 확산을 보이는 악성코드는 심리적 접근 방법을 사용하고 있으며, 단순 피해에서 구체적인 정보유출로 발전해 가고 있다. 이렇게 악성코드의 동향을 보게 되면 바이러스로부터 네트워크를 통하여 스스로 전파되고 피해를 입히는 웜 형태로 변화해 나가고 있다. 이렇듯 웜과 트로이목마, 바이러스 등의 보안사고는 피해 사실이 외부에 공개되는 경우 개인 및 회사 신용의 저하나 사용자가 감염사실을 모르는 경우 장기간 피해가 지속된다는 점 등의 특성으로 인하여 내포된 잠재적 위험성이 더욱 증가되고 있다.

3. 인터넷 웜의 특징과 진화

3.1 인터넷 웜의 정의 및 특징

인터넷 웜(Internet Worm)이란 원격지에서 불특정 시스템의 취약성을 이용하여 자신을 복제한 후 다른 시스템으로 전파하는 프로그램이다. 일반적으로 메모리 내에 자가복제를 하는 프로그램을 의미했으나 최근에는 컴퓨터 상에서 네트워크를 통하여 자가복제를 하는 프로그램을 말한다. 웜은 이메일, 스크립트, IRC, P2P, IM 등과 같이 응용프로그램 기반에 따라 분류될 수 있으며, 대부분 고급언어(C++,Visual Basic)로 제작되어진다. 그리고 자가복제, 빠른전파, 서비스 거부현상 등과 같은 특징들을 가지고 있다.

3.2 인터넷 웜의 진화

자가복제를 하는 인터넷 웜은 기반구조가 잘 갖춰져 있는 현재의 인터넷에서 활동하기에 최적의 환경조건을 가지고 있고 이로 인해 큰 문제로서 부각되고 있다. 인터넷 웜의 과거와 현재를 살펴보면 최초의 활성웜(Active Worm)은 1988년 R.모리스에 의해 제작되어진 모리스웜이다[1]. 이것은 빠른 속도로 확산되었고, 썬과 백스 시스템을 6천대 감염시켰다. 이 사건으로 CERT 가

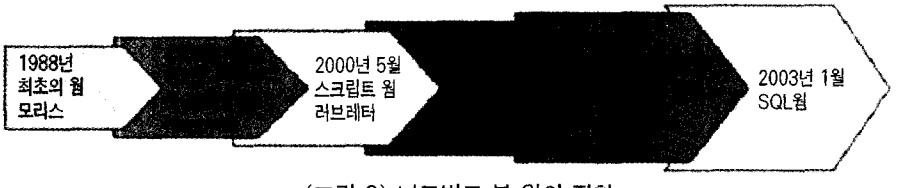
구성되었고, 모리스웜의 동작원리는 현재까지 웜의 전신이 되고 있다. 1996년에는 처음으로 마이크로소프트사의 워드제품에서 매크로를 이용한 바이러스가 나타나 빠른 속도로 전파되었다. 일반 사용자 영역으로 바이러스가 확대되었으며 사용자끼리의 문서교환과 특별한 의심 없이 실행함으로써 그 효과가 더욱 컸다. 물론, 과거에 이론적으로 또는 테스트형태로는 존재하였으나, 이렇게 실제로 대중화되어 처음으로 나타난 매크로 바이러스라는 이유도 있었다. 1999년 멜리사(Melissa) 웜이 나타나면서 이메일을 이용한 웜이 많이 나타났다. 멜리사는 아웃룩에 등록되어있는 주소를 이용하여 웜을 전송하기 때문에, 메일을 받은 사용자는 의심하지 않고 열어보게 됨으로써 전파속도가 빠르게 이어졌다[2]. 그리고 2000년 러브레터, 2001년 안나쿠르니코바 등 사회공학적 기법의 이용이 증가하였고, 개인사용자로 범위가 크게 확대되었던 계기가 되었다. 이메일 웜 이후로는 활성웜이 다시 등장하기 시작하였고 1998년 ADM, 1999년 밀레니엄, 2001년도에 라면, 라이온, 아도르, 새드마인즈, 치즈, 코드레드, 님다 등이 발견되면서 네트워크 보안의 필요성이 대두되었다. 2002년에는 Spida, Slapper 와 같이 관리상의 문제가 부각되었다.

3.3 대표적인 인터넷 웜

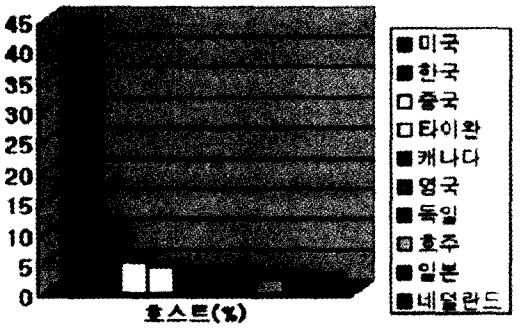
웜의 대표적인 것들로는 코드레드, 님다웜 등을 꼽을 수가 있다. 그 이유로는 전세계에 치명적인 피해를 입혔으며, 기존의 바이러스 형태보다 한 단계 진보한 모습으로 현재의 인터넷 기반시설에 커다란 위험을 줄 수 있다는 것을 보여주는 극명한 계기였다. 2001년 7월 코드레드웜은 마이크로소프트사의 IIS 4.0 과 IIS 5.0 이 작동하는 인덱스 서버의 .ida 취약점을 이용한 것이었다. 이 취약점은 'Unchecked Buffer in Index Server ISAPI Extension (MS01-033)' 으로 명명되며, 이를 이용해 웜은 감염된 시스템의 권한을 얻을 수 있게 되고, 감염된 시스템이 영문 윈도우 NT/2000 시스템이면 "Hacked by Chinese" 의 페이지로 변경되는 것이었다. 그리고 이 웜의 목적이라고 할 수 있는 www.whitehouse.gov 의 웹 사이트에 대하여 서비스 거부 공격을 시도하였으며, 감염된 시스템은 100K 가량의 데이터 패킷을 전송하게 되었고 이때 발생된 트래픽은 정상적인 서비스를 방해할 만큼의 많은 네트웍 트래픽을 유발하였다[3]. 다른 시스템의 감염방법에 있어서는 임의의 IP 주소 스캔으로 TCP 80번 포트로 연결시도를 하게 되며, 기존의 웜들은 특정 파일 형태를 유지하는 반면 코드레드는 메모리에만 상주하는 방법을 사용하였다.

〈표 2〉 국가별 패치 비율

국 가	패치적용(%)	패치 미적용(%)
영국	65.65	34.34
미국	59.59	40.41
캐나다	57.57	42.42
독일	55.55	44.44
네델란드	46.46	53.53
일본	39.39	60.61
호주	37.37	62.62
한국	20.20	79.79
타이완	15.15	84.84
중국	13.13	86.86



(그림 2) 년도별로 본 웜의 진화



(그림 3) 7월 19일 국가별 코드레드 감염 호스트

이 후에 나온 코드레드II 는 코드레드와 동일한 취약점을 이용하여 확산되며, 다른 점이라면 홈페이지의 첫 화면을 변경하지 않는다는 것이다. 다만, 트로이목마를 설치하여 기밀자료 유출 및 데이터 손실을 가져왔고, 백도어 설치로 인한 재공격 위험성이 존재하게 되었다. 코드레드로 인한 피해는 서버 40만대가 감염되었으며, 복구비용으로 26억불 피해가 발생했다. (그림 3)은 7월 19일 코드레드에 감염된 호스트 수를 국가별로 조사한 그래프이며, 한국은 10.57%를 차지해 2위이다. 〈표 2〉는 7월 19일 감염된 호스트를 8월 14일에 재조사하여 패치적용 여부를 국가별로 나타낸 것으로, 한국은 감염된 호스트 수가 2위였으나 패치적용은 7위를 보여주고 있다[4]. 이는 보안사고 발생 이후에 대응에 미진하였음을 보여준다. 2001년 9월 발견된 Win32/Nimda는 윈도우 파일바이러스로 다양한 감염경로를 활용하는 복합적인 형태의 특징을 가진다. 감염 경로로는 4가지가 있고 다음과 같다.

3.3.1 전자메일(Email)
- "Automatic Execution of Embedded MIME Types" 취약점 이용
- 전자메일 주소록을 검색하여 사용자에게 메일발송
- 사용자가 첨부파일을 실행하거나 메일을 읽기만 해도 감염

3.3.2 공유 드라이브(Shared Drives)
- 네트워크 읽기/쓰기 권한으로 공유된 시스템 감염
- 바이러스는 *.EML, *.NWS 파일로 확산

3.3.3 웹 서버(Web Server)
- 감염된 웹 서버의 모든 웹컨텐츠 파일(HTM,HTML,ASP)변경
- 클라이언트가 감염된 웹 서버에 접속하면 바이러스를 다운로드 하며 이때 바이러스를 열거나 실행하는 경우 클라이언트 감염

3.3.4 IIS(Internet Information Server)취약점
- "Unicode Web Traversal"취약점 이용
- 서버 감염 후 바이러스 파일 생성 (admin.dll, mmc.exe,readme.exe)

위와 같이 네트워크 자원을 최대한 활용하게 된다. 또한 시스템폴더에 RICHED20.dll 파일을 생성하거나 덮어 씌우기를 하여 일부 프로그램들이 정상적으로 실행되지 않거나 작업도중 에러를 발생할 수 있으며, 대표적인 프로그램으로는 오피스가 있다. 이외에 모든 폴더의 파일을 주기적으로 감염시키는 등의 기존의 한가지 방법만을 전파수단으로 이용하는 것이라 다양한 전파방법을 가지고 있다[5]. 님다 바이러스로 인한 피해도 전세계에서 약 830만대 컴퓨터 피해와 손실액이 5억 9천만 달러에 이른다고 한다[6]. 국내에서는 15만 건 이상의 컴퓨터가 피해를 입은 것으로 추정하며 피해신고접수만 7000여건 이상이었다고 보고되었다[7].

4. 1.25 인터넷대란의 사건분석

4.1 사건개요

2003년 1월 25일 배포된 SQL_Overflow웜은 전세계적으로 웜으로 인한 커다란 피해를 주었다.

SQL_Overflow 웜은 마이크로소프트사의 SQL 서버의 취약점을 이용하여 전파되었으며, 이로 인하여 국내뿐만 아니라 전세계적으로 인터넷 망이 마비되는 초유의 사태를 발생시켰다. 오후 2시 10분경 미국, 호주 등의 해외에서 유입된 것으로 추정되며 우리나라의 경우는 해외의 감염 정도를 비교해 보았을 때, 상대적으로 피해가 컸다. 그리고 이번 웜의 동작형태는 감염된 서버의 부하를 증가시키고 네트워크 트래픽을 크게 유발하게 되는 직접적인 원인을 제공하였다. 이런 직접적인 원인 이외에 네임서버 장애, 네트워크 속도 지연 및 마비 그리고 서비스거부공격(DoS)과 같은 간접적인 효과를 이끌어 냈다[9].

4.2 사고분석

SQL 웜은 2002 년 7월 24일 MS02-039에서 보고된 취약점을 이용하여 확산되었다. 이 웜은 코드레드와 같이 메모리에 상주하며, 감염된 시스템은 불특정 IP 주소로 UDP 1434 포트로 웜의 복제를 위하여 무한루프를 돌면서 ws2_32.dll 의 sendto() 함수를 이용하여 패킷을 반복적으로 보내게 된다. 취약성을 내포한 시스템뿐만 아니라 취약성을 가지고 있지 않은 시스템에도 패킷을 보내게 됨으로써 네트워크 트래픽의 과부화와 이로 인한 이상 현상들을 야기시켰다. SQL_Overflow 웜의 특징적인 부분은 바로 웜의 코드 자체가 매우 간단하게 구성되어 있다는 점이다. 즉, 376 바이트의 웜은 취약점이 있는 시스템에 버퍼오버플로우(Buffer Overflow)를 일으켜 외부로부터 임의의 명령어를 실행할 수 있도록 하여, 웜 코드를 불특정 IP 주소로 반복적으로 발송하는 것이다. 이전의 코드레드와의 차이점을 비교해 보면 다음과 같다.

첫째, 코드래드는 랜덤한 IP 주소를 선택하여 웜을 전파하게 된다. 하지만, 항상 랜덤하게 생성하는 것이 아니라 지정되어 있는 seed 값을 통하여 시작되는 IP 주소는 같게 만들어 새로운 IP 를 생성하기도 한다. 그러므로, 감염된 시스템은 이러한 루틴으로 같은 IP 주소리스트를 사용하게 된다. 이로 인하여 감염된 시스템에 다시 재감염이 시도되고 이때 많은 트래픽이 증가하게 된다. 하지만, SQL_Overflow웜은 취약점 시스템을 찾지 않고 완전히 랜덤한 IP 주소로 웜 코드를 전송한다. 만약 코드레드도 이와 같이 완전한 랜덤을 사용하였다면 더욱 빠른 속도로 많은 시스템을 감염시킬 수 있었을 것이다. 그렇지만, SQL_Overflow 웜 자체도 완벽하게 랜덤한 IP주소생성에 문제가 있다. 바로, 랜덤한 IP주소를 생성하기 위해 사용하는 GetTickCount() 함수의 계산시 논리적인 버그로 인하여 같은 대역에서 계속 루프가 발생할 수 있다.

둘째, 추가적인 연결과정을 필요로 하지 않는 UDP 프로토콜을 사용한다. TCP를 이용하는 코드레드는 Three-Way Handshake 설정과정이 필요하였지만, UDP의 경우는 이와 같은 과정이 필요하지 않으므로 짧은 시간 안에 많은 패킷을 생성해 낼 수가 있다. 이러한 이유로 SQL_Overlflow 웜이 빠른 속도로 전파될 수 있었던 이유중의 하나다. 현재 SQL서버로 사용하는 시스템과 네트워크 환경에 따라 감염 패킷을 전송하게 되는 속도가 달라지게 된다[8].

4.3 동작원리 분석

이 SQL 웜의 동작방식은 다음과 같이 크게 4단계로 나누어 볼 수가 있으며, 마지막 단계에서는 무한 루프로 계속 수행되게 된다.

I. 스택에 원하는 값을 채우는 단계
II. 활동을 위해 필요한 함수를 얻기 위한 준비 단계
III. 통신을 시작하기 위한 설정 단계
IV. 무한루프를 돌면서 데이터를 전송하는 단계

이러한 4 단계는 세부적으로 10개의 스텝으로 구성이 된다. 스텝1은 단계1에 해당되며, 스텝 2~4 까지는 단계2, 스텝 5~8 까지는 단계3 그리고 마지막으로 스텝 9~10 까지는 단계4에 해당한다.

4.3.1 스택에 원하는 값을 채우는 단계

(1) 스택에 원하는 값을 채우는 단계

원하는 값(sqlsort.dll 리턴주소인 42B0C9DCh: 42B0C9DCh에는 "jmp esp"명령이 있음)을 원하는 스택영역에 기록하기 위해, 스택에 42B0C9DCh와 24(18h)개의 쓰레기 값(01010101h)을 적는다.

4.3.2 활동을 위해 필요한 함수를 얻기 위한 준비 단계

(2) 사용할 dll 및 Win32 API 스트링 값 저장

필요한 함수를 얻기 위해 엔트리포인트를 얻는 작업을 하게 된다. 이때 원하는 dll 이름 및 Win32 API 이름의 스트링 값을 스택에 저장하며, 사용하는 dll 및 Win32 API 이름은 아래와 같다.

- dll 명) kernel32.dll, ws2_32.dll
- Win32 API명) GetTickCount, socket, sendto

(3) ws2_32.dll 과 kernel32.dll 의 기본 주소 얻기

코드레드는 RVA(Relative Virtual Addresses)라는 기술을 사용하고 있다. 이것을 한 마디로 정의하자면 GetProcAddress 의 주소를 얻어내는데 사용되는 기술을 의미한다. 즉,GetProcAddress 는 LoadLibraryA 의 주소를 다시 얻어내는데 이용되고 웜은 이 두 개의 functions 으로 웜이 필요로 하는 function 을 쉽게 알아낼 수 있다. ws2_32.dll과 kernel32.dll 의 기본주소를 얻기 위하여, sqlsort.dll 의 IAT(Import Address Table) Loadlibrary 엔트리 값(42AE1018h)을 이용하여, ws2_32.dll의 베이스와 kernel32.dll의 베이스를 얻어, 스택에 저장해 둔다.

(4) GetProcAddress의 엔트리포인트 찾기

GetProcAddress의 엔트리포인트를 찾기 위하여 핑거프린트를 검사하여, 버전에 맞는 적절한 GetProcAddress의 엔트리포인트 값을 찾는다. 다음 단계에서는 네 가지 설정을 수행한다. 먼저, 목적지 IP를 랜덤하게 설정하기 위해 GetTickCount() 함수로부터 얻은 값을 seed로 사용한다. 공격할 포트를 UDP 1434번으로 설정한 후, socket을 열고 sendto 함수를 사용할 준비를 한다.

4.3.3 통신을 시작하기 위한 설정 단계

(5) GetTickCount() 호출

과정 4를 통해 GetProcAddress의 엔트리 포인트를 찾았으면, 시스템이 최종으로 시작된 이후의 경과시간을 1/1000초 단위로 보여주는 GetTickCount()를 호출하고, 그 값을 얻는다.

(6) 목적지 포트 설정

0x1010101 과 0x9B040103 를 xor 하여 목적지 포트번호인 1434 번을 얻는다.

(7) socket() 호출

소켓을 생성하기 위해 socket()함수를 호출해야 한다. GetProcAddress의 두번째 argument로 "socket" 스트링을 넘긴다. 이 부분은 앞서, GetTickCount를 호출하기 위한 부분과 동일하며, 코드상으로는 아래와 같이 된다.

socket(AF_INET, SOCK_DGRAM, IPPORTO_UDP);

(8) sendto() 호출 준비

오픈한 소켓을 통해 데이터를 전송하기 위해 sendto()함수를 호출해야 한다. GetProcAddress의 두번째 argument로 "sendto" 스트링을 넘긴다.

여기까지의 단계가 완료되면, 다른 시스템으로 데이터를 전송하기 위한 모든 준비가 완료된다. 이제 무한루프를 돌면서 데이터를 전송하며, 이로 인해 시스템의 CPU를 상당부분 점유하게 된다. 아래 9, 10번 과정을 연속적으로 수행하며 루프 상태가 된다.

4.3.4 무한루프를 돌면서 데이터를 전송하는 단계

(9) 목적지 IP 설정

목적지주소를 랜덤하게 생성하기 위해 앞에서 얻은 GetTickCount의 값을 seed로 이용하여 목적지 IP 를 만든다.

(10) sendto() 함수 호출

sendto()함수의 원하는 값들을 설정하여 전송한다. Payload의 길이는 376(178h)로 설정된다.

4.4 피해현황

CAIDA(Cooperative Association for Internet Data Analysis)의 분석보고서에 따르면 전세계 모든 취약한 서버의 90% 이상을 약 10분 이내에 감염시킴으로써 지금까지 가장 빠른 전파 속도를 가진 웜으로 기록되었다. 또한, 코드레드에 감염된 서버가 매 37분마다 2배로 늘어났던 것에 비해 SQL_Overflow 웜에 감염된 서버는 8.5 초마다 2배로 증가했다. 빠른 전파속도만큼 한국은 전체 감염비율 면에서 11.8%를 차지해 세계 2위로 감염률이 높았다[8]. 하지만 피해상황으로는 한국이 가장 컸으며, 그 이유로는 주요 ISP의 인터넷 서비스가 5시간 이상 불통되었으며 이로 인해 온라인 쇼핑몰의 2~5억원 가량의 매출 피해, 인터넷 뱅킹의 마비, 온라인 예매 시스템 불능과 전국적으로 많은 PC 방들이 인터넷 불능으로 약 225원의 피해를 입은 것으로 삼성경제연구소는 밝혔다. 그리고 이번 SQL_Overflow웜으로 인하여 300% 이상의 트래픽 증가가 발생하였다고 보고되었지만 코드레드나 님다에 비해 복구에 드는 비용은 훨씬 적었다. 하지만, 이번 웜의 직접적인 피해자는 일반 사용자들이 대부분이어서 실제 피해액수에 비해 체감 피해가 다른 어떤 바이러스보다도 컸던 것으로 분석된다.

5. 인터넷 웜의 전망과 대책

앞으로의 웜의 미래는 어떻게 변화할 것인가? 인터넷의 범위는 더욱 넓어져 가고 있으며, 보안의 중요성은 더욱 높아지고 있다. 이제 많은 기업과 개인들은 이메일을 이용한 웜이나 또는 바이러스로부터 스스로를 보호하기 위해 바이러스 백신의 사용이나 바이러스 게이트웨이 제품등 여러 솔루션을 도입하고 있다. 그만큼 이러한 위험에 대한 변화의 인식이 이뤄지고 있으나, 위협들은 더욱 지능적이고 영리해져 가며 이런 보호기술을 피해가고 있다.

5.1 웜의 기술적 발전전망

미래의 웜의 발전방향은 인프라의 변화와 컴퓨팅의 미래에 따라 달라질 것이며 새로운 패러다임을 예고할 것이다. 새로운 변화에 대응하고 준비하기 위하여 어떤 행동적 특성을 갖는 웜이 나타날 것인가 다음의 몇 가지를 제시해 본다.

첫째, 보안취약점의 이용과 다중 전파방법이 사용될 것이다. 이것은 보안 취약점을 이용한 웜이 더욱 많아질 것을 의미하며, 이러한 형태는 한 가지의 취약점만이 아니라 여러 취약점을 이용하여 인터넷상의 모든 시스템에 웜 특성의 감염을 최대화하게 될 것이다. 또한 효과적이고도 빠른 전파를 위하여 한 가지 전파방법뿐만 아니라 다중취약성을 이용하는 것과 같이 이메일, 공유폴더 등과 같이 전파방법을 다양화 하여 확산과 전파속도에 더욱 초점을 맞출 것이다. 네트워크에 있는 호스트 시스템을 가능한 빠르게 감염시키는 것이 웜의

기본 목적인 만큼 앞으로 다중 전파방법 이외에도 웜의 효율적인 확산을 위하여 취약한 시스템을 스캐닝(Scanning)하여 감염시키는 'Warhol Worm' 개념 즉, 인터넷의 모든 시스템을 15분 안에 감염시킨다는 이론과 이보다 더 나아가 30초안에 감염시킨다는 'Flash Worm' 이론을 가진 웜들이 주류를 이루며 나타나게 될 것이다[10,11,12].

둘째, 취약점 공개 후 웜의 출현은 더욱 빨라질 것이다. 새로운 소프트웨어의 취약성이 발표되면 해당 패치를 제공하기 이전에 제작되어 배포되어야 감염 범위를 더욱 빠르게 넓힐 수 있기 때문이다. 과거의 취약점과 웜의 관계 추이를 살펴보면 주기가 점점 짧아지고 있는 것을 알 수 있으며, 취약점의 발표와 함께 하루 또는 몇 시간 안에 해당 취약점을 이용한 웜이 나타날 가능성이 있다.

셋째, 검출이 쉽지 않도록 지능화 되어갈 것이다. 안티바이러스(Anti-Virus) 프로그램 또는 IDS(Intrusion Detection System)와 같이 미리 정의된 행동적 패턴에 검출되지 않도록 하는 것이다. 감염된 시스템에서 동작하는 프로세스는 자신의 형태를 노출되지 않도록 프로세스를 숨기거나, 이러한 방법으로 백그라운드 상태에서 에이전트로 동작하며 마스터의 지시에 따라 공격도구로 변모 하기도 하며, 각 통신채널에는 암호화를 통해 외부의 노출에도 그 내용을 알아내지 못하도록 하거나 공개키 구조의 형태를 가질 수도 있다. 이미 현재의 웜 들이 이런 방법을 취하고 있거나 또는 새로운 방법을 도출해 낼 것이다. 이외에 지능화된 형태로 웜이 정보를 교환하는 상호협력형 웜(Coordinated Worm)을 전망해본다. 이것은 감염을 더욱 최적화하기 위하여 웜들이 상호통신하는 형태를 의미한다. 즉, 이미 감염된 시스템을 또 다른 감염시스템에서 감염을 시도하기 보다는 감염되어 있지 않은 시스템을 찾아 감염하는 것으로 웜 자체가 웜 네트워크를 만들어 감염속도를 더욱 높일 수 있다는 것이다[13].

5.2 IT환경변화에 따른 웜의 전망

웜의 발전방향을 예측해보면 다양한 행동패턴들을 가지고 있어, 앞으로의 웜이 사회적으로 미칠 영향이 적지 않음을 예견해 볼 수가 있다. 그렇다면 웜의 발전방향을 기준으로 더욱 대중화 되어 발전될 가능성을 가지고 있는 기술은 어떤 것들이 있는지 다음과 같이 전망해 본다.

5.2.1 Peer-to-Peer (P2P)

P2P의 의미를 찾아보면 "각 컴퓨터가 동등한 능력을 가지고 있어, 어떤 컴퓨터에서라도 통신 세션을 시작할 수 있는 통신 모델을 지칭한다"고 기술되어 있다. 바로 현재의 인터넷을 통해 개인 대 개인으로 자료를 주고받을 수 있는 서비스라고 생각할 수 있다. 초기의 P2P 서비스가 단순히 음악파일이나 동영상 및 그림파일 등을 주고 받는데 사용되었지만, 최근에는 실행 파일뿐만 아니라 모든 파일을 공유할 수 있는 방식으로 발전되고 있고 이것은 곧 자신이 원하지 않는 파일이 공유되거나 다른 사람에게 받는 파일의 안전성이 떨어진다는 것을 의미한다. 현재의 P2P 는 인터넷 시대에 각광 받고 있는 기술 중 하나이며, 보안적인 측면에서는 서버를 통하지 않고 개인 대 개인으로 파일을 주고 받기 때문에 파일의 안전성은 개인에 좌우되는 문제가 남아있어 앞으로 P2P를 활용한 웜의 증가가 예상된다.

5.2.2 인스턴트 메시지(Instant Messaging)

인스턴트 메시지는 현재 인터넷상에서 개인 대 개인의 또 다른 통신수단으로 많이 자리잡고 있다. 기업에서는 비즈니스를 수행하기 위한 하나의 용도로 사용되기도 하며, 인스턴트메시지를 정보제공의 또 다른 수단으로 만들어 정보를 제공 중이기도 하다. 대중적으로 이용되고 있는 만큼 악

성코드나 웜의 또 다른 전파경로로도 많이 이용될 것으로 보인다. 많이 사용되는 IM중의 하나인 MSN 을 통하여 전파되는 웜으로 BR2002, Supova 들이 이미 나타났다. 인스턴트 메시지의 시장규모는 더욱 커지고 있고 일반 사용자들의 일반적인 행동패턴은 무의식적인 클릭으로 이어지고 그 결과는 웜의 감염으로 제 2, 3의 피해자를 낳게 하는 악순환의 과정을 거치게 된다.

5.2.3 모바일 디바이스(Mobile Device)

바이러스와 웜의 다음 목표로 모바일 디바이스가 될 확률이 크다. 무선망의 증가와 PDA, 개인용노트북, 휴대폰의 무선망 연동은 모바일 상의 이메일 개인정보 관리와 같은 부분등 전체적으로 크게 성장할 것으로 보고 있다. 이러한 모바일 디바이스는 24 시간 지속적으로 연결되어 있는 특성 때문에 외부적 접근에 의해 중요한 정보가 누설될 수도 있으며 이런 무선 디바이스를 통하여 단숨에 전세계로 확산될 우려를 안고 있다[14].

5.3 인터넷웜의 대응방안

웜의 진화과정을 돌이켜보면 전파 방식과 감염속도 등의 특징에 있어서 계속 변화해 나가고 있고, 특히 감염 속도 면에서는 매우 빨라지고 있다. 그만큼 웜의 진화속도에 비추어볼 때 얼마나 빨리 탐지하고 대응할 수 있느냐 하는 문제는 지금까지 계속 제기되어 왔으며 현 프로세스를 발전시키기 위하여 다양한 조사와 연구가 이루어져 왔다. 전형적인 바이러스 특성에 네트워크적인 특성을 이용하는 악성코드가 점점 활성화 될 것으로 보이며, 이는 바이러스 단독 형태보다 웜의 전파방식과 트로이목마의 설치, 그리고 추후 공격에 이용할 수 있도록 하는 혼합된 형태의 악성코드가 더욱 활성화 될 것이다. 이를 대응하기 위해서는 바이러스 프로그램의 분석 및 치료기술과 네트워크 보안의 트래픽 제어기술등이 함께 고려가 되어진 통합 기술이 필요하다. 이것은 사후 처리에 해당하는 대응에 있어서도 통합된 대응의 필요성을 의미한다. 또한, 물리적 영역별로 구분을 하였던 국가기관의 역할도 보안사고에 대처하기 위해서는 변화가 필요하며, 효과적인 대응을 위해서는 모든 영역이 유기적으로 협력하여야만 한다. 시대가 변해가면 기존의 방식이 변해가듯이 이제는 이러한 변화의 추이를 인식하고 통합대응의 방안에 대해 기존의 대응방법과 병행하면서 최적의 방법을 모색해 나가야 할 것이다.

6. 결 론

앞으로의 웜은 지금까지 알아본 것과 같이 다양한 방법으로 확산경로가 넓어질 것이며 지능화 될 것으로 보인다. 웜은 현재 삶의 발전방향과 밀접한 관계가 있게 발전을 하고 있으며 피해 또한 국가 기반 시스템들이 마비되는 것과 같이 우리 생활에 직접적인 영향을 끼치게 된다. 이러한 악성코드의 진화 방향은 우리에게 보안의 중요성을 더욱 일깨워 주고 있으며, 이러한 변화 추세에 적합한 앞으로의 대비가 필요하다.

참고문헌

[1] Zesheng Chen, Lixin Gao and Kevin Kwiat, "Modeling the Spread of Active Worms," Proc. of IEEE INFOCOM, 2003.

[2] Nicholas Weaver, "A Brief History of The Worm," Nov. 1997, http://www.securityfocus.com/infocus/1515.

[3] eEye Digital Security, ".ida Code Red Worm," July 17, 2001, http://www.eeye.com/html/Research/Advisories/AL20010717.html.

[4] David Moore, Colleen Shannon, Jeff Brown, "Code-Red: a case study on the spread and victims of an internet worm," Proc. of SIGCOMM IMW 2002, Nov. 2002.
[5] Ahnlab, Inc, "Win32/Nimda," Sep. 2001, http://home.ahnlab.com/smart2u/virus_detail_866.html.
[6] 전자신문, 2001년10월04일자 기사
[7] 한국경제, 2001년09월21일자 기사
[8] CAIDA, "Analysis of the Sapphire Worm," Jan. 2003, http://www.caida.org/analysis/security/sapphire/.
[9] SQL_Overflow Worm disassemble code, http://www.eeye.com/html/Research/Flash/sapphire.txt, http://www.boredom.org/~cstone/worm-annotated.txt, http://www.nextgenss.com/advisories/mssql-udp.txt.
[10] Nicholas C Weaver, "Warhol Worms: The Potential for Very Fast Internet Plagues," http://www.cs.berkeley.edu/~nweaver/warhol.html.
[11] Stuart Staniford, Gary Grim, Roelof Jonkman, "Flash Worms: Thirty Seconds to Infect the Internet", Aug. 2001, http://www.silicondefense.com/flash/,Silicon Defense.
[12] Stuart Staniford, Vern Paxson and Nicholas Weaver, "How to Own the Internet in Your Spare Time," USENIX Security Symposium, Aug. 2002.
[13] Jose Nazario, with Jeremy Anderson, Rick Wash and Chris Connelly, "The Future of Internet Worms", Crimelabs research, July 20, 2001.
[14] Symantec, " Symantec Internet Security Threat Report - Attack Trends for Q3 and Q4 2002", Feb. 2003.

저자약력

정 관 진

1997년~2000년 NEXTEL 기술부
2000년 ~ 2002년 PSINet Korea 시스템 및 네트워크 보안 담당
2002년 ~ 현재 안철수연구소 시큐리티대응센터 (ASEC)
1997년 ~ 현재 Apache.Kr.net 운영
관심분야: 보안 취약성 분석, 보안사고 긴급대응, 시스템 및 네트워크 보안, 무선 보안

이 희 조

1993년 포항공과대학교 컴퓨터공학과 (공학사)
1995년 포항공과대학교 컴퓨터공학과 (공학석사)
2000년 포항공과대학교 컴퓨터공학과 (공학박사)
2000년 ~ 2001년 Purdue Univ. CS & CERIAS 연구원
2001년 ~ 현재 안철수연구소 기술기획실장
관심분야: 네트워크 보안, 인터넷 기반구조 보호기술, 결함포용 컴퓨팅, 병렬 알고리즘