리눅스 사용자를 위한 개인용 방화벽 설계 및 구현

이성미° 권문상 현상원 조유근
서울대학교 컴퓨터공학부
{smlee°, kmscom, swhyun, cho}@ssrnet.snu.ac.kr

Design and Implementation of Linux PC Firewall

Sungmi Lee° Moonsang Kwon Sangweon Hyun Yookun Cho
Dept. of Computer Science and Engineering, Seoul National University

요 약

최근 개인용 컴퓨터 사용자들의 인터넷 사용이 빈번해지면서, 개인용 컴퓨터 시스템에 대한 다양한 네트워크 공격의 피해가 늘고 있다. 이에 개인용 방화벽이 개발되었으나 현재 대부분이 윈도우 시스템에 기반하고 있다. 하지만 최근 리눅스 시스템이 개인용 컴퓨터의 운영체제로 많이 사용되고 있고, 피해 사례가 상당히 많음에도 불구하고 아직까지 리눅스 기반 개인용 방화벽은 전무한 실정이다. 본 논문에서는 기존의 리눅스 기반 네트워크 방화벽에서 제공하는 기능들 외에 동적 정책 설정 기능과 악의적인 네트워크 프로그램 실행의 제어 기능들을 추가적으로 포함하는 리눅스 기반 개인용 방화벽을 설계하고 구현하였다.

1. 서 론

최근 빈번하게 발생되고 있는 네트워크를 통한 공격에는 악의적인 프로그램을 이용한 공격, 바이러스 공격, 서비스 거부 공격 등이 있다. 그런데 2002년 CERT의 피해 시스템 분석 결과에 따르면 네트워크 공격 중 트로이목마 또는 백도어 같은 악의적인 프로그램을 이용한 공격이 가장 많았다고 한다[1]. 이 같은 공격은 사용자 몰래 시스템에 설치되어 시스템의 중요한 정보를 빼내거나 변경, 삭제하는 형태 또는 재 침입을 용이하게 하는 형태를 띤다[2][3].

현재 네트워크 공격에 대처하기 위해 리눅스에서 가장 많이 사용되는 것이 네트워크 방화벽이다. 네트워크 방화벽은 네트워크 인터페이스를 통해 들어오고 나가는 패킷들의 헤더를 검사하여 미리 정해둔 정책에 따라 허용하거나 차단하는 기능을 수행한다[3]. 그러나 네트워크 방화벽은 개인 컴퓨터 시스템을 완벽하게 보호하기에는 많은 문제점을 가지고 있다. 첫째, 공격을 차단하기 위한 정책이라는 것이 패킷의 헤더에 포함된 IP 주소나 포트를 확인하는 수준에 머물기 때문에 트로이목마 또는 백도어 같은 악의적인 프로그램을 통한 공격 또는 바이러스를 이용한 공격을 막는 데는 한계가 있다.[4][5]. 둘째, 정책의 적용이 정적이기 때문에 새로운 형태의 공격에 능동적으로 대처하기 어렵다. 네트워크 방화벽이 갖는 이 같은 문제점들을 보완하기 위해 나온 것이 개인용 방화벽이다.

개인용 방화벽은 개인 컴퓨터에 설치되어 네트워크 인터페이스를 통해 들어오고 나가는 모든 패킷에 대해 보안 관련 작업을 수행한다. 패킷 필터링, 정적인 정책 설정과 같은 네트워크 방화벽에서도 제공되는 기능 외에 개인용 방화벽에서만 제공되는 주요 기능에는 다음과 같은 것들이 있다. 첫째는 패킷의 도착 시점에 질의를 통해 해당 패킷의 허용 여부를 사용자에게 동적으로 확인받을 수 있다는 것이다. 따라서 새로운 공격에 대한 사용자의 능동적인 대처가 가능하다. 둘째는 네트워크 프로그램의 실행 시 프로그램의 변경 여부를 확인함으로서 트로이목마나 백도어 같은 악의적인 프로그램을 이용한 공격에 효과적으로 대처할 수 있다[1][2].

이처럼 개인용 방화벽이 여러 장점들을 가지고 있음에도 불구하고 아직까지 리눅스 기반의 개인용 방화벽은 전무한 실정이다. 이에 본 논문에서는 기존의 리눅스 시스템에서 제공하는 네트워크 방화벽의 패킷 필터링 기능 외에 추가적으로 동적 정책 설정 기능과 악의적인 네트워크 프로그램 실행의 제어 기능들을 포함하는 리눅스 기반 개인용 방화벽을 설계하고 구현하였다. 제안하는 개인용 방화벽은 커널 모듈 형태로 구현되었으며, X-window를 사용하는 사용자를 위한 그래픽 모드 인터페이스와 원격 사용자를 위한 텍스트 모드 인터페이스 모두가 가능하도록 구현했다.

2. 개인용 방화벽 설계

본 논문에서 제안하는 리눅스 기반 개인용 방화벽은 윈도우 기반의 개인용 방화벽 중 하나인 Tiny Personal Firewall 2.0 버전을 모델로 설계되었으며, 크게 커널 영역과 사용자 영역 프로그램들로 나누어져 설계되었다. 방화벽 제어 유저프로그램은 사용자 영역에, 개인용 방화벽의 핵심 모듈인 인증 모듈, 패킷 필터링 모듈, 무결성 검사 모듈, 방화벽 핵심 모듈은 커널 영역에 존재한다. 포함된 모듈들의 전체적인 구조는 다음 장의 그림1과 같다.

2.1 인증 모듈

인증 모듈은 방화벽 제어 유저 프로그램과 함께 동작하며 해당 사용자가 방화벽 제어 권한을 갖고 있는지를 확인하고 등록하는 기능을 수행한다. 먼저 방화벽 제어 유저 프로그램이 사용자가 입력한 패스워드를 검사하고, 일치하면 인증 모듈에게 해당 사용자가 인증 받았음을 알리고 해당 사용자를 등록하도록 지시한다.

2.2 프로그램 무결성 검사 모듈

무결성 검사 모듈은 방화벽 핵심 모듈, MD5 계산 모듈과 함께 동작하며 실행되고 있는 네트워크 프로그램의 변경 여부를 검사한다. 검사 과정은 다음과 같다. 먼저

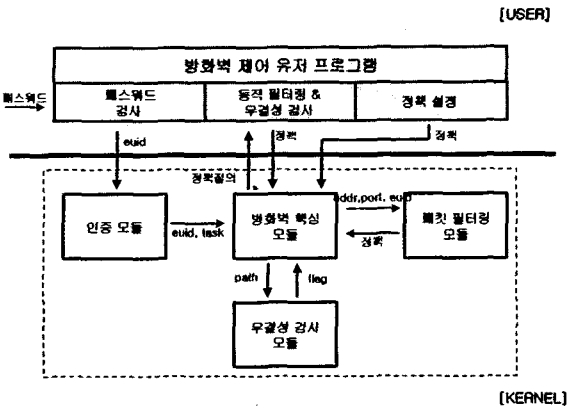

그림 1. 개인용 방화벽 전체 구조

네트워크 프로그램이 실행될 때 방화벽 핵심 모듈이 그 프로그램의 절대 경로를 무결성 검사 모듈로 전달한다. 무결성 검사 모듈은 해당 프로그램의 MD5 값을 MD5 계산 모듈을 이용해서 계산하고, 그 결과 값과 기존의 무결성 값 리스트의 모든 값들을 비교한다. 이전에 실행된 적이 있는 프로그램의 경우 값이 같으면 EQUAL_HASH를, 다를 경우엔 DIFF_HASH를 Flag에 전달한다. 이전에 실행된 적이 없는 프로그램의 경우는 계산된 MD5 값이 무결성 값 리스트에 존재하지 않을 것이므로 NOT_EXIST가 Flag로 전달된다. 그리고 그 Flag가 방화벽 핵심 모듈로 전달된다.

2.3 패킷 필터링 모듈

패킷 필터링 모듈은 네트워크 프로그램의 실행을 요청하는 패킷의 정보(패킷 헤더 정보, 유저 아이디)를 이용하여 해당 사용자가 정한 기존 정책이 존재하는지를 확인하고 해당 정책을 방화벽 핵심 모듈에게 전달한다. 이때 Filter_Policy라는 자료구조가 정의되어 사용되는데, 패킷의 방향을 나타내는 direction 필드, 사용되는 프로토콜을 나타내는 TYPE 필드, 포트 번호를 저장하는 port 필드, 해당 패킷에 대한 정책을 나타내는 rule 필드 등이 포함된다. 방화벽 핵심 모듈에서 패킷 필터링 모듈로 Filter_Policy 자료 구조가 전달될 때는 rule 필드를 제외한 다른 모든 필드들이 채워진 상태로 전달되며, 패킷 필터링 모듈이 해당 패킷에 대한 정책 정보를 rule 필드에 기록하게 된다. rule 필드가 가질 수 있는 가능한 값들에는 정책 없음(NO_POLICY), 패킷 차단(DENY), 패킷 허용(ACCEPT)이 있다.

2.4 방화벽 핵심 모듈

본 논문에서 제안하는 개인용 방화벽의 가장 핵심이 되는 모듈로서 커널 영역과 사용자 영역 사이의 모든 통신을 수행하며 개인용 방화벽 기능 제공을 위한 총괄적인 역할을 수행한다. 네트워크 프로그램의 실행 요청이 있을 경우 프로그램의 실행을 지연하고 무결성 검사와 패킷 필터링을 수행한다. 패킷 필터링 수행 시 사용자의 유저 아이디를 이용하여 해당 사용자가 정한 정책을 리턴하는 것이 가능하므로 여러 사용자가 하나의 시스템을 각자의 시스템처럼 사용하는 것이 가능하다.

무결성 검사와 리턴된 패킷 필터링 결과를 바탕으로 동적인 정책 질의 여부, 정책 수정/삭제/추가 여부와 무결성 값 수정/삭제/추가 여부를 결정한다. 이 외에도 정적인 정책 설정 또한 가능한데, 방화벽 제어 유저 프로그램의 정책 설정 부분에서 정적으로 설정된 정책이 방화벽 핵심 모듈을 거쳐 패킷 필터링 모듈로 전달되어 패킷 정책 리스트에 추가될 수 있다.

2.4 방화벽 제어 유저 프로그램

방화벽 제어 유저 프로그램은 패스워드 검사 부분, 동적 정책 설정 및 네트워크 프로그램 실행 결정 부분, 정적 정책 설정 부분으로 구성된다. 패스워드 검사 부분은 패스워드를 확인하여 인증 모듈에게 인증 받은 사용자임을 알리며, 정적 정책 설정 부분은 사용자가 정적으로 정책을 설정하고자 할 때 사용되는 부분이다. 마지막으로 동적 정책 검사 및 네트워크 프로그램 실행 결정 부분은 방화벽 핵심 모듈로부터 제공받는 패킷 허용 여부와 네트워크 프로그램의 변경 정보를 확인하여 동적으로 정책을 설정하는 기능을 제공한다.

3. 개인용 방화벽 구현

본 논문에서 제안하는 개인용 방화벽은 커널 모듈 형태로 구현되었다. 또 X-window를 사용하는 사용자를 위한 그래픽 모드 인터페이스와 원격 사용자를 위한 텍스트 모드 인터페이스 모두를 제공한다.

두 번째 절 "설계"를 통해 알 수 있듯이, 제안된 방화벽은 사용자 영역과 커널 영역 사이의 통신이 필요하다. 여기서는 시스템 콜 함수 호출과 시그널을 통해 이것을 구현했다.

3.1 사용자 영역 프로그램 구현

방화벽 제어 유저 프로그램의 핵심 알고리즘은 다음의 표1과 같다.

표 1. 방화벽 제어 유저 프로그램

```
main()
{
   패스워드 입력
   if (패스워드 일치)
      geteuid() 시스템 콜 호출
   else
      실행 종지
   SIGUSR1 시그널 핸들러를 등록
   write(특정 파일 touser , ,) 시스템 콜 호출
   if ( 정적으로 정책을 설정할 경우 )
      write ( , 정적 설정 정책, ) 시스템 콜 호출
   대기상태 유지
}
시그널 핸들러(SIGUSR1)
{
   특정파일 touser로부터 패킷 정보 확인하고 사용자에게 질의
   사용자는 해당 패킷에 대한 차단 혹은 허용 등의 정책을 동적으로 결정
   write( , 결정된 동적 정책 , ) 시스템 콜 호출
}
```

먼저 사용자로부터 패스워드를 입력받아 비교한 뒤 일치할 경우 geteuid 시스템 콜을 호출하여 인증 모듈에게 해당 사용자가 인증 받은 사용자임을 알린다. 다음으로 write 시스템 콜이 호출되는데 이는 커널에게 동적 질의

를 위해 패킷에 대한 정보를 넣을 저장 장소가 특정 파일 "touser"임을 알리기 위한 것이다. 커널에서 사용자 영역으로의 동적 질의가 필요한 경우 커널은 시그널을 보내게 되며 사용자 프로그램의 시그널 핸들러 함수가 특정 파일 "touser"로부터 해당 패킷에 대한 정보를 확인하여 사용자에게 질의한다. 그 패킷에 대해 사용자가 내린 결정은 write 시스템 콜을 통해 커널로 전달되고, 커널은 사용자가 내린 결정에 따라 그 패킷의 허용여부, 프로그램의 실행 여부를 결정한다.

3.2 인증 모듈 구현

인증 모듈에서 가장 핵심이 되는 부분은 인증된 사용자를 유저 리스트에 등록하는 일이다. geteuid 시스템 콜이 호출된 경우 인증 모듈은 이 시스템 콜을 후킹하게 되고, 현재 실행중인 태스크가 방화벽 제어 유저 프로그램일 경우 그 태스크의 euid를 인증된 유저 리스트에 추가한다. 유저 리스트는 auth_usr라는 구조체들로 구성되는데, 이 구조체는 사용자의 euid, 방화벽 제어 유저 프로그램의 태스크 포인터, file_path, write_addr 정보들을 포함한다. 태스크 포인터는 이후에 시그널을 보내고자 할 때 프로세스 아이디에 대한 정보를 얻기 위해 사용되며, file_path 정보는 커널에서 동적 정책 질의를 위해 사용자에게 보내는 패킷 정보를 넣을 파일 경로이고 write_addr은 사용자가 정한 정적 정책 정보를 커널에게 보내기 위해 사용되는 메모리의 주소이다.

3.3 무결성 검사 모듈 구현

무결성 검사 모듈에서는 MD5_res 구조체가 사용되는데, 이 구조체는 네트워크 프로그램의 절대 경로와 그 프로그램에 대해 계산된 MD5 값을 포함한다. 그래서 현재 실행중인 네트워크 프로그램의 MD5 값과 이전까지 만들어진 MD5_res 구조체로 이루어진 리스트에 포함된 MD5 값들을 비교해서 상황에 따라 적절한 Flag 값을 리턴한다.

3.4 패킷 필터링 모듈 구현

패킷 필터링 모듈에서 관리되는 정책들은 Filter_Policy 구조체들로 이루어진 리스트의 형태로 보관된다. 그래서 실행을 요청한 패킷에 대해 적용될 수 있는 정책을 찾기 위해 이 리스트를 검색하게 된다. 아래의 표2는 사용자가 정적으로 설정한 정책의 예이다.

표 2. 정책 설정 예

% F_ctrl -A	
Direction	: INCOMING
Protocol Type	: TCP
First_Addr	: 147.46.121.50
Last_Addr	: 147.46.121.100
Port	: 22
Rule	: DENY

이 경우 만약 어떤 사용자가 IP 주소 147.46.121.76, TCP 22번 포트로 접속을 시도한다면 접속이 차단될 것이다.

3.5 방화벽 핵심 모듈 구현

방화벽 핵심 모듈에서는 sys_socketcall, sys_write 두 개의 시스템 콜이 후킹된다. sys_socketcall은 해당 패킷에 대한 정보를 얻기 위해 후킹되는데, 이렇게 획득된 패킷 정보와 네트워크 프로그램의 경로 정보가 방화벽 핵심 함수의 매개변수로 이용된다. 호출된 방화벽 핵심 함수는 패킷의 허용 여부, 프로그램의 실행 여부를 결정하고 그 결과를 반환하게 되는데, 이 결과에 따라 정상적인 socketcall 시스템 콜을 호출하거나 수행을 중지하게 된다. sys_write 시스템 콜은 사용자가 정적 또는 동적으로 설정한 정책을 커널에게 보내기 위해 사용된다. 동적 또는 정적 정책 설정을 구분하는 데는 Filter_Policy의 flag 필드가 이용된다. flag 값이 동적 정책 설정인 경우에는 동적 질의에 대한 결과(rule 필드)값을 획득하는 작업이 수행되고, 정적 정책 설정인 경우는 그 정책을 정책 리스트에 추가하는 작업을 수행한다.

이외에 방화벽 핵심 모듈에 포함된 주요 루틴으로 dyn_filter가 있는데, 이 루틴은 사용자에게 동적 질의를 보내기 위해 사용된다. 루틴이 수행하는 작업은 다음과 같다. 해당 패킷의 정보, 무결성 검사 정보를 특정 파일 "touser"에 저장한 후 방화벽 제어 유저 프로그램에게 시그널을 보낸뒤 사용자의 응답이 있을 때까지 기다린다. 일정 시간동안 응답이 없을 경우에는 DENY 값이 반환된다.

4. 결 론

기존 리눅스 시스템에서 제공되고 있는 네트워크 방화벽을 이용해서 개인용 시스템을 보호하는 데는 한계가 있었다. 이와 같은 문제점을 해결하기 위해 본 논문에서는 동적 정책 설정 기능과 불법적으로 변조된 네트워크 프로그램 실행의 제어 기능들을 포함하는 리눅스 기반 개인용 방화벽을 설계하고 구현하였다. 제안된 방화벽은 커널 모듈 형태로 구현되었으며, 텍스트 모드 인터페이스와 그래픽 모드 인터페이스를 모두 지원한다.

참 고 문 헌

[1] 2002년 2월 침해 사고 접수 및 처리 현황. CERTCC-KR 한국 정보보호센터

[2] 트로이 목마와 백도어 분석 보고서 ,1999. CERTCC-KR 한국 정보보호센터

[3] D.B. Chapman and E.D. Zwicky, "Building Internet Firewalls", O'Reilly & Associates, Sebastopol. 1995

[4] Verwoerd, T. and Hunt, R., "Policy and Implementation of an Adaptive Firewall", Proceedings of the 10th IEEE International Conference on Networks, National University of Singapore, 27-30 August, 2002

[5] Kai Hwang, Muralidaran Gangadharan , "Micro-Firewalls for Dynamic Network Security with Distributed Intrusion Detection" IEEE International Symposium on Network Computing and Applications (NCA'01) October 8 - 10, 2001

[6] E.-G.Haffner, U. Roth, A. Heuer, Th. Engel, Ch. Meinel "Managing Distributed Personal Firewalls with Smart Data Servers" Proc. WebNet 2001, Orlando, (Florida, USA), 2001, pp. 466-471