VMware를 이용한 바이러스 테스트 시뮬레이션 설계 및 구현

한서대학교 이 중 식, 이 종 일, 김 홍 윤
한국정보보호진흥원 전 완 근

요 약

최근에 들어와서는 컴퓨터 바이러스와 해킹에 대한 공격이 심각한 수준에 와 있다. 이제 컴퓨터 바이러스는 특정 사건이 아니라 우리의 생활 속에서 발생할 수 있는 실질적인 피해임을 느낄 수 있다. 특히 1999년 이후에 나타난 바이러스는 다양한 변화를 보였고, 진보된 형태의 바이러스도 많이 나타났다. 일부 바이러스는 자신의 코드를 재배치하는 암호화 기술을 사용하기도 한다. 이에 따라서 백신 프로그램들은 바이러스의 암호화를 다시 복호화 하기 위해 가상 실행 엔진(emulation engine)을 사용하고 있다. 이러한 바이러스의 복잡한 암호화 기술 및 복호화 기술은 O.S의 종류에 따라 형식이 다양하다. 따라서 하나의 시스템에서 여러 가지 운영체제의 가상 실행 엔진을 사용하기 위해서 다수의 운영체제를 같이 사용할 수 있게 해주는 응용 소프트웨어인 VMware를 사용하여 '바이러스 Test Simulation'의 설계 및 구현을 할 수 있다.

Design and Implementation of Virus Test Simulation using VMware

Lee Joong Sik, Lee Jong Il, Kim Hong Yoon, Jeon Wan Keun

ABSTRACT

Comes in into recent times and there is on with a level where the attack against the computer virus and the hacking which stand is serious. The recently computer virus specific event knows is the substantial damage it will be able to occur from our life inside is a possibility of feeling. The virus which appears specially in 1999 year after seemed the change which is various, also the virus of the form which progresses appeared plentifully. The part virus does it uses the password anger technique which relocates the cord of the oneself. Hereupon consequently the vaccine programs in order decode anger to do the password anger of the virus again are using emulation engine. The password anger technique which the like this virus is complicated and decode anger technique follow in type of O.S and the type is various. It uses a multi emulation engine branch operation setup consequently from one system and to respect it will be able to use a multiple operation setup together it will use the VMware which is an application software which it does as a favor there is a possibility where it will plan 'Virus Test Simulation' and it will embody.

1. 서 론

21세기가 시작되면서 인터넷은 하나의 거대한 네트워크로 발달하여 세계는 하나로 묶여가고 있다. 이로 인하여 오늘 만들어진 바이러스가 내일이면 인터넷을 통해 약 6천만 대의 컴퓨터에서 활동할 수 있게 되었다. 멜리사(Melissa)와 'I Love You' 같은 웜 바이러스는 몇 시간 이내에 수만 개의 네트워크를 다운시키고, 세기말에 등장한 CIH 바이러스는 하루도 되지 않아 만여 대 이상의 ROM 바이오스 칩을 파괴하면서 하드디스크의 데이터를 무용지물로 만든다. 그리고 요즘은 님다(Nimda)나 코드레드(Codered)와 같이 해킹 기법을 이용한 바이러스로 인해 의도적으로 개인의 사생활과 데이터를 파괴하기도 한다[1][2].

그러나 이러한 현실에서 마이크로소프트 Windows 계열의 여러 O.S(Operating System)의 기본 보안 설정이 안전하지 못하다는 것은 널리 알려진 사실이다. 이와 더불어 유닉스 계열의 O.S 에서는 훨씬 더 심각한 보안상의 취약점이 발견되고 있다. 즉 Linux, Sendmail, TCP/IP, Buffer Overflow, 네트워크 파일 시스템 등은 시스템의 종류에 상관없이 바이러스와 해킹에 공격당할 수 있는 취약점을 가지고 있다. 이렇듯이 지금의 인터넷에는 마이크로소프트의 Windows를 포함하여 매킨토시(Macintosh)나 Unix, Linux 등의 여러 운영체제를 대상으로 만든 수많은 바이러스가 활동하고 있으며, 크로스 플랫폼 매크로 바이러스처럼 운영체제에 상관없는 바이러스 또한 무수히 존재한다[3].

최근에는 일부 바이러스가 백신 프로그램을 무력화시키기 위해 바이러스 자신의 코드를 재배치하는 암호화 기술을 사용하기도 한다. 이에 따라서 백신 프로그램들은 바이러스의 암호화를 다시 복호화 하기 위해 가상 실행 엔진(emulation engine), 즉 바이러스 테스트 시뮬레이션을 사용하고 있다. 그러나 이러한 바이러스의 복잡한 암호화 기술 및 복호화 기술은 O.S의 종류에 따라 형식이 다양하다. 따라서 하나의 시스템에서 여러 가지 운영체제의 가상 실행 엔진을 사용할 수 있게 하는 환경을 만들어 주기 위해 다수의 운영체제를 같이 사용할 수 있게 해주는 VMware 소프트웨어를 사용하여 바이러스 테스트 시뮬레이션의 설계 및 구현을 할 수 있다[3][4].

본 논문에서는 먼저 바이러스 Test bed에 대한 기본 지식을 이해하고, VMware 소프트웨어를 이용하여 하나의 시스템 내에서 여러 가지 운영체제를 이용하여 바이러스 테스트를 할 수 있는 시뮬레이션을 설계 및 구현하여 테스트를 해 봄으로서 현재 존재하고 앞으로 나타날 수 있는 바이러스에 대한 즉각적인 대처와 예방을 할 수 있게 하는데 목적이 있다.

2. 바이러스 Test bed

이 바이러스 테스트 베드는 각 바이러스의 정확한 특성 및 활동 범위를 파악하여 앞으로 있을 더 큰 피해를 막기 위하여 사용하는 것에 주된 목적이 있다고 볼 수 있다[5][6].

2.1 바이러스 Test bed의 필요성

여러 종류의 바이러스와 더불어 매크로 바이러스는 윈도우 계열의 운영체제 환경과 함께 anti-바이러스를 평가하는 제품의 발전하는데 있어서 많은 어려움을 가져왔다. 그리고 윈도우 환경의 사용자 인터페이스는 anti-바이러스 제품 평가의 작업을 자동화하는 것에 대해서 많은 문제점이 도출되고 있다[7].

이러한 문제에 직면하여 매크로 바이러스에 대하여 활동 영역과 피해 정도를 윈도우 환경 안에서 자동으로 테스트를 해주는 시스템의 필

요성에 의하여 바이러스 테스트 베드가 이루어진 것이다. 이 방법은 특히 anti-바이러스 제품의 평가에 목적이 있는 사람들에게는 대단한 관심을 주게 되었다. 이 바이러스 테스트 시스템은 자동으로 컴퓨터 바이러스의 복사본을 만들 수 있고, 윈도우 환경 또한 자동화 할 수 있다[6].

2.2 바이러스 Test bed 시스템

지금까지 많은 종류의 바이러스 Test bed 시스템이 논의되고 연구되어 왔다. 그러나 현재까지 발표된 바이러스 Test bed 시스템은 주로 MS-DOS 환경에 치우쳐 있고, 바이러스 프로그램 파일의 실행에 관해서만 집중되어 왔다[8][9]. 다음의 (그림 1)은 바이러스 Test bed 시스템의 일반적인 원리와 구성요소를 나타낸 것이다. .

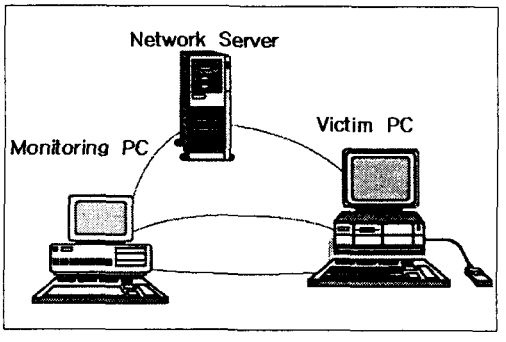

그림 1 바이러스 Test bed 시스템 구성요소

2.2.1 Victim PC

Victim PC의 주된 기능은 바이러스에 직접 감염되어 바이러스 코드가 실행되는 PC이다. 자동적으로 바이러스 코드가 실행하고 있는 동안 Victim PC는 바이러스에 감염이 되었는지 안되었는지는 모르고 있다. 바이러스에 감염이 안된 상태에서 Victim PC는 테스트 시스템 감염 분석을 실행한다. 그것은 바이러스에 감염된 파일을 찾고 시스템의 원상대로의 회복을 수행한다. 예를 들면 감염된 바이러스 코드에 대해 시스템은 바이러스 코드의 복사본을 자동적으로 만들려고 한다[10].

2.3.2 Monitoring PC

Monitoring PC의 기본적인 기능은 Victim PC로부터 바이러스에 감염되어 움직이고 있는 프로그램에서 한 세트의 작업을 기다린다. Victim PC에 프로그램을 하고 난 후에도 모든 작업이 실행되고 또 다른 집단의 작업을 기다린다. Monitoring PC는 필요할 때마다 확장된 시스템의 자동적으로 제어되는 바이러스 코드 실행 시스템과 더불어 Victim PC를 다시 원상태로 고쳐놓고 Victim PC의 잘못된 수행을 제어한다.

Victim PC가 Reset 하게 될 때 Monitoring PC는 Victim PC를 실행시키는 Boot 드라이버를 선택할 수 있다. Monitoring PC는 하드 디스크로부터 어떤 모양의 플로피 디스켓 드라이버라도 네트워크에서 자동적으로 실행될 수 있다. 이 때 변환을 하고 있는 Boot 드라이버는 부트 섹터 바이러스를 위해 필요하다. Monitoring PC는 또한 수행한 작업의 기록을 저장해 두고, Victim PC의 기억 공간을 변환한다[10][11].

2.2.3 Network Server

네트워크 서버는 몇 가지의 기능을 가지고 있다. 바이러스에 감염된 파일의 처리가 완료되고 난 후의 처리되었던 감염된 파일은 네트워크 서버로 하나의 파일로 들어가게 된다. 그 파일은 수행 과정에 있는 하위 디렉토리까지 옮겨지며, 만일 변경된 파일을 찾게되면 변경된 파일과 Boot 이미지는 목표 디렉토리 쪽으로 옮겨진다. 그리고 감염된 파일의 수행과 파일의 이

름에 맞추어 대응하고 있는 하위 디렉토리로 만들어진다. 변하는 디스켓의 이미지는 감염된 파일에 기록된다[11].

네트워크 서버는 또한 네트워크 위에서 바이러스에 감염되지 않은 고정 디스크와 플로피 디스크 이미지의 저장에 이용된다. 여기에서 바이러스에 감염되지 않은 Victim PC는 이미지 파일에서 자동적으로 저장된다. 네트워크 서버는 Victim PC를 바이러스에 감염되지 않은 상태에서 실행하기 위해 사용한다. 저장되어지는 바이러스에 감염 안된 Boot 이미지는 네트워크 서버의 로그인 디렉토리에 있고, 바이러스에 감염 안된 Boot가 필요할 때마다 Victim PC의 Boot ROM은 Boot 이미지를 사용한다[11][12].

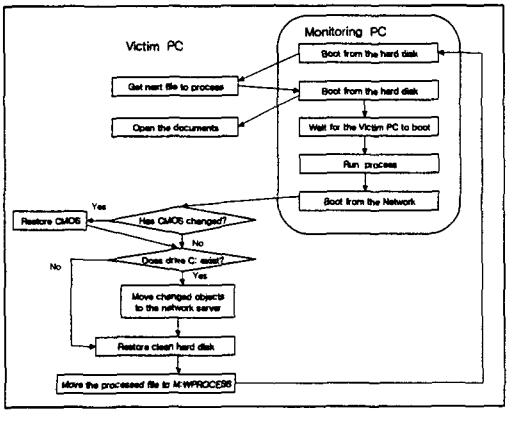

그림 2 일반적인 Victim PC와 Monitoring PC의 실행 과정

위의 (그림 2)는 매크로 바이러스를 테스트하는 동안 Victim Pc와 Monitoring PC의 수행과정을 간략히 도식화하여 나타낸 것이다. 그림에서 직사각형은 PC, 다이아몬드는 수단 방법의 선택과 화살의 방향에 따라 나아가는 것을 의미하고 있다.

3. VMware의 사용 및 원리

앞에서 설명한 최근 바이러스의 복잡한 암호화 기술 및 복호화 기술은 O.S의 종류에 따라 형식이 다양하다. 따라서 하나의 시스템에서 여러 가지 운영체제의 가상 실행 엔진을 사용하기 위해서 다수의 운영체제를 같이 사용할 수 있게 해주는 'VMware' 프로그램을 사용하여 바이러스 테스트 시뮬레이션의 설계 및 구현을 할 수 있다.

여러 가지 운영체제를 실제 시스템에서 사용하려면 Multi-booting을 해야 하는데, Multi-booting은 여러 운영체제를 사용할 수 있다는 장점은 있지만, 운영체제를 바꾸려면 재부팅 해야하는 번거로움이 있고, 한 운영체제에서 발생한 문제가 다른 운영체제에 영향을 미칠 수도 있다. 하지만 VMware를 사용하면 재부팅을 하지 않고도 서로 다른 운영체제를 필요할 때마다 사용할 수 있을 뿐만 아니라 하드 디스크의 용량이 허용하는 범위 내에서는 수십 개의 운영체제를 실행시킬 수 있다[13].

3.1 VMware의 사용 방식

VMware는 일종의 응용 프로그램이라고 할 수 있다. VMware를 사용하게 되면 DBCS를 지원하는 Windows NT 같은 계열의 운영체제를 구동 할 수 있고, 여러 플랫폼상에서 개발한 Web 솔루션이나 바이러스 등을 테스트하는 것에 많은 도움이 된다. VMware에서는 윈도우 창 없이도 윈도우 브라우저를 사용하여 프로그램 코드를 테스트 할 수 있으며, 광범위 LAN으로 네트워크가 가능하므로 각각의 네트워크에서 솔루션을 테스트 할 수 있다[9].

VMware는 실존하는 하드웨어 정보들을 공유하여 가상으로 하드웨어의 층을 만든다. 다시 말해 시스템의 하드웨어를 공유함으로 해서 현재 사용중인 운영체제에 영향을 끼치지 않은 상

태에서 다른 운영체제를 실행하는 것이다. 이러한 원리로 작동하기 때문에 다른 운영체제에 Error가 생기더라도 창 내에서만 멈추게 되므로 안정성도 뛰어나다. VMware는 지금까지의 컴퓨팅 환경에서 보다 유연하고 생산성 있는 컴퓨팅 환경을 사용자에게 제공한다[14].

3.2 VMware의 구동 원리

VMware는 인텔 X86 계열의 CPU를 사용하는 컴퓨터에서 작동하도록 만들어 졌으며, 호환칩 개발사인 AMD의 K6 시리즈와 Cyrix MII 시리즈에서도 동작한다. VMware는 기존의 컴퓨팅 환경에서 제한되었던 한가지 O.S만을 선택할 수밖에 없는 문제를 VMware Virtual Platform 과 VMware Virtual Machin 개념을 도입해 어떻게 보면 참으로 혁신적이고 실험적인 성과를 낳았다고 볼 수 있다.

VMware Virtual Platform은 멀티플 오퍼레이팅 시스템 환경을 위하여 두꺼운 소프트웨어 레이어를 생성한다. 이 소프트웨어 레이어에서는 동시 다발적으로 X86 기반의 같은 하드웨어와 같은 리소스를 사용할 수 있도록 제어하게 된다[13][14]. 다음의 (그림 3)에서 보듯이 VMware 가상 플랫폼은 각각의 VMware 가상 머신들이 서로 파일과 디바이스들을 공유하여 작동하도록 해주는 기반이 된다. 이것이 가능한 것은 각각의 가상 머신들은 자신만의 고유한 네트워크 아이디를 가지게 되어 VMware 가상 플랫폼과 통신하기 때문이다. 이것을 이용하여 VMware는 멀티플 O.S 환경과 그 응용 프로그램들을 싱글 컴퓨터 기반에서 수행이 가능하도록 해주는 것이다.

VMware 가상 플랫폼 상에서 Vmware 가상 머신 기반의 응용 프로그램들이 치명적인 Error를 일으켰을 경우에도 이것은 가상 머신 바깥의 Real 머신에는 전혀 영향을 주지 않는다[14][15].

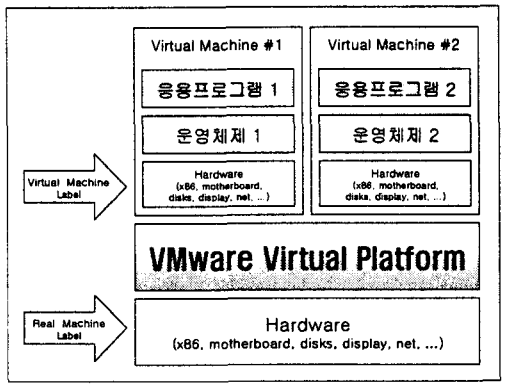

그림 3 VMware의 구동 원리

4. VMware를 이용한 바이러스 Test Simulation 설계 및 구현

4.1 바이러스 Test Simulation 설계

먼저 시뮬레이션을 구현 할 PC에 같은 하드웨어의 옵션과 함께 BIOS 버전이 같은지 확인해야 한다. 그리고 구현을 하면서 설정했던 옵션값들이 쉽게 변화될 수 있으므로 옵션에서 설정해 주었던 모든 값에 주의를 기울여야 한다. Windows 계열의 운영체제에서는 적당한 언어로 FDISK를 실행시켜 설정을 해 준다. 주의해야 할 점은 Windows NT는 디스크의 FAT32 형식을 지원하지 않고, Windows 98은 추가 도구가 없이는 디스크의 NTFS 분할을 읽을 수 없기 때문에 시뮬레이션을 구현하는 PC에는 오로지 디스크의 FAT16 형식을 사용해야만 한다.

이 시점에서 모든 운영체제는 디스크의 파티션을 분할하는데 있어서 FAT16의 형식을 사용함으로 해서 각각 최대 2 GB의 용량을 초과할 수 없다. 디스크의 마지막 파티션은 Swap 파일뿐만 아니라 파티션 이미지를 위한 도구를 더하여 모든 운영체제의 이미지 파일을 위해서 준비된 영역이다. 그러나 이러한 조건은 시뮬레이션을 구현하는데 있어서 별로 문제를 주지

는 않는다[15].

4.2 바이러스 Test Simulation 구현

앞에서 설명한 바이러스 Test Simulation의 환경과 설계에 맞추어 여러 가지 윈도우 운영체제에 적합한 바이러스 Test Simulation을 구현할 수 있다.

4.2.1 Windows 98

윈도우 98 운영체제에서는 Booting 메뉴를 MSDOS.SYS 파일로 바꾸어 놓고, [Options] 섹션 아래의 BootMenu를 'BootMenu=1'로 설정하는 것을 시작으로 구현한다. 그리고 만일 이미지 파일을 저장하거나, 본래 상태로 되돌리기 위해 필요하면 Command 라인에서부터 시작할 수 있다. 이것을 위해, 'PATH' 변수는 AUTOEXEC.BAT 파일로 변환하여 직접 접근하기 위해 설정되어야만 한다. 또한, SYSTRAY.EXE 파일을 제외하고 'HKLM\Software\Microsoft\CurrentVersion\Run' 의 아래에서 'Run' 등록 키에서 모든 프로그램의 실행을 삭제한다. 시스템이 매우 빠르고 전혀 필요하지 않는 작업을 생략한 이 방법은 모든 Boot에 대하여 실행된다.

4.2.2 Windows NT

Windows NT의 Server와 Workstation은 윈도우 98 이나 윈도우 ME보다 상당히 작다. 그러나 성능 향상을 위한 공간은 여전히 남아 있다. 예를 들면, 윈도우 NT는 구현 과정에서 모든 'Help' 파일을 정확한 'Help' 폴더에서 뿐만 아니라 'SYSTEM32'의 아래에서 두 번 발견한다. 'SP 6a'와 보안의 재설치가 'Rollup Package'를 시스템에 장착시킨 후에는 '$NTUninstall$'과 같은 모든 폴더를 삭제한다.

서비스 팩 또는 'Hotfixes'에 의한 모든 수정되었던 자료의 보관은 여기에 저장된다. 그것은 공간을 저장하지 않는 Log를 하는 것에 도움이 된다. 그리고 테스트 시뮬레이션을 시작하면 더 좋은 결과를 얻기 위해서도 필요하다[16].

4.2.3 Windows 2000

Windows 2000에서는 이미지 파일의 크기가 상당히 축소될 수 있다. 우선 모든 Driver 파일은 서비스 팩에 의해 설치되는 더 작은 한 두 개의 CAB 파일뿐만 아니라 DRIVER.CAB라고 이름이 지어지는 큰 CAB 파일에서 '%windir%\Drive Cache\i386' 라는 폴더에 보관될 것이다. 이 파일들은 삭제해서는 안되며, 네트워크의 설정을 위해 저장을 해야한다. 모든 테스트 시뮬레이션은 이 저장된 파일을 공유 할 수 있으며, 단지 'DriverCachePath'을 검색하는 것에 의해 발견될 수 있도록 등록키를 바꾸어준다.

다음 단계로 '%windir%\ServicePackFiles\i386' 라고 이름이 지어지는 폴더를 볼 수 있다. 그것은 설치되었던 모든 서비스 팩 파일의 복사본을 포함하고 있다. 모든 파일을 네트워크 드라이버에 복사하고, 재부팅 한 다음 복사본 파일을 삭제시킨다.

여기까지 모든 윈도우 계열의 운영체제를 설치하고 난 후에 테스트하고자 하는 테스트 프로그램이나 테스트 바이러스에 필요한 Simulator를 구현하였다. 그리고 좀 더 작고 빠른 시스템을 사용하기 위해 윈도우 98을 설치하여 Simulator와 함께 사용한다. 이렇게 모든 구현을 마치면, Simulator 디스크 분할의 공간을 달리 설정해 본다. 이것은 조금의 저장 공간도 필요로 하지 않으며 더 많은 운영체제로 통합되는 테스트 Simulator를 만들게 해준다.

4.3 바이러스 Test Simulation 결과

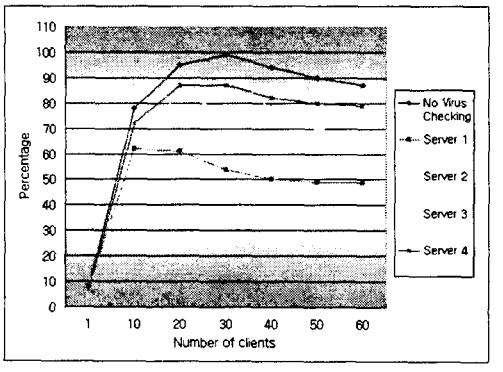

그림 4 Server에서 anti-바이러스 프로그램의 Load-point의 평균값

위의 (그림 4)는 4대의 외부 바이러스 Scanning 서버를 사용하는 anti-바이러스 Scanning 프로그램에서의 Load -point의 평균 결과 값을 보여주는 그래프이다. 이 테스트는 먼저 anti-바이러스 Scanning의 전체적인 실행에 대한 결과 값을 측정하기 위해 첫 번째로 어떠한 anti-바이러스도 포함하지 않은 일련의 기준선 테스트를 실행했다. 그 다음으로 서버를 감시하고 있는 외부의 바이러스에 관한 여러 가지 anti-바이러스 소프트웨어 패키지를 설치하고, 기준선의 수를 생성하기 위해 사용되는 테스트에 같은 실행의 테스트를 연속 되풀이했다.

시뮬레이션을 테스트하는 동안 서버를 조사하고 있는 각 새로운 바이러스와 함께 올라가고 있는 실행 수준과 더불어 테스트 동안 내내 처리되는 Scan을 요청하는 수의 증가가 나타났다. 파일 서버 위에서 가능하게 되는 anti-바이러스의 체크와 더불어 기준선 시험 결과와 비교될 때 전체적인 파일 서버의 실행은 낮게 나왔다.

그러나 이 시뮬레이션에서는 전체적인 실행 수준이 파일 서버에서 'Load balancing system'에 붙여지는 각 추가 외부의 서버로 상당히 증가하는 것을 보게 되었다. 두 번째의 외부의 바이러스 Scanning 서버를 더할 때는 실행의 최대의 유효 증가를 보여 주었다.

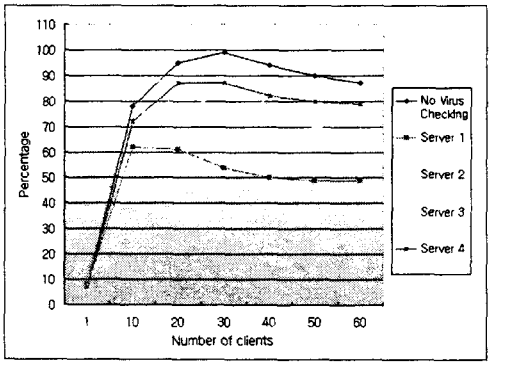

그림 5 Server에서 모든 anti-바이러스 프로그램의 실행 결과

위의 (그림 5)는 파일 서버에서 모든 anti-바이러스 프로그램의 실행 결과 값을 보여주는 그래프이다. 그래프에 대한 결과 값은 바이러스 테스트를 하는 동안 내내 여러 가지 Load-point에서 모든 anti-바이러스 소프트웨어를 사용하면서 얻게 되는 처리량의 평균 값이다.

파일 서버 위에서 감염된 파일을 조사하고 있는 바이러스와 함께 생성되는 기준선의 결과는 이 테스트 동안 얻게 되는 가능한 재료 처리량의 100퍼센트를 표현하고 있다.

(그림 5)의 그래프는 바이러스 테스트에서의 최상의 결과치를 뽑아낸 것이다. 그래프에서 Server 1의 곡선은 감염이 가능하게 하는 바이러스 Scanning과 함께 파일 서버를 사용하면서 얻게 되는 재료 처리량을 표현한 것이다. 나머지 곡선들은 외부의 바이러스가 기준선 테스트 동안 이루어지는 재료 처리량의 퍼센트이다. 그래프에서 나타나듯이 파일 서버 위에서 감염된 파일을 조사하고 있는 바이러스와 함께 테스트를 하는 동안 시뮬레이션의 시스템을 조사하고

있는 단지 하나의 외부의 바이러스를 사용하는 것은 대략 40퍼센트가 낮게 이루어지는 최고의 자료 처리량의 결과 값을 가져왔다.

5. 결 론

이제 바이러스는 더 이상 정보 자체에 대한 보호만이 아니라 컴퓨터 같은 정보 저장매체는 물론 정보 유통수단과 정보 생성도구에 대한 보호까지도 포함하는 광의의 개념을 포함하게 되었다. 바이러스가 네트워크를 통해 전자우편과 IRC 등으로 전파됨으로써 그 확산 속도가 빠르고 파괴적인 특징을 갖게 되었다. 특히 CIH 바이러스의 출현과 함께 1999년에 나타난 바이러스는 다양성 면에서 많은 변화를 보였고, 진보된 형태의 바이러스도 많이 나타났다. 1990년대 중반 이후 바이러스는 계속적으로 증가하고 있다. 바이러스 제작기술의 향상으로 양적인 증가와 함께 그 위험성도 많이 높아졌다. 그리고 해킹기술을 응용한 트로이 목마 프로그램도 많은 변종과 함께 발견되어, 악성 프로그램의 위협이 이제는 정보유출과 정보 변조에까지 미치고 있고, 그 대응도 점점 어려워지고 있다.

최근에 들어와서는 바이러스의 기법이 백신 프로그램의 진단을 피하기 위해 자신의 코드를 재배치하는 '암호화(Encryption)' 기술, 이러한 바이러스의 암호화와 복호화 부분을 불규칙하게 변경시키는 '다형성(Polymorphism)' 기술, 바이러스 자신의 코드를 은폐하거나 사용자나 백신 프로그램에게 거짓 정보를 제공하는 코드를 가지는 '은폐형(Stealth)' 기술, 다양한 암호화 기법과 은폐형 기법을 모두 포함하는 '갑옷형(Armour)' 기술 등을 사용함으로써 바이러스를 예방하는데 있어서 상당한 어려움이 따르고 있다. 더불어 이러한 기술들은 운영체제의 종류에 따라서도 기법이 달라진다.

이에 따라서 하나의 시스템에서 여러 가지 운영체제를 같이 사용하게 해주는 응용 소프트웨어인 'VMware'를 사용하여 '바이러스 Test Simulation'을 구현하였다. 그리고 바이러스 테스트를 해 봄으로써 바이러스에 대한 감염증상과 작동원리, 감염경로 등을 찾을 수 있다. 결국 이 'VMware'를 이용한 '바이러스 Test Simulation'의 결과를 이용하여 최근의 바이러스에 대한 효과적인 백신 프로그램을 만들 수 있다.

참고문헌

[1] Roger A. Grimes 저, 왕성현 역, "Malicious Mobile Code", 한빛미디어, 2001. 12.

[2] The WildList Organization International, Read Joe Wells' update http://temp.wildlist.org/

[3] 한국전산원 편집, "2000 국가정보화백서", pp. 354-367, 2000. 12.

[4] 아주대학교, "학습형 바이러스 면역 기본 시스템 개발", 한국정보보호 진흥원 - 연구보고서, 2000.

[5] Lammer, Victoria, "Survivor's Guide to Computer Viruses : Virus Bulletin '93", 1999. 6

[6] 안철수 연구소, http://home.ahnlab.com/-securityinfo/

[7] Ward, Brian, "Book of VMware", 2001. 11.

[8] Bontchev Vesselin (1996), "Possible Macro Virus Attacks and How to Prevent Them", In proceedings of the International Eicar Conference 1996, Lintz, Austria. Hosted by DataPROT Linz. pp. 61-87.

[9] Virus Bulletin, http://www.virusbtn.com/-magazine/overview/index.xml

[10] Polk, W. Timothy/ Wack, John P./ Bassham, Lawrence E./ Carnahan, Lisa J./ William Andrew Inc, "Anti-Virus Tools and Techniques for Computer Systems", 1995.
[11] Linuxlab, http://www.linuxlab.co.kr/docs/00-11-3.htm
[12] E-Testing Labs, http://www.etestinglabs.com/main/reports/emccelerra.pdf, 2001. 7.
[13] VMware Co. Ltd, http://www.vmware.com/support/ws3/doc/whatsnew_ws.html
[14] Helenius Marko (1997), "Antivirus Scanner Analysis Based on Joe Well's List of PC viruses in the wild 7/1997", Available electronically via anonymous ftp as ftp://ftp.cs.uta.fi/pub/vru/documents/test1997.zip (January 29, 1998)
[15] Applications of Informatics in Arts and Science University of Hamburg - CSDepartment, http://agn-www.informatik.uni-hamburg.de/
[16] Annual EICAR Conference, http://conference.eicar.org/pastconfer-ences/1998/other/autodbl.pdf, 2001.

이 중 식

2000년 한서대학교 컴퓨터과학과(이학사)
2002년 한서대학교 정보보호공학과(공학석사)
2002년 ~ 현재 한서대학교 시간강사
관심분야 : 컴퓨터 바이러스, 네트워크 보안
E-mail : jslee@hanseo.ac.kr

전 완 근

1998년 한서대학교 전산정보학과(이학사)
2000년 한서대학교 전산학과(이학석사)
2000 ~ 현재 한국정보보호진흥원 연구원
관심분야 : 컴퓨터 바이러스, 해킹, 인터넷 라우팅
E-mail : wkjeon@kisa.or.kr

이 종 일

2000년 한서대학교 물리학과(이학사)
2002년 한서대학교 정보보호공학과(공학석사)
2002년 ~ 현재 한서대학교 시간강사
관심분야 : 정보보호, 무선 인터넷 보안
E-mail : jepplee@hanseo.ac.kr

김 홍 윤

1982년 인하대학교 전자계산학과(학사)
1984년 인하대학교 전자계산학과(석사)
1996년 인하대학교 전자계산학과(박사)
1995년 ~ 현재 한서대학교 컴퓨터통신공학과 부교수
관심분야 : 인터넷 라우팅, 컴퓨터 바이러스, 인터넷 보안
E-mail : hykim@hanseo.ac.kr