Mobile IP 세대를 대비한 VMware 기반의 바이러스 Test bed

(Virus Test bed using VMware in Preparation for the Mobile IP Generation)

김 홍 일*
(Hongil Kim)

요 약 Mobile IP 단말기들에 탑재되는 운영체제는 일반적인 PC에서 탑재되는 운영체제와는 달리 확실한 독점체제가 확립되지 않은 상태이다. 따라서 당분간 Mobile IP 단말기에 운영되는 운영체제는 다양한 종류의 운영체제가 사용될 것으로 추정된다. 이러한 환경은 운영체제에 상관없이 변종되는 바이러스들에 매우 취약한 환경이며, 개별 단말기에서 사용되는 운영체제가 다양하기 때문에 이들 바이러스를 효율적으로 검출하여 치료하는 기술 개발을 매우 난해하게 한다. 본 논문에서는 VMware를 이용하여 다양한 운영체제에서 변종 바이러스를 테스팅할 수 있는 시스템을 설계하고 구현하였다. 또한 이를 실제 Mobile 환경에서 운영하기 위한 Prototype 시스템을 구축하였다.

핵심주제어 : Mobile IP 단말기, VMware, Prototype 시스템

Abstract Contrast operating system installed on traditional PC, operating system installed to mobile IP host is a status not to be established a definite monopoly system. Accordingly, for the time being, it is expected that operating system operated in mobile IP host will be used various operating systems. Regardless of operating system, this condition is a very vulnerable to modified virus. Because operating system used in an individual host is very diverse, It makes technical development to cure complicated in detection efficiently these virus. In this thesis, By using VMware under the various operating system, I desinged and implemented system which is able to test a modified virus. Also I implemented prototype system to operate under the real mobile condition.

Key Words : Mobile IP host, VMware, Prototype System

1. 서 론

이제 인터넷은 유비쿼터스와 Mobile IP 시대에 도래하면서 수억개 이상의 정보 단말기 유기체로 확장되고 있다. 이는 또한 인터넷에 확산될 수 있는 바이러스의 파급 범위가 넓어졌다는 것을 의미한다. 초기 바이러스는 데스크탑 컴퓨터나 일부 서버를 목표로 하여 제작되었고 전파될 수 있는 운영체제도 한두가지에 그쳤으나, 여러 운영체제에서 동작하는 바이러스로 진화하였으며, 마이크로소프트사 계열의 운영체제 독점이 점차적으로 줄어드는 현 시점에서 차기 운영체제 시장을 석권하기 위한 다양한 운영체제들이 제시되고 있는 상황이다. 더욱이 Mobile IP 시대의 도래에 따라 휴대가

* 대진대학교 컴퓨터공학과 부교수

가능한 소형 정보단말기들도 작은 규모의 독자적인 운영체제를 탑재하고 있기 때문에 운영체제의 다양화는 당분간 가속화될 전망이다.그러나 이러한 현실에서 마이크로소프트 Windows 계열의 여러 가지OS(Operating System)의 기본 보안 설정이 안전하지 못하다는 것은 널리 알려진 사실이다. 이와 더불어 유닉스 계열의 OS 세계에서는 훨씬 더 심각한 보안상의 취약점이 발견되고 있다. 즉 Linux, 센드메일, TCP/IP, Buffer Overflow, 네트워크 파일 시스템 등은 시스템의 종류에 상관없이 바이러스와 해킹에 공격당할 수 있는 취약점을 가지고 있다[3].

이렇듯이 지금의 인터넷에는 마이크로소프트의 Windows를 포함하여 매킨토시(macintosh)나 Unix, Linux 등의 여러 가지 운영체제를 대상으로 만든 수많은 바이러스가 활동하고 있으며, 크로스 플랫폼 매크로 바이러스처럼 운영체제에 상관없는 바이러스 또한 무수히 존재한다[4].

최근에는 일부 바이러스가 백신 프로그램을 무력화시키기 위해 바이러스가 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[4][5]. 그러나 이러한 바이러스의 복잡한 암호화 기술 및 복호화 기술은 OS의 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 시뮬레이션을 구현하여 바이러스 테스트를 할 수 있다[6].

2. 바이러스 Test bed 시스템

바이러스 테스트 베드는 1995년 'Eicar' 컨퍼런스에서 자동적으로 제어되는 바이러스 코드 실행 시스템에 대한 발표가 있었다[7][8]. 그러나 그때는 바이러스가 많이 존재하지 않았으며 실행 시스템은 단지 MS-DOS 운영체제 환경에서만 작업을 수행할 수 있었다. 이 바이러스 코드 실행 시스템은 매크로 바이러스의 문제 해결에 많은 도움을 주었으나 윈도우 환경에서는 해결되지 않은 문제가 많이 있었다. 그때는 윈도우 환경에서의 매크로 바이러스의 문제를 해결할 연구가 실행되지 않고 있었으며, 바이러스 프로그램 파일의 실행에 관해서만 집중되어 왔다[6].

2.1 Test bed의 필요성

여러 종류의 바이러스와 더불어 매크로 바이러스는 윈도우 계열의 운영체제의 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[9].

이러한 문제에 직면하여 매크로 바이러스에 대하여 활동 영역과 피해 정도를 윈도우 환경 안에서 자동으로 테스트를 해주는 시스템의 필요성에 의하여 바이러스 테스트 베드가 이루어진 것이다. 이 방법은 특히 anti-바이러스 제품 평가에 목적이 있는 사람들에게는 대단한 관심을 주게 되었다. 이 바이러스 테스트 시스템은 자동적으로 컴퓨터 바이러스의 복사본을 만들 수 있고, 윈도우 환경 또한 자동화 할 수 있다[10].

1996년 이후 매크로 바이러스를 비롯한 바이러스의 수는 빠른 속도로 증가했다. 바이러스의 급속한 증가 추세는 anti-바이러스 제품을 평가하는 사람들에게 있어서는 많은 어려움을 유발시켰다.

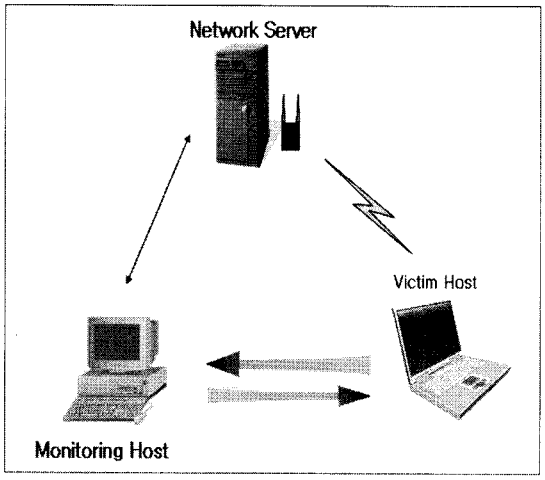
앞으로 계속 매크로 바이러스와 더불어 전통적인 이진수의 바이러스 또한 빠른 속도로 퍼지고 증가 할 수 있다. 추가적으로 윈도우 환경 내에서 Scan 하는 작업은 아직까지 매크로 바이러스를 발견할 능력이 없다는 것이다[9][11].

이러한 문제점은 매크로 바이러스가 윈도우 환경에서 얼마나 자동적으로 복사를 할 수 있는지, 윈도우 안에서 얼마나 다른 작업을 자동적으로 실행하게 하는지에 대한 의문점을 유발시키기에 충분했다.

2.2 Test bed의 원리

바이러스 테스트 베드 시스템의 구성은 몇 개의 기능들로 이루어진 각각의 PC들로 이루어진다. 이들 각 PC들은 저마다의 실행 기능이 틀리다. 이러한 구성 PC들의 기능이 조합을 이루면서 바이러스 테스트 베드의 종합적인 기능을 원활하게 실행되어지는 것이다[10].

다음의 그림 1은 바이러스 Test bed 시스템의 일반적인 원리와 구성요소를 나타낸 것이다. 그림 1에서 모든 시스템의 바이러스의 구성 요소는 실행 시스템을 부호화한다. 그리고 Victim PC와 Monitoring PC는 바이러스가 감염시키는 파일을 저장하기 위해 사용되는 네트워크 서버로 붙여지며, Monitoring PC는 Victim PC의 키보드와 부트 실행을 제어한다.



(그림 1) 시스템 개념도

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 감염시키는 파일을 찾고 시스템의 원상대로의 회복을 수행한다. 예를 들면 감염된 바이러스 코드에 대해 시스템은 바이러스 코드의 복사본을 자동적으로 만들려고 한다.

2.2.2 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 재설정하게 될 때 Monitoring PC는 Victim PC를 실행시키는 부트 드라이버를 선택할 수 있다. Monitoring PC는 하드 디스크로부터 어떤 모양의 플로피 디스켓 드라이버라도 네트워크에서 자동적으로 실행될 수 있다. 이 때 변환을 하고 있는 부트 드라이버는 부트 섹터 바이러스를 위해 필요하다.

Monitoring PC에서 네트워크의 부트 옵션이 하드 디스크와 플로피 디스켓에 감염된다고 할지라도 네트워크에 바이러스가 감염되지 않도록 하는 것은 근본적으로 가능한 일이다. Monitoring PC 또한 수행한 작업의 기록을 저장해 두고, Victim PC의 기억 공간을 변환한다[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 만든다. 만일 변경된 파일을 찾게되면 변경된 파일과 부트 이미지는 목표 디렉토리 쪽으로 옮겨진다. 그리고 감염된 파일의 수행과 파일의 이름에 맞추어 대응하고 있는 하위 디렉토리로 만들어진다.

예를 들면, 만일 원형 파일이 시스템의 경로에 저장되었다라고 생각하면 플로피 디스켓의 드라이버에서 찾게 된다. 변하는 디스켓의 이미지는 감염된 파일에 기록된다[11].

네트워크 서버는 또한 네트워크 위에서 바이러스에 감염되지 않은 고정 디스크와 플로피 디스크

이미지를 저장하는 것에 의해 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 부트 이미지는 네트워크 서버의 로그인 디렉토리에 있고, 바이러스에 감염 안된 부트가 필요할 때마다 Victim PC의 부트 ROM은 부트 이미지를 사용한다[12].

다음의 그림 2는 매크로 바이러스를 테스트하는 동안 Victim PC와 Monitoring PC의 수행과정을 간략히 도식화하여 나타낸 것이다. 그림 2에서 직사각형은 PC, 다이아몬드는 수단 방법의 선택과 화살의 방향에 따라 나아가는 것을 의미하고 있다.

2.3 Test bed의 사례

2.3.1 Automatic Virus Analyser System

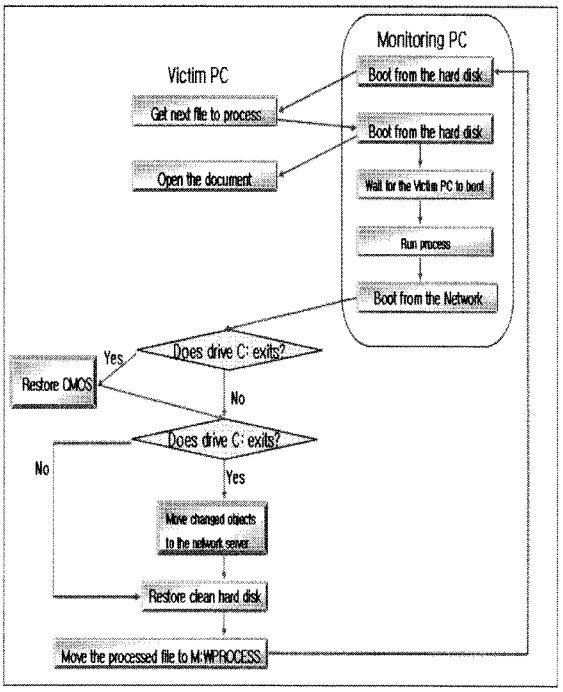
'Ferench Leitold'는 'Virus Bulletin' 컨퍼런스에서 'Automatic Virus Analyser System(자동 바이러스 분석 시스템)'을 발표했다. 이 시스템은 전자게시판 시스템에 기반을 두고 있으며, 자동적으로 제어되는 바이러스 실행 시스템과 똑같은 기능을 많이 가지고 있다. 그림 1을 참고하면 자동적으로 제어되는 바이러스의 Victim PC와 Monitoring PC에서 실행 시스템을 부호화하는 것은 Slave PC와 Master PC를 일치하도록 하는 것과 같다. 이 시스템에서의Master PC는 Victim PC의 바이러스에 감염된 플로피 디스크 드라이브를 다시 원상태로 회복시킬 수 있고, 바꿀 수도 있다. 그러나 어떠한 네트워크 도구나 드라이버도 키보드를 제어하는 장치는 없다. 따라서 이 시스템은 초기의 바이러스 코드 실행 시스템처럼 많은 중요한 기능들이 부족한 상태였다[13].

2.3.2 Virus Intrusion Detection Expert System

1995년에 'Morton Swimmer'는 Hamburg 대학의 바이러스 테스트 센터에서 개발한 'Virus Intrusion Detection Expert System(바이러스 침입탐지 전문가 시스템)'에 대하여 발표했다[11]. 이 시스템은 UNIX 운영체제 환경을 기반으로 8086 프로세서에 의존하고 있다. 시스템의 많은 기능들은 많은 함수들에 의해 이루어졌고, 간단한 바이러스 프로그램만이 실행 가능했으며, 윈도우 운영체제에서는 조금도 사용할 수가 없다. 시스템의 특징은 보통 PC의 지시로 바이러스 코드의 활동 내용을 알 수 있게 했다는 점으로 이것은 Test bed 시스템의 주된 목표이다[12].

2.3.3 Dr Solomon's Anti-Virus Toolkit

'David Aubrey-Jones'는 1995년에 같은 컨퍼런스에서 'Automatic Testing of Memory Resident anti-Virus Software'를 발표했다. 이 시스템은 Control PC, Test PC와 함께 각 PC들의 하드웨어의 연결과 네트워크 서버로 이루어져 있다. Test PC의 네트워크 Booting까지도 시스템의 일부로 이용되었다. 이 시스템 또한 겉으로 보기에는 바이러스 코드 실행 시스템과 많은 비슷한 점을 가지고 있다. 주된 차이점은 Memory Resident Scan이



(그림 2) 일반적인 Victim PC와 Monitoring PC의 실행과정

활동할 때 시스템이 파일 바이러스의 자동적인 실행에 대해 디자인되었다는 것이다. 이 시스템은 비록 더 한층 발달할 잠재성을 가지고 있었지만 바이러스 코드 실행 시스템의 틀에서 많이 벗어나지는 못했다. 현재 이 시스템은 윈도우 운영체제의 기반에서 실행할 수 있도록 확장되었다[11].

3. VMware를 이용한 Virus Test bed

여러 가지 운영체제를 사용하여 바이러스 Test bed 시스템을 실제 시스템에서 사용하려면 Multi-booting을 해야 하는데, Multi-booting은 여러 운영체제를 사용할 수 있다는 장점은 있지만, 운영체제를 바꾸려면 재부팅해야 하는 번거로움이 있고, 한 운영체제에서 발생한 문제가 다른 운영체제에 영향을 미칠 수도 있다. 하지만 VMware를 사용하면 재부팅을 하지 않고도 서로 다른 운영체제의 바이러스 Test bed를 필요할 때마다 사용할 수 있을 뿐만 아니라 하드 디스크의 용량이 허용하는 범위 내에서는 각각 수십 개의 운영체제를 사용하는 바이러스 Test bed를 실행시킬 수 있다.

3.1 VMware의 환경의 장점

VMware는 일종의 응용 프로그램이라고 할 수 있다. VMware를 사용하게 되면 DBCS (Double Byte Character Set)를 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 각자 네트워크에서 솔루션을 테스트 할 수 있다.

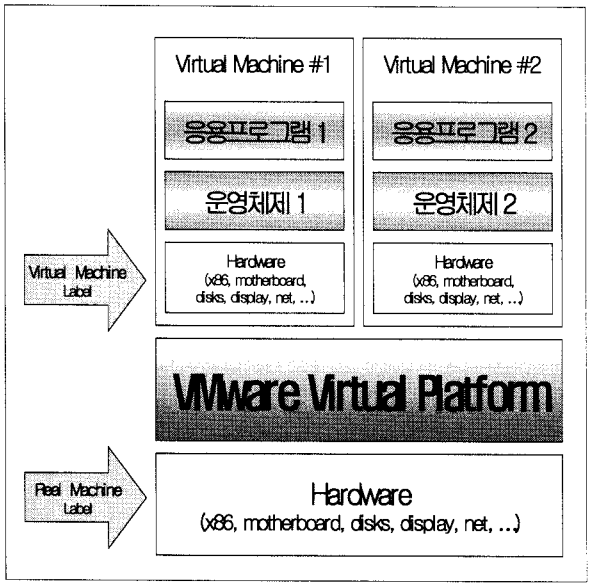
VMware는 실존하는 하드웨어 정보들을 공유하여 가상으로 하드웨어의 층을 만든다. 다시 말해 시스템의 하드웨어를 공유함으로 해서 현재 사용중인 운영체제에 영향을 끼치지 않은 상태에서 다른 운영체제를 실행하는 것이다. 이러한 원리로 작동하기 때문에 다른 운영체제에 오류가 생기더라도 창 내에서만 멈추게 되므로 안정성도 뛰어나다. VMware는 지금까지의 컴퓨팅 환경에서 보다 유연하고 생산성 있는 컴퓨팅 환경을 사용자에게 제공한다.

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 OS를 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[14].

다음의 그림 3에서 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 OS 환경과 그 응용 프로그램들을 싱글 컴퓨터 기반에서 수행이 가능하도록 해주는 것이다. VMware 가상 플랫폼 상에서 VMware 가상 머신 기반의 응용 프로그램들이 치명적인 오류를 일으켰을 경우에도 이것은 가상 머신 바깥의 Real 머신에는 전혀 영향을 주지 않는다[14].

바이러스 테스트 시뮬레이션을 구현하는데 있어서 VMware의 가장 큰 장점이라고 할 수 있는 것은 VMware는 PC의 CPU 타입을 제거함으로 해서 PC의 하드웨어와 완전하게 모의 테스트를 할 수 있다는 것이다. 또한 한 대의 PC에서 다른 하드웨어나 프로세서를 위한 화상 파일을 쉽게 복사하는 것이 가능하다는 것과 바이러스 테스트 시뮬레이션을 하게된 운영체제가 시뮬레이션을 하고 난 후에도 조금의 오류나 시스템 변경 사항이 나타나지 않는다는 것이다.



(그림 3) VMware의 구동 원리

이러한 VMware를 이용한 Simulator는 다른 PC의 하드웨어 위에서 필요한 만큼 자주 사용하고 실행 될 수 있다. 그러나 VMware는 아직까지는 Simulator이고, 일반적인 PC의 속도보다는 현저하게 느리다. 따라서 VdMware를 이용한 Simulator는 시간이 많이 걸리는 테스트보다는 바이러스 테스트 같은 작고 시간이 많이 걸리지 않는 테스트 시뮬레이션에 도움이 된다.

VMware를 이용한 Simulator PC는 평상시에는 VMware를 'Suspend' 모드로 하여 보통 일반적인 PC처럼 활용 할 수 있으며, 언제라도 Simulator 상태로 되돌려 실행시킬 수 있다. PC에서 시뮬레이션을 하고 난 후의 정보는 디스크에 기록되고 사용자는 그 기록을 보며 활용할 수 있다[6][12].

3.2 VMware를 이용한 Virus Test bed 사례

VMawre를 이용한 바이러스 테스트 베드 시스템은 아직 많이 활용되고 있지는 않다. 그러나 다음에 나오는 Symantec에서 활용하고 있는 Solution의 사례를 보면 알 수 있듯이 막대한 비용의 절감을 포함하여 시간과 공간의 절약, 기계설비의 절감 등 많은 이점이 발생한다.

3.2.1 VMware Solutions - Symantec

Symantec은 anti-바이러스 기술과 인터넷 보안 기술에 있어 선두로 나서고 있는 기업이다. 이러한 기술력을 바탕으로 개인 및 각 기업, 정부조직, 교육시설 등 약 5천만의 고객들에게 환경에 맞는 anti-바이러스 제품과 네트워크 보안 솔루션을 제공하고 있다. Symantec에서는 완벽한 제품을 생산하기 위한 과정중의 하나인 제품의 테스트 시스템에 너무 많은 기계설비와 비용이 필요하게 되었다.

또한 Symantec은 Symantec 내에서 사용하는 각 서버당 약 2만 여명이 넘는 클라이언트 사용자가 이용하는 등, 많은 제품의 사용자를 형성하고 있다. 이러한 대규모의 클라이언트 사용자를 충족시키기 위해 Symantec은 보다 더 큰 규모의 테스트 베드가 필요했고, 다양한 클라이언트 사용자의 운영체제 환경을 가상하여 필요한 시스템을 만들어 테스트를 해야만 했다.

특히 Symantec은 anti-바이러스 제품의 테스트를 위하여 다양한 클라이언트 운영체제 환경의 내부와 외부 네트워크를 가상하는 테스트베드를 필요로 했다. 이러한 테스트 베드 시스템의 운영 체계 및 다양한 환경에서의 바이러스 테스트는 더 많은 바이러스 테스트의 결과 값과 anti-바이러스 제품의 질을 높이는데 기여를 할 수 있기 때문이다.

Symantec에서 VMware 프로그램은 이러한 바이러스 테스트 베드 시스템 환경을 쉽게 만들어 주었다. 이 가상 시스템 환경은 Windows NT 운영체제에 VMware를 설치하고 Windows 9x, Windows NT, Windows 2000, Windows 3.1 등의 운영체제 환경을 만들어 각각의 환경에 대해 다른 노드로 네트워크로 연결되며, 다른 인터넷 주소를 갖도록 만들었다.

이러한 가상 시스템의 사용은 Symantec이 실제 기계설비에 투자한 것과 같은 결과를 가져왔다. 물리적인 호스트에 VMware를 사용하면 3개의 기계설비를 추가한 것과 같았고, 100개의 호스트로 300대의 클라이언트를 얻을 수 있었다. 또한 200대의 클라이언트는 가상이기 때문에 공간과 에너지의 절약을 가져올 수 있었다.

Symantec에서 VMware를 사용한 바이러스 테스트 베드 시스템 사용의 가장 큰 이득은 600대의 테스트 베드 시스템을 만드는데 하루가 걸리지 않

을 정도의 시간의 절약과, 기계설비의 라이센스 비용과 아웃소싱 비용의 절감 등, 비용을 대폭적으로 줄여주었다는 것이다[20].

4. 결 론

1990년대 중반 이후 바이러스는 계속 폭발적으로 증가하고 있다. 또한 매크로 바이러스와 윈도우 바이러스 그리고 최근의 웜 바이러스처럼 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. 뿐만 아니라 이러한 바이러스들이 여러 가지 다양한 운영체제를 통하여 활동하기 때문에 바이러스 테스트 베드 시스템을 구축하는 환경에 더욱더 어려움을 주고 있다.

이에 본 고에서는 최근의 복잡해지고 암호화 기술을 사용하며 해킹 기술까지도 접목한 바이러스들이 여러 가지 다양한 운영체제에서 활동하고 있다는 현실에 주목하여 먼저 바이러스 테스트 베드 시스템의 필요성과 기본 원리에 대해서 알아보았다. 이를 바탕으로 멀티플 운영체제 환경을 가능하게 해주는 응용 프로그램인 VMware를 이용한 바이러스 테스트 베드 시스템의 장점과 이러한 시스템을 사용하고 있는 사례를 확인해 보았다. 따라서 VMware를 이용한 바이러스 테스트 베드 시스템이 다양한 운영체제 환경에서의 바이러스 테스트를 하는데 효과적으로 사용될 수 있을 것으로 기대된다.

참 고 문 헌

[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] Rado, "Viruses on Unix systems", LinuxTicker, http://lionuxticker.com/article/459.html, 2000. 3.

[4] 한국정보보호진흥원, "바이러스 및 해킹대응관리 시스템 개발", 정보통신부 - 연구보고서, 2000.

[5] 아주대학교, "학습형 바이러스 면역 기본 시스템 개발", 한국정보보호진흥원 - 연구보고서, 2000.

[6] Ludwig, MarkA, PaperBack, "Computer Virus Super Technology. 1996. 3

[7] Helenius Marko (1995a), "Automatic and Controlled Virus Code Execution System", In proceedings of the Eicar 1995 Conference held in Zurich, Switzerland 27.-29.11 1995, hosted by CIMA AG. pp. T3 13-21. 1998

[8] Annual EICAR Conference, http://conference.eicar.org/pastconferences/1998/other/autodbl.pdf, 2001.

[9] Ward, Brian, "Book of VMware", 2001.

[10] Aybrey-Jones David. 'Automatic Testing of Memory Resident Scanners', In the proceedings of the Fifth International Virus Bulletin Conference. pp. 125-132. 1995.

[11] Applications of Informatics in Arts and Science University of Hamburg - CSDepartment, http://agn-www.informatik.uni-hamburg.de/

[12] E-Testing Labs, http://www.etest-inglabs.com/main/reports/emccelerra.-pdf, 2001. 7.

[13] Bontchev Vesselin, "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. 1996
[14] Helenius Marko, "Antivirus Scanner Analysis Based on Joe Well's List of PC viruses in the wild 7/1997"

김 홍 일 (Hongil Kim)

- 정회원
- 1986년 2월 : 홍익대학교 전산계산학과 (이공학사)
- 1989년 2월 : 인하대학교 전산계산학과 (이공석사)
- 2000년 2월 : 홍익대학교 컴퓨터공학과 (이공박사)
- 1994년 3월 ~ 현재 : 대진대학교 컴퓨터공학과 부교수
- 관심분야: 인터넷응용, P2P, IPv6

논문접수일 : 2009년 1월 5일
논문수정일 : 2009년 2월 17일
게재확정일 ; 2009년 2월 25일