2002 년 한국정보처리학회 춘계학술발표논문집 제 9 권 제 1 호

# 리눅스 기반 바이러스 VDPM 시스템 개발

정훈호, 이명옥, 이은미
동신대학교 전기전자정보통신공학부
e-mail : mikelee@white.dongshinu.ac.kr

# Linux-Based Virus Detection, Protection & Management System

Hoon-Ho Jung, Mike Myung-Ok Lee and Eun-Mi Lee
School of EE, Information & Communication Engineering, Dongshin University

**요 약**

이 논문에서는 리눅스 기반 VDPM(Virus Detection Protection Management)시스템을 제안하고 개발한 응용 SW 로 감지, 차단 및 관리 방법을 제시한다. 개발된 VDPM 시스템은 신종 바이러스까지 탐지하는 모든 종류의 바이러스 탐지(VDPM_hawkeye) 모듈, Virus 체크하는 감시 및 Virus 체크후 교정, 제거하는 방지(VDPM_medic)모듈, DB 를 update 하는 기능을 가지는 관리(VDPM_manager)모듈과 원격 DB 관리 및 Virus 결과 보고 기능(VDPM_reporter) 모듈로 되어 있으며 지능적인 Virus 방지 시스템을 구현 하였다.

## 1. 서 론

리눅스는 핀란드 헬싱키대학의 리누즈 토발즈라는 학생이 취미삼아 처음으로 만들었던 운영체제 시스템으로서 GPL(General Public License)에 따라 개발되고 있으며 소스코드는 누구나 자유롭게 사용할 수 있다. 그러나 자유로운 것만큼 외부로부터 쉽게 공격을 받을 수 있고 이러한 문제는 앞으로 네트워크를 통한 인터넷 사용자가 엄청나게 증가함에 따라 그리고 인터넷 접속 가능한 차세대 이동통신 휴대폰이나 PC 나 무선 단말기에도 해킹(혹은 바이러스)라는 공격으로 더욱 빠르고 전세계적인 넓은 지역으로 전파되는 위험성이 증가될 전망이다. 한국정보보호센터에서 처리한 해킹 사고를 운영체제별로 분류가 가능한 경우 약 절반 이상의 시스템이 리눅스였으며, 리눅스 계열에서는 90% 이상이 리눅스에서 발생한 해킹 사고였다. 국내외적으로 많은 피해를 입었던 DDoS(Distributed Denial of Service)도 결국엔 일부 취약한 기계들이 사전에 해킹을 당해 이용된 것이다. 또한 최근 급속도로 증가하고 있는 인터넷 쇼핑몰 및 개인 운영 서버들이 리눅스 플랫폼 기반으로 만들어지고 있음을 볼 때, 리눅스 보안 관리에 대한 노력이 더욱 절실하게 요구되고 있다. 이 논문에서는 리눅스 기반 VDPM(Virus Detection Protection Management)시스템을 제안하고 개발한 응용 SW 로 감지, 차단 및 관리 방법을 제시한다.

## 2. 리눅스 보안 및 VDPM

리눅스가 보안상 문제가 되는 이유는 먼저 OS 내부 구조가 완전히 공개되어 있어 리눅스 커널(kernel)은 누구나 소스를 구해서 분석할 수 있다는 점이다. 또한 가격이 저렴하고, 성능이 뛰어나고 사용방법이 쉬워짐에 따라 리눅스 사용자의 급속한 증가를 가져왔다는 점이다. 그러한 장점아닌 단점으로 인하여 보안 구멍이 발견되면 그것을 이용한 공격 도구는 리눅스용이 가장 먼저 만들어 진다는 점이다[1]. 리눅스 혹은 유닉스 바이러스는 기존 윈도우나 DOS 형태의 바이러스와 구현 방법이 유사하며 커널(Kernel)의 함수를 이용하여 감염되는 경우가 많으며 관리자 권한이 있는 경우는 원할하게 작동되고 있다. 커널 관련 보안 기능은 IP 패킷 수준에서 걸러내기(패킷 필터링), 매스커레이딩, NAT 지원과 리눅스 커널의 버전업에 따른 패킷필터링 툴의 변화등이 있다. 체계적인 정책과 룰을 통한 강력한 방화벽 구축 및 보안 강화가 가능하다. 이러한 리눅스 바이러스 유형은 트로이 목마, 백도어 공격이 있는데, 기존의 프로그램이나 스크립트가 실행될 때 허가되지 않은 행위를 할 수 있는 숨겨진 프로그램이나 쉘 스크립트를 이용한 형태와 리눅스 시스템 자체의 버그나 설정상 취약점을 이용한 해킹 기법인 웜(Worm) 형태가 있는데 이는 다른 시스템에는 직접적

인 영향을 미치지 않고 기억장소 내에서 자기 자신을 계속적으로 증식하는 프로그램으로 네트워크를 통해 대규모로 자동 전파된다는 점이다[1][2]. 통신정보보호기술의 한 부류인 서비스 차원인 안티바이러스와 침입차단시스템의 본 연구는 리눅스 응용으로 VDPM 의 절차는 네트워크를 통한 바이러스를 탐지한 후에 네트워크 데몬에 의한 패킷 흐름도를 감시하여 DB 에 저장하여 관리하는 시스템(그림 1)이다.

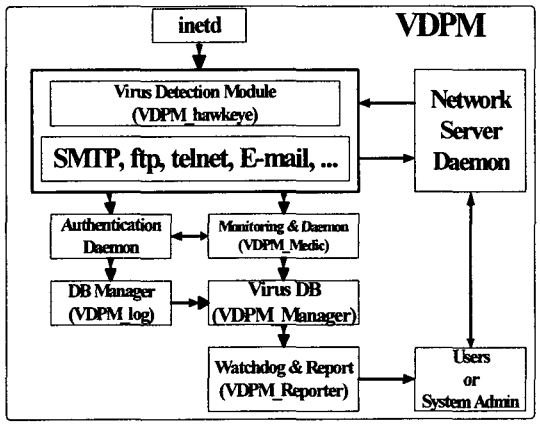



그림 1. VDPM 시스템 구성도

현재 정보보호 기술은 미국이 전세계의 50% 이상을 차지하고 있으며 미국 제품 시장은 본 연구에서 주장하는 바이러스 탐지, 침입차단시스템 및 인증 분야가 주도하고 있다[3]. 이러한 제품 중의 하나가 본 연구가 제안하는 VDPM 시스템이다. 부분별로 요약하면 아래와 같다.

• Monitoring System Calls : File open, Write, delete 및 Append 와 같은 File 관련 System 을 Call 을 감시하고 문제점을 log 파일로 기록을 남기도록 설계

• Virus 판단 여부 : Virus pattern 을 Normalize 시켜 수치화된 결과값에 근거하여 Virus 여부를 판단

• Virus 차단 방법 : 모든 ELF 파일 체크하여 DB 로 관리하고, 정기적인 점검이 필요하다. 특히 ELF 파일들이 변경될 때마다 DB Manager 에게 알려주는 방법: - Linux : Firewall 로 모든 외부 traffic 을 차단하든지 혹은 Linux Router 를 사용하여 ICMP 기반이나 Spoofing 으로 차단

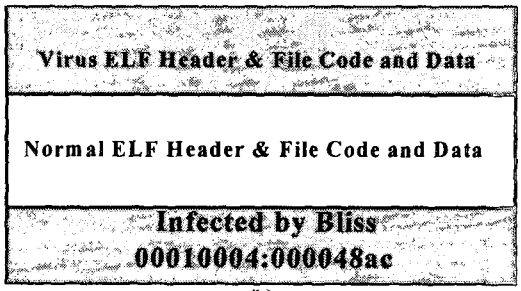
## 3. 바이러스 방지 모듈 개발

### 3.1. 바이러스 탐지 및 차단 기술 구현

유닉스 바이러스는 Linux/Staog, Bliss 등과 같은 직접적으로 유닉스의 ELF(Executable and Linking Format) 실행파일에 삽입되는 일반적인 바이러스 형태와 Morris Worm, ADM Worm, Millennium Internet Worm 등과 같이 직접적으로 파일이나 시스템을 감염시키지 않고 유닉스 시스템 자체의 버그나 설정상의 취약성을 이용한 해킹 기법을 이용하여 네트워크를 통해 대규모로 자동 전파할 수 있는 웜의 형태로 나누어 볼 수 있다. 컴퓨터 바이러스 제작자들은 자신이 만든 바이러스를 안티바이러스 프로그램(Anti-virus Program)들의 진단방법을 피하면서 감염되도록 여러 가지 기법들을 사용하였다. 초기에는 감염패턴이 일정한 바이러스들이 등장하다가 백신들이 쉽게 진단모듈을 만들어내자 이를 다시 암호화한 바이러스 그리고 나중에는 암호화한 내용도 알아보기 힘들게 만든 다형성 바이러스가 나타나게 되었다. 유닉스 시스템의 실행파일인 ELF(Executable and Linking Format) 파일을 실제적으로 감염시키는 유형으로 임의의 악의적인 코드(실행 코드나 텍스트)를 실행파일에 삽입 할 수 있기 때문에 그 전파 가능성은 매우 높다고 할 수 있다[4]. 악의적인 바이러스 제작자가 PC 바이러스의 유포 방법처럼 유닉스 시스템의 주요 프로그램이나 패치 등에 바이러스를 감염시켜 배포하게 된다면 현재까지는 유닉스 서버 용 백신에 대한 인식이 거의 없기 때문에 그 확산과 피해는 매우 클 것으로 예상된다. 예로 Linux/Staog, Linux/Bliss, VLP 등의 바이러스가 있으며, 계속적으로 새로운 유형이 연구되고 있다. 리눅스 ELF 파일 변형 바이러스 탐지 기술 개발은 1997 년에 발견된 리눅스 이진 바이러스인 Bliss 는 상대적으로 단순한 바이러스이다(그림 2). 즉, 원본 파일에 바이러스의 헤더와 코드를 덧붙이는 바이러스이다. 감염된 파일들은 2 개의 ELF 헤더를 갖게 된다. 사실 Bliss 바이러스를 받을수 없어 실제 Bliss 바이러스 파일을 생성하는 프로그램을 개발하였다. 생성된 프로그램 결과로 여러 번 시스템을 재설치하는 과정을 거쳤고 이로 인한 결과를 분석하면, 첫 번째는 바이러스로부터, 두 번째는 정상적인 파일로부터 갖게 된다. 그래서, 감염된 파일의 두 번째 헤더(원본 ELF 헤더)는 오프셋 48AC(hex)에서 시작한다.

```
#include "stdio.h"
main()
{
FILE *input,*output;
char i,j,k;
        input=fopen("bliss.gz.swapped-bits","rb");
        output=fopen("bliss.gz","wb");
        while(1) {
                fread(&i,1,1,input);
                if(feof(input)) break;
                printf("%i\n",i);
        ……
                k=k & 0xAA;
                printf("%i %i\n",j,k);
                i=j|k;
                printf("%i\n",i);
                fwrite(&i,1,1,output);
        }
}
```

(a)




(b)

그림 2. (a)바이러스 생성 프로그램, (b)프로그램에 의한 리눅스 이진 바이러스 Bliss 바이러스의 형태.

그러므로 진단과 치료가 비교적 쉬운 바이러스라고 할 수 있다. 바이러스 탐색 방법으로는 전체 검색법과 특정위치 검색법을 사용하고 있다. 전체 바이러스 탐색 기법은 파일 전체를 바이러스 패턴에 의해서 검색하기 때문에 바이러스 탐지 효율이 우수 할 수는 있으나 단점으로는 검색 속도가 느리고 또 바이러스가 아닌 파일을 바이러스로 오인 할 수 있다. 특정위치 검색법은 전체 검색기법에 비해 속도가 빠르다는 장점은 있지만 알려지지 않은 바이러스를 찾아내는 데는 다소 어려움이 있다. 본 연구에서 수행하고자 하는 탐색 방법은 알려진 바이러스에 대해서 ELF 파일 바이러스를 탐지하고자 한다. 그 방법으로 전체 검색 기법을 사용하였고 탐지하고자 하는 파일을 선택한 다음 그 파일이 예를들어 ELF 파일인지를 검사한다. ELF 파일이면 그 파일의 최초로 수행되는 위치부터서 바이러스 패턴을 검사하여 바이러스를 탐지하는 방법을 사용하였다. 그 내용을 그림 3 에서 도식으로 표현하였다. 그러나 모든 종류의 바이러스를 각 바이러스의 패턴에 따라 알고리즘을 개발하였다.

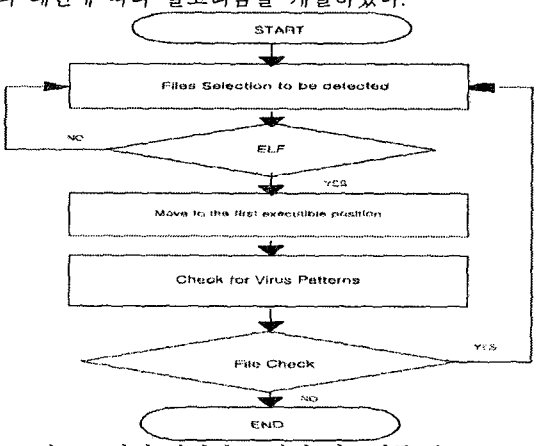



그림 3. 파일 바이러스 탐지 알고리즘(예: ELF)[4]

자체 개발한 첫 1.0 버전인 VDPM(바이러스 탐지, 차단, 관리 및 보고) 모듈를 정리하면 아래와 같다.

• 탐지(VDPM_hawkeye): 탐지 파일들은 자체 Virus, Stealth/invisible Virus, Window Virus/Unix/Os2 Virus, Java Applet 신종 Virus, Macro Virus(Word, ppt, Excel 등) 및 네트워크 웜 및 Trojan Virus(그림 4).

• 강시(VDPM_medic): Virus 체크 및 교정, 제거 및 DB 를 update 하는 기능(그림 5 와 6).

• 관리(VDPM_manager): Virus 의 DB 관리(그림 7)

• 관리(VDPM_reporter): Virus 결과 보고 기능(그림 7)

개발된 VDPM 시스템을 테스트한 결과는 아래와 같으며 위 모듈 기능을 실행한 결과들이다.

(1)슈퍼유저로 실행해야하면 사용자의 홈디렉토리의 내용을 검사한다. 만약 압축된 파일이 있으면 압축을 풀어 실행한다.

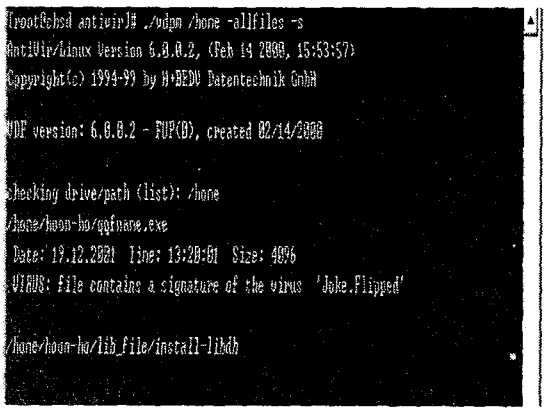



그림 4. VDPM 초기 실행화면(VDPM_hawkeye)

--> 실행화면은 도스상의 바이러스 탐지프로그램과 유사하고 앞으로 GUI 로구현할 계획이다.

(2) 실행화면 결과

그림 5. 검사 진행중인 모습(VDPM_medic)

--> 검사 화면은 윈도우 백신의 도스모드 모습과 유사하게 Linux 상에서 구현하였다.

(3) 바이러스 탐지 요약 및 진단 보고

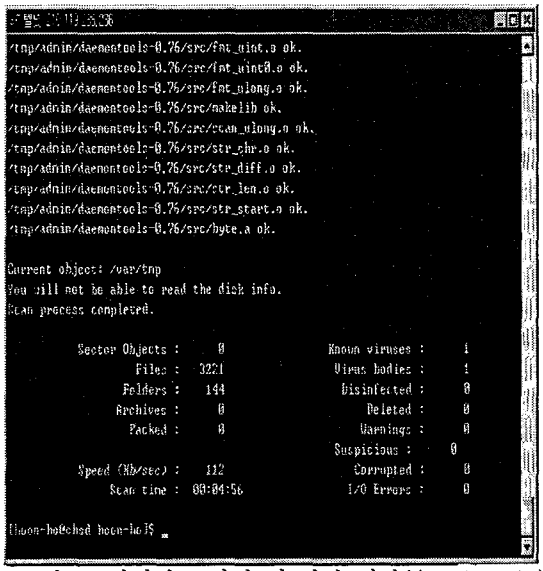



그림 6. 바이러스 탐지 및 차단 결과((VDPM_medic)

위 그림 6 은 개발한 VDPM_medic 에 의한 결과를 나타내며 검사 종료 후 결과를 보여준다. 바이러스 탐지을 실행하기 위하여 그림 2(a)에서 생성된 바이러스 파일을 임의 디렉토리에 조심스럽게 바이러스가 걸린 파일을 넣고 실제 실행을 했을 때 위와 같이 바이러스가 탐지되었고 이를 진단할 수 있는 연구는 아직 진행 중이다.

(5) 바이러스 관리 및 결과 보고

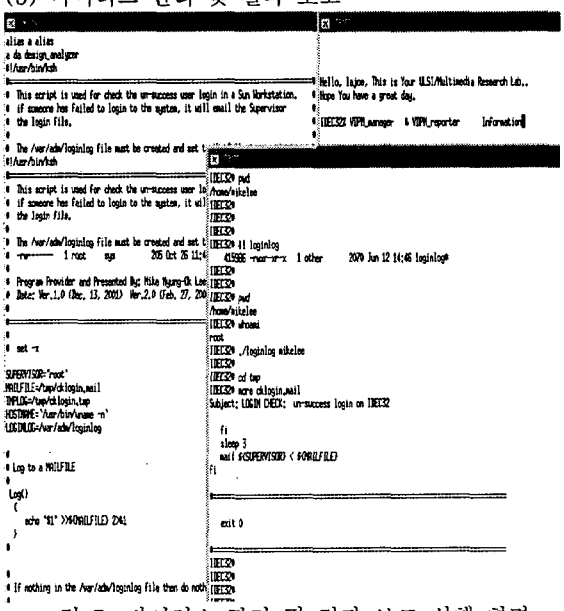


그림 7. 바이러스 관리 및 결과 보고 실행 화면

위 그림 7 은 한 유저가 부당하게 침투하여 바이러스를 감염 하였을 경우 수퍼유저나 시스템 관리자에게 알려주는 화면이고 아직 완숙한 단계는 아니지만 탐지한 바이러스를 관리하고 결과를 보고하는 과정이다.

## 4. 맺음말 및 대책

리눅스 기반 바이러스 방지 모듈 개발의 VDPM 이라는 체계적인 바이러스 탐지 및 차단 기술 구현하였고 이러한 바이러스의 문제점과 그 대책을 정리하였다.

- 항상 최신의 프로그램을 사용: 보안상 문제가 있는 프로그램 패치 및 업데이트하는 것이고,
- 필요 없는 프로그램은 사용하지 않음: 보안상 문제가 되는 프로그램과 사용하지 않는 프로그램은 시스템에서 제거하고,
- 보안 관련 공지를 자주 확인하고,
- 시스템 관련 감시를 이용하며,
- 자신의 컴퓨터를 해킹해도 해커가 얻을 것이 없다는 생각을 하지 않음: 리눅스관리자의 mind 가 가장 중요하다는 것이다.

문제점으로서는 국가적인 기술개발 지원(예: 보안이나 바이러스 예방에 대한 투자, 컴퓨터 바이러스 백신개발에 대한 투자, 악성 SW 에 대한 국가적 차원에서 목표를 분명히 설정하고 체계적인 대처방안 강구 등)이 부족하다. 바이러스 문제점을 대응하기 위하여 연구개발 전문가 인력 양성하는 것이고 악성 SW 대응 전문센터를 추진할 필요가 있다. 물론 바이러스 침입에 대한 안전하고 신뢰성 있는 사이버공간을 구축하기 위해서 개인· 기관별 해킹 · 바이러스 예방 및 대응의식 고취시키고, 정부정책에 대한 적극적인 지원이 필요하며 유관기관의 예방 경보 및 대응활동 강화에 대한 대비책 마련이 시급하다고 본다.

## 참고문헌

[1] 최홍진, "[리눅스 시큐리티 강좌 1]리눅스의 시스템 보안 및 취약점", *Onthenet*, 2001 년 4 월호.

[2] 김판구, "컴퓨터 바이러스의 이해와 대응 방안", *IDEC 보안알고리즘 및 VLSI 설계 강좌*, pp. 59-99, 2001 년 8 월.

[3] MACRO Technology, "MACRO Security Report", 2001 년 5 월, 제 1 호.

[4] 김판구, "리눅스 상에서의 ELF 파일 바이러스 진단 및 차단 시스템", 리눅스 보안 연구 센터 Workshop 프로그램, 전남대학교, 2001 년 5 월.

[5]http://www.certcc.or.kr/paper/incident_note/2001/in2001_010.html

[6]http://www.certcc.or.kr/paper/incident_note/2001/in2001_009.html

[7]http://www.cisco.com/warp/public/63/nbar_acl_codered.shtml#8