Linux 운영체제에서 Packet Filtering 방식을 이용한 방화벽 시스템의 구현

*한상현, 안동언, 정성종
*학산정보산업고등학교, 전북대학교 컴퓨터공학과
e-mail : shyeonh@hitel.net, duan@chonbuk.ac.kr, sjchung@chonbuk.ac.kr

Implementation of Firewall System Using Packet Filtering Method in the Linux OS

*Sang Hyeon Han, Dong Un An, Seong Jong Chung
*Haksan Information&Industry High School,
Department of Computer Engineering Chonbuk National University

Abstract

Complying with highly demand of information through internet, the utility of computer and network is rapidly provided with to schools.

This situation brings about many problems. For example, the stolen information through false identification(Hacking) is the most greatest concern.

In this paper, it tells that the efficient way of preservating computer use is by using operating system of Open Source, which is Linux system. Further more, it shows the system which was organized by IP-Tabling (offered service-Packet Filtering method from the Linux system) functions well as a security system.

I. 서 론

정부의 초·중등학교 정보화 기반 구축 사업은 새로운 교육 방법으로 창의적인 인재를 육성할 수 있는 여건은 조성되으나,[1] 컴퓨터나 네트워크 구성 등의 물적인 측면만을 강조하여 그에 따르는 해킹의 증가 문제가 새롭게 대두되고 있다.

본 논문에서는 Linux 운영체제에서 제공하고 있는 Packet Filtering 모듈인 IPTables의 정책 설정을 이용하여 경제적으로 부담이 적은 학내 보안 시스템을 구축할 수 있는 방안을 제시하고자 한다.

리눅스를 사용하는 장점으로는 네트워크 호환성이 뛰어나며, 크기가 작으며, Hacking과 바이러스감염에 대한 안정성이 강하고 경제성이 뛰어나다는 점이다.

Packet Filtering 방식은 패킷의 헤더만을 대상으로 검색하기 때문에 네트워크에 부하를 주지 않고 효율적인 보안 모듈을 구현할 수 있다.

II. 관련 연구

2.1. 패킷 필터링(Packet Filtering)

패킷 필터링은 통상 라우터 인터페이스를 지나는 패킷을 필터링하기 위해 설계된 패킷 필터링 라우터(Packet Filtering Router)에 의해 행해진다. 패킷 필터링은 네트워크 레벨에서 작동하는데 방화벽의 규칙에서 허용되면 데이터가 시스템을 통과할 수 있다.[2] 그리고 각 패킷의 포트 번호에 따라 필터링 된다.

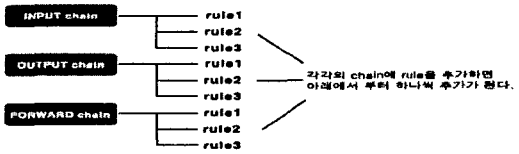
대부분 패킷 필터링은 필터링 규칙에 따라 패킷을 포워딩할 수 있는 라우터를 이용하여 이루어진다. 패킷이 패킷 필터링 라우터를 지나갈 때 라우터는 패킷의 헤더에서 특정 정보를 검출하여 필터링 규칙에 따라 패킷의 통과 여부를 결정한다.[8]

2.2. Linux에서 IPtables의 기능

리눅스 커널 2.4에서 지원되고 있는 방화벽구축과 네트워크 패킷 필터링 도구인 IPtables는 경제성과, 성능면에서 우수한 방화벽 구축 도구이다. IPtables는 설정하기 쉬울 뿐만 아니라 리눅스에서는 처음으로 상태 추적 기법이 도입된 방화벽으로 상당한 기술적 진보를 이

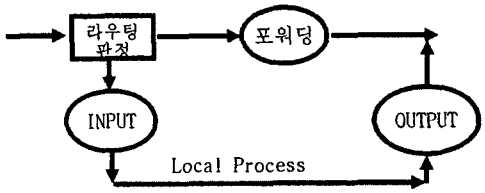
룬 지능형 방화벽이다.[4]

2.3. IPtables 동작 방식



[그림 1] Iptable의 3가지 Chain

IPtables는 [그림 1]과 같이 세 가지 체인에 적용되는 Rule에 의해서 작동된다.



[그림 2] Chain 구조

[그림 2]에서 세 개의 원은 위에서 언급한 세 개의 체인을 나타낸다. 패킷이 해당 체인에 도달하면 그 체인은 그 패킷의 운명을 결정하기 위하여 시험한다. 체인이 그 패킷을 DROP하라고 하면 패킷은 그곳에서 삭제된다. 그러나 그 체인이 ACCEPT하라고 하면 이 그림의 다음 부분으로 계속 전달된다.

패킷이 커널에 도착하면 그 패킷의 목적지를 확인하여 목적지가 정확하면 패킷은 INPUT 체인에 전달되며 체인이 접근을 허락하면 내부 프로세서에서 Access가 가능해진다. 체인이 패킷의 접근을 금지했다면, 패킷은 DROP되며, INPUT에 정책이 없다면 이 패킷은 포워딩 체인으로 가게 된다, 포워딩이 가능하게 되어 있고 다른 곳이 목적지이면 패킷은 그림의 오른쪽 방향으로 전달되어 포워딩 체인으로 간다. 이 체인이 ACCEPT하게 되면 이것은 포워딩할 네트워크로 보내진다.

마지막으로, 내부 Local Process에서 외부로 보내지는 패킷은 즉시 출력 체인에 보내진다. 이 체인이 ACCEPT하게 되면 이 패킷은 그 목적지가 어디든지 보내진다.[8]

III. 보안 시스템 구현

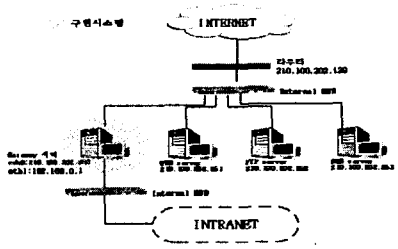
3.1. 시스템 구성

본 시스템을 구성한 하드웨어 사양은 Pentium IV 2GHz 256RAM이며, 사용하고 있는 NIC는 3Com 3c905B, Intel EtherExpress Pro100이다. 운영체제는 RedHat 9.0, 커널 버전은 2.4.20-8이다. 보안 모듈로 사용되는 IPtables는 Open Source로 공개된 패키지를 사용하였으며 버전은 1.2.7이다. 네트워크 환경은 초고속 국가망 E1(2048Kbps)을 연결하였다.

3.2. 보안 모듈 구현

3.2.1. 게이트웨이 서버 개념

[그림 3]은 구현하고자 하는 Gateway 서버를 포함한 네트워크 구성도를 보여주고 있다. 여기에서 Gateway 서버는 외부의 인터넷으로부터 내부 인트라넷의 접속을 차단하며 내부인트라넷에서 외부 인터넷으로 보내지는 데이터는 Gateway 서버에서 "마스커레이딩" 하여 전달한다.



[그림 3] 구현 네트워크 구성도

3.2.2. IPtables 스크립트 작성

본 논문에서 작성하고자 하는 게이트웨이 서버는 패킷 필터링 방식이며, 이는 다음과 같이 설명할 수 있다.

1) 패킷 필터 기준은 패킷 필터 장치의 포트에 대해서만 저장되어야만 된다.

2) 패킷이 포트에 도착할 때, 패킷 헤더를 분석한다. 대부분의 패킷 필터 장치는 IP, TCP, UDP 헤더에 들어 있는 영역만 조사한다.

3) 패킷 필터 규칙은 특정 순서로 저장된다. 각 규칙은 패킷 필터 규칙에 저장되어 있는 순서대로 패킷에 적용한다.

4) 만약 규칙이 패킷의 전송이나 수신을 막으면, 그 패킷은 허용되지 않는다.

5) 만약 규칙이 패킷의 전송이나 수신을 허락하면, 그 패킷은 진행하도록 허용된다.

6) 만약 패킷이 어떠한 규칙도 만족하지 않으면 그 패킷은 가로막는다.

스크립트파일은 부팅 시 실행될 수 있도록 /etc/rc.d/init.d 디렉터리에 "iptables"라는 file명으로 작성되며 파일 속성에 실행 모드를 추가해 준다.

3.3. 스크립트 파일 분석

3.3.1. 기본 필터링 정책

먼저 기본 필터링 정책은 모두 DROP으로 설정한다. 그리고 모든 내부 호스트는 방화벽 접근이 가능하게 설정한다.

```
iptables -P INPUT  DROP
iptables -P OUTPUT DROP
```

```
iptables -A INPUT  -i $LOOPBACK_INTERFACE  -j ACCEPT
.....
```

3.3.2. 비정상적인 패킷의 거부

안정성을 확보하기 위해 비정상적인 패킷은 모두 거부하도록 설정하였다.

```
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
iptables -A FORWARD -p tcp --tcp-flags ALL ALL -j DROP
.....
```

3.3.3. 내부 호스트에 대한 서비스 제공

내부에 연결되어 있는 호스트들의 서비스를 위해 내부에서 나가는 패킷과 내부의 응답에 대한 외부 답변 패킷은 접근을 허용하였다.

```
iptables -A FORWARD -m state --state NEW -i $LOCAL_INTERFACE_1 W
-s $INTRANET -j ACCEPT
.....
```

IV. 실험 및 평가

4.1. 시스템 환경

게이트웨이 서버의 성능 실험을 위해 동일 네트워크 상에서 게이트웨이 서버와 동일한 위치에 외부 클라이언트를 두었으며, 게이트웨이 하단에 내부 클라이언트를 두어 외부와 내부의 클라이언트에 대한 게이트웨이 접속 여부를 테스트하였다. 평가의 정확성을 위해 내부 클라이언트와 외부 클라이언트의 시스템은 동일한 사양으로 하였다. 게이트웨이 서버의 운영체제는 RedHat Linux 9.0이며, 클라이언트 시스템의 운영체제는 Windows XP를 사용하였다. 게이트웨이 서버에서 내부 클라이언트에 대해 외부 접속을 허용함으로써, 간접적인 접속이 되도록 한다. 보안 역할을 하는 게이트웨이 서버는 하드웨어적으로 공개되어 있어, 소프트웨어적인 보안을 통해 외부로부터의 접속을 차단해야 한다.

4.2. 체인 설정

Input chain의 기본 설정은 모든 연결을 거부하는 "DROP" 이며, TCP, UDP, ICMP 등의 특정 연결을 허용하는 PORT에 대해서 접속을 허가하는 "ACCEPT" 설정을 한다. Output chain의 기본 설정 역시 모든 연결을 거부하는 "DROP" 이며, TCP, UDP, ICMP 등의 특정 연결을 허용하는 PORT에 대해서 접속을 허가하는 "ACCEPT" 설정을 한다. 마지막으로 FORWARD 체인은 앞의 INPUT, OUTPUT 체인과는 달리 기본 설정값이 ACCEPT이며 본 논문에서 제어한 체인은 내부 네트워크에 대해서 마스커레이딩에 대한 허용이다.

4.3. 성능 실험

게이트웨이 서버의 보안성을 시험하기 위해 Nmap과 IP-Tools를 이용하였다. Nmap은 리눅스/유닉스용 네트워크 보안 및 분석 유틸리티이며, IP-Tools는 윈도우용 분석 유틸리티로 네트워크 분석 및 해킹 도구로 시스템 관리자는 물론이고 해커를 포함한 네트워크 환경을 점검하기를 원하는 모든 사람에게 매우 유용한 도구로 사용된다.[6] 본 논문에서는 외부 클라이언트 영역에 존재하는 호스트 중 RedHat Linux 9.0 OS에 Nmap을 설치하여 테스트하였으며, 또한 Windows XP에 IP-Tools를 설치하여 테스트하여 결과를 분석해 보았다. 그리고 BenchBee를 이용하여 게이트웨이 서버의 포워딩 성능을 분석하였다.

4.3.1. Nmap을 이용한 성능 분석

Nmap(Network Mapper)은 네트워크 보안을 위한 유틸리티로, 대규모 네트워크를 고속으로 스캔하는 도구이다. Nmap은 raw IP 패킷을 사용하여 네트워크에 어느 호스트가 살아있고, 그들이 어떠한 서비스(포트)를 제공하며, 운영체제(OS 버전)가 무엇이며, filter / firewall의 패킷 타입이 무엇인지 등 네트워크의 수많은 특징들을 점검할 수 있다. [15,16]

```
[han@ns bin]$ ./nmap -sP 210.100.202.250

Starting nmap 3.45 ( http://www.insecure.org/nmap/ ) at 2003-10-08 01:28 KST
Note: Host seems down. If it is really up, but blocking our ping probes, try -P0
Nmap run completed -- 1 IP address (0 hosts up) scanned in 24.008 seconds
[han@ns bin]$
[han@ns bin]$
[han@ns bin]$
[han@ns bin]$ ./nmap -sP 192.168.0.2

Starting nmap 3.45 ( http://www.insecure.org/nmap/ ) at 2003-10-08 01:28 KST
Note: Host seems down. If it is really up, but blocking our ping probes, try -P0
Nmap run completed -- 1 IP address (0 hosts up) scanned in 24.008 seconds
[han@ns bin]$
```

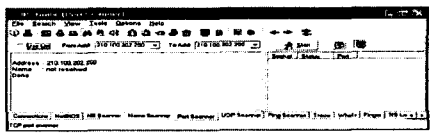
[그림 4] Nmap을 이용한 Gateway 및 호스트 검색

[그림 4]에서 보여주는 것과 같이, Nmap을 이용하여 성능을 실험한 결과 24초동안 게이트웨이의 모든 포트를 검색하였으나 모든 Port를 스캔할 수가 없었다. 이는 게이트웨이 서버에서 허가되지 않은 외부의 접근을 효율적으로 차단하고 있다는 것을 보여주고 있다.

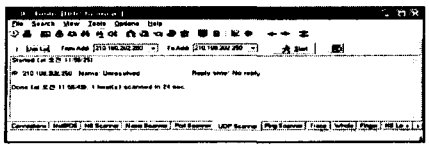
4.3.2. IP-Tools를 이용한 서버 분석

IP-Tools는 TCP/IP 프로토콜을 사용하는 네트워크에서 공유된 자원 검색, Port Scan, UDP Scan 등을 할 수 있는 Windows OS용 네트워크 분석 도구이다.[6,17]

아래 두 개의 그림은 IP-Tools를 이용하여 Port와 UDP를 검색한 모습을 보여주는 화면이다. 그림에서 보여주는 것과 같이, 게이트웨이 서버의 Port와 UDP Port를 IP-Tools에서 검색을 할 수 없었다. 이는 게이트웨이 서버가 윈도우즈 운영체제를 이용한 허가되지 않은 외부의 접근을 서버에서 효율적으로 차단하고 있음을 보여주는 것이라 할 수 있다.



[그림 5] IP-Tools를 이용한 Port Scan



[그림 6] IP-Tools를 이용한 UDP Scan

4.3.3. Forwarding 성능 실험

다음은 게이트웨이 하단에 연결되어 있는 내부 클라이언트가 게이트웨이를 통해 외부 네트워크에 잘 연결되고 있는 지 즉, 게이트웨이 서버의 Forwarding 성능을 실험한 내용이다. 이 실험을 위해서 인성정보의 BenchBee(http://www.benchbee.co.kr)에서 제공하는 인터넷 접속 서비스 성능 테스트 사이트를 이용하였다. 인성정보의 벤치비는 초고속 인터넷의 객관적 품질평가를 통해 초고속 인터넷 서비스의 품질을 알려주는 서비스이다.[18]

구분	외부클라이언트	내부클라이언트
Download 속도	1.433Mbps	1.421Mbps
Upload 속도	1.212Mbps	1.482Mbps
Packet 지연율	15.6ms	0.0ms
Packet 손실률	0.0%	0.0%

[표 1]인터넷접속 성능 평가 결과

[표 1]은 외부 클라이언트와 내부 클라이언트에서 각각의 인터넷 접속 성능을 실험한 결과를 표로 정리한 것이다. 위 표에서 보듯 다운로드나 업로드에서 거의 속도의 저하를 느낄 수 없음을 알 수 있다. 반면 패킷 지연은 게이트웨이를 통과한 패킷에서는 거의 발생하지 않는 것을 볼 수 있다.

V. 결론

본 논문에서는 Open Source 운영체제인 Linux 시스템에서 제공하고 있는 Packet Filtering 방식의 보안 커널 모듈인 IPtables를 이용하여 경제적이면서 효율적으로 보안 문제를 해결할 수 있는 모델을 제안하였다.

Iptable 커널의 Input, Output, Forward 세 가지 체인을 제어하여, 보호하고자 하는 Host에 접근하는 모든 패킷을 감시하여, 감시대상 패킷을 차단하거나 거부함으로써, 내부 시스템의 안정성을 확보하였으며, 보안 기능을 담당하는 게이트웨이 서버의 모든 Port를 숨김으로써 해킹으로부터 방지할 수 있었다.

Linux OS 상에서 운영되는 본 모델은 다른 상용화 된 많은 보안 시스템과 비교하여 경제적인 부담은 최소화 시키면서 보안 성능 또한 그에 뒤지지 않음을 확인할 수 있다.

참고 문헌

[1] 한국교육학술정보원, 2002년 교육정보화 백서 http://www.keris.or.kr/data/book_2001.jsp?layerNo=dataleft3#3.

[2] 한국정보통신인력개발센터, 리눅스마스터1급필기+실기 특별대비, 2002

[3] 김선정, Unix 환경에서 방화벽 시스템의 효율적 운용에 관한 연구, 호남대학교 정보산업대학원 소프트웨어공학전공 석사학위 논문, 2001

[4] *Gerhard Mourani, Securing&optimizing Linux*, OpenNA.com 2003

[5] 정준목, 브릿지+넷필터로 브릿지-방화벽 구축하기, http://chunmok.hihome.com/publish/bridge-iptable/index.html

[6] 노용환외, 해킹과 보안 내가 최고, 영진닷컴, 2000

[7] http://oshelp.net/oswiki

[8] Rusty Russel, 김상훈 리눅스 2.4 패킷필터링 하우투, http://doc.kldp.org/

[9] 배철수, 커널 2.4에서 NAT(network address translation) 구현, http://www.linuxlab.co.kr/docs/01-03-2.htm, 2003

[10] Linux 2.4 Packet Filtering HOWTO
http://netfilter.kernelnotes.org/unreliable-guides/index.html
http://kldp.org/Translations/html/Packet_Filtering-KLDP/Packet_Filtering-KLDP-7.html

[13] Masquerading Made Simple HOWTO, http://kldp.org/HOWTO/html/Masquerading-Simple-HOWTO

[14] 이재홍, 리눅스로 Bridge Firewall 만들기 http://doc.kldp.org/DcobookSgml/Bridge_Firewall_KLDP

[15] 전숙, http://certcc.or.kr/tools/Nmap.html

[16] http://www.insure.org/nmap

[17] http://www.ks-soft.net/ip-tools.eng/

[18] http://www.benchbee.co.kr