병렬처리 HIT 기법과 로드밸런싱 WLC기법이 적용된 HWbF(Hit and WLC based Firewall) 설계

HWbF(Hit and WLC based Firewall) Design using HIT technique for the parallel-processing and WLC(Weight Least Connection) technique for load balancing

이 병 관* 권 동 혁** 정 은 희***
Byung-kwan Lee Dong-Hyeok Kwon Eun-Hee Jeong

요 약

본 논문에서 설계한 HWbF(Hit and WLC based Firewall)은 PFS(Packet Filter Station)과 APS(Application Proxy Station)으로 구성된다. PFS는 로드 밸런싱을 이용한 PLB(Packet Load Balancing) 모듈을 이용해 패킷들을 분산시켜 패킷 전송 지연을 방지하고 병목현상을 줄이도록 설계하였고 APS는 로드 밸런싱을 이용한 PCSLB(Proxy Cash Server Load Balancing)모듈을 이용해 효율적인 프록시 캐쉬 서버를 관리하고, 패킷 트래픽량을 이용해 DoS 공격을 탐지하도록 설계하였다. 따라서, 본 논문에서 설계한 HWbF는 기존이 방화벽의 단점이었던 패킷 전송 지연을 방지하고 병목현상을 줄여 패킷 처리속도를 향상시킨다. 또한, 패킷 트래픽 임계값을 패킷 트래픽 량에 따라 자동 조절함으로써 기존의 평균값과 고정 임계치에 대한 각각의 DoS 공격 오탐지율(TCP)이 50%와 25%에서 제안한 수식에 의해 각각 38%와 17%으로 감소시켜, DoS 공격 트패픽의 탐지 능력을 향상시킬 뿐만 아니라 프록시 캐쉬 서버의 부하도 줄인다.

Abstract

This paper proposes HWbF(Hit and WLC based Firewall) design which consists of an PFS(Packet Filter Station) and APS(Application Proxy Station). PFS is designed to reduce bottleneck and to prevent the transmission delay of them by distributing packets with PLB(Packet Load Balancing) module, and APS is designed to manage a proxy cash server by using PCSLB(Proxy Cash Server Load Balancing) module and to detect a DoS attack with packet traffic quantity. Therefore, the proposed HWbF in this paper prevents packet transmission delay that was a drawback in an existing Firewall, diminishes bottleneck, and then increases the processing speed of the packet. Also, as HWbF reduce the 50% and 25% of the respective DoS attack error detection rate(TCP) about average value and the fixed critical value to 38% and 17%. with the proposed expression by manipulating the critical value according to the packet traffic quantity, it not only improve the detection of DoS attack traffic but also diminishes the overload of a proxy cash server.

☞ keyword : HWbF, PLB, PCSLB, Hit ACL, 로드 밸런싱, 가중최소연결, 방화벽

1. 서 론

최근 네트워크 시스템에 대한 악의적인 접근과 정보 위협이 증가하고 그 피해 또한 기업에서 개인 사용자까지 확대되고 있으며, 세계 각국은 다가오는 유비쿼터스 시대의 새로운 강자가 되기 위해 자국의 정보화에 발전에 총력을 기울이고 있다. 우리나라 역시 정보화 사회 구현 및 산업육성에 힘쓰고 있지만 정보통신기술의 비약적인 발

* 종신회원 : 관동대학교 컴퓨터학과 교수
bklee@kwandong.ac.kr
** 정 회 원 : 관동대학교 대학원 전자계산공학과 재학 중
(박사과정) kdh0108@kwandong.ac.kr
*** 정 회 원 : 강원대학교 지역경제학과 조교수
jeongeh@kangwon.ac.kr(교신저자)
[2008/06/17 투고 - 2008/06/18 심사 - 2008/10/15 심사완료]

전과 함께 이를 악용한 해킹, 웜, 바이러스 등 사이버 침해 위협이 날로 지능화, 다양화되고 있는 실정이다.

이것은 개인 사용자의 요구 조건에 맞는 보안 시스템이 필요하다는 것을 의미한다. 하지만 기존의 방화벽은 사용 용도에 따라서 규격화된 솔루션을 제공하므로, 높은 가격의 거대한 보안 솔루션은 기업에서 조차 피하고 있다. 기존 보안 솔루션의 문제인 관리상의 문제, 성능의 문제, 비용 상의 문제를 보안하기 위해 본 논문에서는 프로토콜의 병렬처리를 위한 Hit 기법과 로드밸런싱을 위한 WLC(Weight Least Connection) 기법이 적용된 HWbF(Hit and WLC based Firewall)를 설계한다. HWbF는 객체화된 모듈로 이루어져 있으며, 전체 시스템에서 사용하고자 하는 모듈만 사용할 수 있으므로 기존의 규격화된 고가의 보안 솔루션에 대한 비용상의 문제점을 제거하고, 로드 밸런싱을 이용해 기존의 방화벽의 단점이었던 패킷 전송 지연을 방지하고 병목현상을 줄여 패킷 처리 속도를 향상시킴으로써 성능의 문제점을 제거하였다. 또한, 임계값을 유동적으로 설정함으로써 DoS 공격 트래픽을 탐지 및 차단하도록 설계하여 관리자의 부담을 최소화함으로써 관리상의 문제점을 제거하였다.

2. 관련연구

2.1 방화벽

방화벽(firewall)이란 외부로부터 내부 네트워크를 보호하기 위한 네트워크 구성요소 중의 하나로 외부의 불법침입으로부터 내부의 정보 자산을 보호하고 외부로부터 유해정보 유입을 차단하기 위한 정책과 이를 지원하는 하드웨어와 소프트웨어를 말한다.

방화벽은 외부 네트워크와 연동하는 유일한 창구로서 외부로부터 내부 네트워크를 보호하기 위해 각 서비스(ftp, telnet 등)별로 서비스를 요구한 시스템의 IP주소 및 port번호를 이용하여 외부의 접속을 차단하거나 또는 사용자 인증에 기반을 두고 외부접속을 차단하며, 상호 접속된 내·외부 네트워크에 대한 트래픽을 감시하고 기록한다.

따라서 방화벽은 네트워크의 출입로를 단일화함으로써 보안관리 범위를 좁히고 접근제어를 효율적으로 할 수 있어, 외부에서 불법으로 네트워크에 침입하는 것을 방지하면서 내부의 사용자가 네트워크를 자유롭게 사용할 수 있다. 또한 방화벽에는 역추적 기능이 있어서 어떠한 네트워크 접근이라도 그 흔적을 찾아 역추적이 가능하다.

방화벽은 동작 원리에 근거하여 패킷 필터링 방화벽과 애플리케이션 게이트웨이로 분류할 수 있다. 표 1은 방화벽 동작원리에 따른 분류로 패킷 필터 방화벽과 애플리케이션 방화벽에 대한 설명이다[1,2].

(표 1) 방화벽의 동작 원리에 따른 분류

분 류	설 명
패킷필터링 방화벽	OSI 7계층 중 3계층인 네트워크 층과 4계층인 트랜스포트 층에서 동작. 데이터링크 층에서 네트워크 층으로 전달되는 패킷을 가로채서 해당 패킷안의 주소와 서비스 포트 검색.
애플리케이션 게이트웨이	OSI 7계층에서 상위 계층인 애플리케이션 층에서 동작. 관리자가 지정해 놓은 허용 범위와 비교하여 허용여부 판단.

소프트웨어 방화벽은 많은 리소스 점유율과 지나친 보안설정으로 인해 인터넷이 느려지거나 병목현상이 발생하여 시스템의 부하가 증가하고, DoS 공격에 취약하다는 문제점을 갖고 있다.

본 논문에서는 제안하는 HWbF는 로드밸런싱을 이용한 PLB(Pakcet Load Balancing)이 패킷들을 병렬처리 하도록 PFDS(Packet Filter Data Segment)에 분배하도록 하여 병목현상을 해결하고, TAMS 모듈에서 정상적인 트래픽량에 따라 변하는 유동 임계치를 이용하여 임계치보다 높은 트래픽량은 갖는 패킷들은 DoS 공격으로 탐지하고, 탐지된 DoS 공격을 차단하도록 하였다. 또한, 탐지된 DoS 공격에 대한 정보를 Hit ACL에 등록하여,

PFDS 모듈은 Hit ACL에 등록된 정보를 이용해 패킷을 필터링하여 DoS 공격에 이용된 IP 및 Port 번호를 차단하도록 하였다. 따라서, HWbF는 단순히 패킷을 모니터링하는 방화벽이 아니라, DoS 공격을 판단하고, DoS 공격을 차단하는 IPS(Intrusion Prevention System) 기능이 도입된 방화벽이라 할 수 있다.

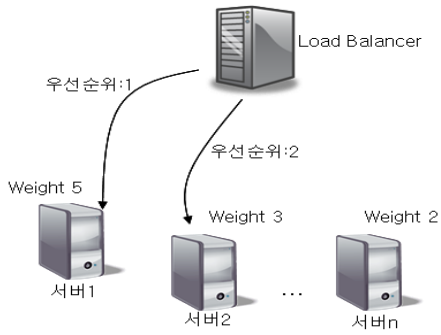
2.2 로드밸런싱

로드 밸런싱이란 작업을 여러 대의 컴퓨터에 균등하게 분산시킴으로서 한 대의 컴퓨터를 사용하는 것보다 짧은 시간에 사용자가 원하는 작업을 수행하는 것을 말한다. 즉, 수십 대의 컴퓨터를 사용한다고 하더라도 한 대의 컴퓨터에서 작업이 이루어진다면 성능 향상을 얻을 수 없기 때문에 작업을 균등하게 분산시키는 것이 중요하다.

2.2.1 가중치 기반 라운드 로빈 (WRR, Weight Round Robin)

가중치 기반 라운드 로빈 스케줄링은 라운드 로빈 방식의 일종으로 서버의 용량이 다를 때, 각 서버의 처리 용량에 비례하는 가중치를 두어 요청을 분산한다. 즉, 기본적으로 요청을 분산할 때 라운드 로빈 방식을 사용할 경우 가중치가 큰 서버에 더 많은 요청을 보내며 기본 가중치는 1이다.

가중치 기반 라운드 로빈 스케줄링을 사용하면 리얼 서버에서 네트워크 접속을 셀 필요가 없고 동적 스케줄링 알고리즘보다 스케줄링의 과부하가 적으므로 더 많은 리얼 서버를 운영할 수 있다. 그러나 요청에 대한 부하가 매우 많을 경우 리얼 서버 사이에 동적인 부하 불균형 상태가 생길 수 있다[3,4].

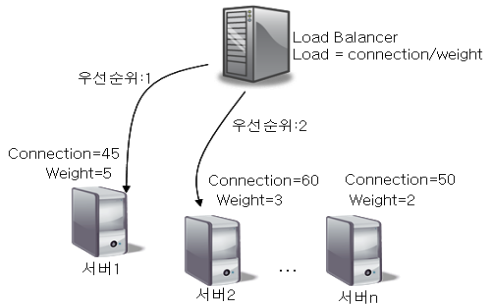


(그림 1) 가중치기반 라운드 로빈

2.2.1 가중치 최소 연결 (WLC, Weight Least Connection)

가중치 기반 최소 접속 스케줄링은 최소 접속 스케줄링의 한 부분으로서 실제 서버에 가중치를 부여하여 가중치에 따라 실제 접속 수가 적은 서버에 요청을 할당한다. 즉, 성능이 높은 서버가 더 많은 요청에 응답하도록 설계한 방법이다. 최소 접속 스케줄링에 비해 가중치를 계산하는 부가적인 배분 작업이 필요하고, 실제 서버의 실제 접속 수를 가중치로 나눈 값이 최소인 실제서버에게 요청을 할당하는데, 기본 가중치는 1이다.

이러한 동적 스케줄링 방식은 스케줄링 자체의 부하가 발생할 수 있기 때문에 이러한 부하를 줄이기 위하여 리얼 서버들이 비슷한 처리 능력을 지니고 있을 때를 대비하여 보통 최소 연결 스케줄링을 가중치 최소 연결 스케줄링과 동시에 구현한다[3,4].



(그림 2) 가중치 최소 연결

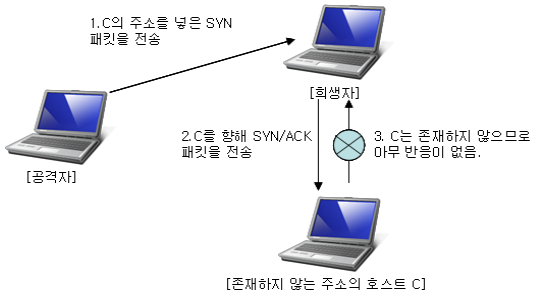
본 논문에서는 두 개의 로드 밸런싱 기법인 가중치기반 라운드 로빈과 가중치 최소 연결을 비교하여 패킷 처리량이 높은 로드밸런싱 기법을 PLB 모듈과 PCSLB 모듈에 적용하고자 한다.

2.3 DoS 공격

DoS 공격 유형은 SYN Flooding, UDP Flooding, ICMP Flooding으로 트래픽 양과 관련된 대표적인 공격 유형이다[5][6][7].

2.3.1 SYN Flooding

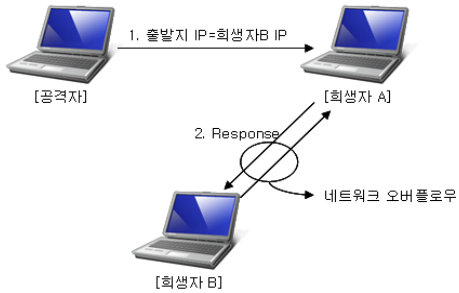
정상적인 TCP 접속은 SYN 패킷과 ACK 패킷을 세 번에 걸쳐서 올바른 값을 넣어서 주고받게 되어있다. 그런데, 공격자가 인터넷상에 존재하지 않는 호스트 C의 주소를 이용한 패킷으로 TCP 접속을 시도하면 희생자는 호스트 C에 SYN/ACK 패킷을 전송하고 호스트 C의 ACK 패킷을 기다리다가 대기시간이 만료되어야 연결을 위해 할당한 자원을 회수하게 된다. 이렇게 잘못된 연결이 많아지면 희생자는 다른 정상적인 TCP 연결을 할 수 없게 되는데 이것을 SYN Flooding 공격이라고 한다[2].



(그림 3) SYN Flooding에 의한 DoS 공격

2.3.2 UDP Flooding

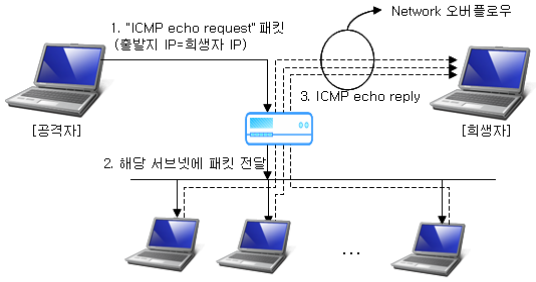
UDP는 비연결형 서비스로 포트 대 포트로 전송한다. UDP Flooding은 UDP의 비연결성 및 비신뢰성 때문에 UDP Flooding은 공격이 용이한 방법이다. UDP는 소스 주소와 소스 포트를 스푸핑하기 쉽다는 약점을 이용해 과다한 트래픽을 희생자에게 전송함으로써 희생자간의 네트워크를 마비시키는 공격이다.[7]



(그림 4) UDP Flooding에 의한 DoS 공격

2.3.3 ICMP Flooding

ICMP는 호스트간 혹은 호스트와 라우터간의 에러 상태 혹은 상태 변화를 알려주고 요청에 응답하는 기능을 담당하는 네트워크 제어 프로토콜이다. 이러한 ICMP는 활성화된 서비스나 포트가 필요하지 않다는 ICMP 특징을 악용한 ICMP Flooding은 대량의 ICMP 패킷을 공격자가 직접 희생자에게 전송하는 방법으로 Smurf, Welchia worm 등이 있다[6].



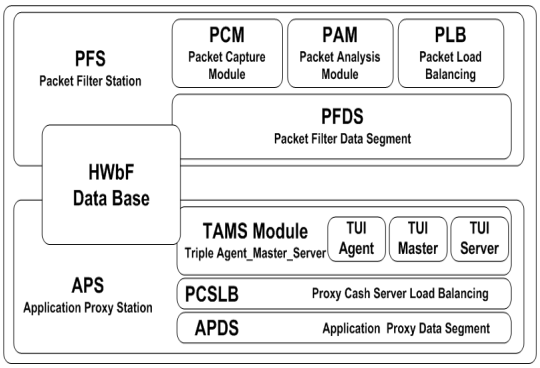
(그림 5) ICMP Flooding에 의한 DoS 공격

3. HWbF 설계

본 논문에서 설계한 차세대 능동형 네트워크

정보보호시스템인 HWbF(Hit and WLC based Firewall)은 로드 밸런싱을 이용해 패킷을 분산시켜 패킷 필터링을 담당하는 PFS(Packet Filter Station)과 패킷 트래픽량을 이용해 DoS 공격을 탐지하고, 프록시 캐쉬 서버를 이용해 서버에 걸리는 트래픽을 감소시키는 APS(Application Proxy Station)로 구성된다.

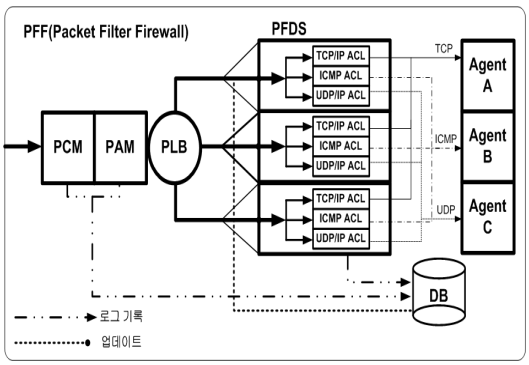
그림 6은 HWbF 시스템의 모듈 구성을 설명한 것이다.



(그림 6) HWbF 시스템 모듈

3.1 PFS 설계

그림 7은 PFS 설계 구성도를 설명한 것이다.



(그림 7) PFS 설계 구성도

PFS(Packet Filter Station)는 패킷이 전송되었을 경우 패킷을 캡쳐하는 패킷 캡쳐 모듈(PCM:packet capture module), 캡쳐된 패킷을 분석하는 패킷 분석 모듈(PAM:packet analysis module), 패킷의 병렬처리를 위한 패킷 로드 밸런싱(PLB:packet load balancing) 그리고 Hit가 장착된 패킷 필터 데이터 세그먼트(PFDS:packet filter data segment)로 구성된다.

3.1.1 PCM 설계

이더넷 환경에서는 내부 네트워크로 향하는 패킷들을 브로드캐스팅 하게 되고 각 시스템은 자신의 주소가 목적지인 패킷만을 받아들여 운영체제가 처리하게 되는데, 이때 인터페이스의 수신 Mode가 Promiscuous로 설정이 되면 목적지가 어디든 상관없이 네트워크상의 모든 패킷을 수신할 수 있게 된다.

그림 8은 패킷을 캡쳐하기 위한 패킷 캡쳐 알고리즘으로, 클라이언트로부터 들어오는 패킷을 분석하기 위하여 우선적으로 패킷을 캡쳐한다 [8,9].

```
#include "capture.h"
void capture_init(u_int8_t *filter, u_int64_t cnt){
  u_int32_t d_link, localnet, netmask;
  pcap_dumper_t *p_dumper = NULL;
  struct bpf_program bpf;
  #ifdef DEBUG
  fprintf(stdout, "DEBUG: capture_init()\n");
  #endif
  memset(&bpf, 0, sizeof(struct bpf_program));
  signal(SIGTERM, capture_clean_exit);
  signal(SIGINT, capture_clean_exit);
  signal(SIGQUIT, capture_clean_exit);
  signal(SIGHUP, capture_clean_exit);
  if(strlen(r_file) > 0){
    if((pkt = pcap_open_offline(r_file, error_buf))
      == NULL)
      fatal_error("Unable to open file: %s", error_buf);
  }
}
```

(그림 8) 패킷 캡쳐 알고리즘

3.1.2 PAM 설계

패킷 분석 모듈은 패시브 모드로 버퍼에 저장

된 모든 패킷에 대하여 read 함수를 통하여 하나의 오픈 파일로 읽어 들여서 이를 여러 프로토콜과 관련된 헤더의 정보 분석을 위하여 패킷 헤더를 추출하여 분석한다.

그림 9는 패킷 분석 알고리즘으로 패킷을 이더넷 헤더 형태로 변화 한 후 이더넷 정보를 가져온다. 이더넷 헤더의 타입 필드를 확인하고 IP, ARP, RARP 로 구분하고 IP 패킷에 대해서는 헤더의 프로토콜을 확인하여 TCP, UDP, ICMP, IGMP 패킷으로 구분하여 각 정보는 해당함수에 의하여 파일에 출력하여 저장한다[8,9].

```
if(eth_header->h_proto ==htons(ETH_P_IP)){
  if(eth_header->ip_p == 6)
       /* TCP protocol */
  else if(ip_header->ip_p == 17)
       /* UDP protocol */
  else if(ip_header->ip_p == IPPROTO_ICMP)
       /* ICMP protocol */
  else if(ip_header->ip_p == IPPROTO_IGMP)
      /* IGMP protocol */
  else if(ip_header->h_proto== htons(ETH_P_ARP))
      /* ARP protocol */
  else if(ip_header->h_proto == htons(ETH_P_RARP))
       /* RARP protocol */
}
else{
  fp = (FILE*)open_file_log_fd(LOGFILE);
  print_eth_info(fp);
  fprintf(fp,"/n DO NOT SUPPORT /n");
  fclose(fp);
}
```

(그림 9) 패킷 분석 알고리즘

3.1.3 PLB 설계

본 논문에서 제안한 PLB는 가중치 최소 연결 알고리즘을 사용하여 각각의 PFDS에 성능 가중치를 부여할 수 있도록 하여, 언제라도 가중치가 높은 PFDS에 더 많은 요청을 처리할 수 있도록 하였다. 이는, 클라이언트에서 수신된 패킷에 대하여 가중치 최소 연결 알고리즘을 실행하여 가중치가 높은 PFDS를 선택하도록 한 것으로, 가중치 최소 연결 알고리즘의 기본 가중치는 1이고, 가중치가 주어진 3개의 각 PFDS 서브 모듈 i는 가중치 $W_i(i=1,2,3)$일 때, PFDS 서브모듈 i의 패킷 처리량은 $Packet_{su_i}(i=1,2,3)$이고 전체 패킷 처리량은 $Packet_{su_i}(i=1,2,3)$의 합이다.

PFDS의 각 서브 모듈 중 처리율이 높은 서브 모듈 j로 가는 패킷 분배가 $(Packet_{su_j}/S)/W_j = \min\{(Packet_{su_i}/S)/W_i\}$일때, $S=\sum Packet_{su_i}$는 상수이기 때문에 S로 $Packet_{su_i}$를 나눌 필요가 없으므로, $Packet_{Su_j}/W_j = \min(Packet_{Su_i}/W_i)$으로 최적화가 된다[3]. 이렇게 PFDS의 최적화된 값(WLC)을 을 비교하여 값이 적은 PFDS를 선택한다.

따라서, 본 논문에서 제안한 PLB는 PFDS의 WLC값을 검사하여 값이 적은 PFDS를 선택하여 패킷을 필터링 하도록 함으로써, 기존의 방화벽의 단점이었던 패킷 전송지연을 방지하고, 병목현상을 줄여 패킷 처리 속도를 향상시킬 수 있다.

그림 10은 PLB 알고리즘이다.

```
PLB(int packet_su[]){
  int i, w[4], wlc[4], min_value, select_pfds;
  for(i=1; i<=3; i++) w[i]=i;
  for (i=1; i<=3; i++) wlc[i]=packet_su[i]/w[i];
  if(wlc[1]<=wlc[2]) {
    min_value=wlc[1];
    select_pfds=1;
  }
  else {
    min_value=wlc[2];
    select_pfds=2;
  }
  if(wlc[3]<=min_value){
     min_value=wlc[3];
     select_pfds=3;
  }
  send_packet(packets, select_pfds);
}
```

(그림 10) PLB 알고리즘

3.1.4 PFDS 설계

PFDS는 패킷 필터를 담당하는 모듈로서 세 개의 서브 모듈로 구성되어 있으며, 각 서브모듈은 TCP/IP, ICMP, UDP/IP 패킷을 나누어 처리한다.

이때, 각각의 TCP/IP, ICMP, UDP/IP 패킷들은 HiT ACL에 의해 허용된 패킷들만 처리하도록 설계되었으며, Hit ACL에 의해 필터링 된 패킷의 처리량을 각 서브모듈의 $packet_{su}$에 기록한다. 이렇게 기록된 $packet_{su}$은 PLB가 PFDS 서브 모듈을 선택할 수 있는 가중치의 값으로 이용된다.

PFDS의 알고리즘은 그림 11과 같다.

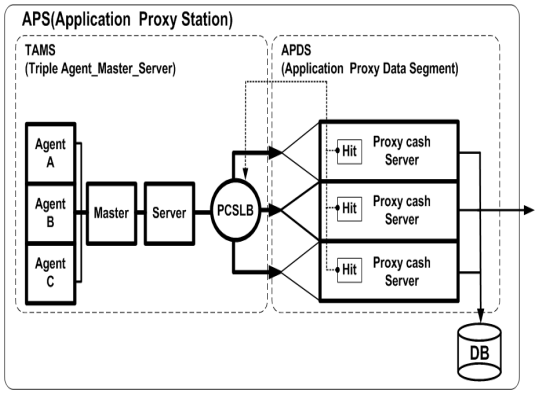
```
PFDS(char packets[], int packet_su[], int select_pfds){
  switch(select_pfds){
    case 1 : if(type(packets)="TCP"){
               if(check(hitACL_pfds, packets){
                 packet_su[1]=packet_su[1]+packets;
                 sendAgent(packets, agentA);
               }
             }
             if(type(packets)="ICMP"){
               if(check(hitACL_pfds, packets){
                 packet_su[2]=packet_su[2]+packets;
                 sendAgent(packets, agentB);
               }
             }
             if(type(packets)="UDP"){
               if(check(hitACL_pfds, packets){
                 packet_su[3]=packet_su[3]+packets;
                 sendAgent(packets, agentC);
               }
             }
             break;
    case 2 :    ;
         :
  }
}
```

(그림 11) PFDS 알고리즘

3.2 APS 설계

APS(Application Proxy Station)는 DoS/DDoS 공격을 탐지하고 차단하는 TAMS(Triple Agent Master Server) 모듈과 역 프록시 방식으로 TAMS 모듈을 통과한 패킷을 APDS로 분배하도록 설계된 PCSLB(Proxy Cash Server Load Balancing) 모듈로 구성된다.

그림 12는 APS 설계 구성도이다.



(그림 12) APS 구성도

3.2.1 TAMS 모듈 설계

TAMS 모듈은 초기 임계값 설정 후 트래픽의 변화에 따른 임계값 설정으로 DoS 공격트래픽을 탐지 및 차단하도록 설계되었다.

TUI Agent는 TCP Agent, UDP Agent, ICMP Agent로 구성되어있고, 각 Agent의 공격탐지 모듈에서는 항상 패킷 이동량을 분석하여 임계치 이상으로 패킷 이동량이 증가할 때 공격 패턴이라고 분석하여 TUI Master에게 보고한다. TUI Server는 정상적인 상태의 패킷 이동량에 대한 로그정보를 바탕으로 프로토콜별 비율분석을 통한 프로토콜별 임계치를 생성하고 TUI Agent의 공격탐지 모듈에서 사용할 수 있도록 TUI Master로 전송한다. 설정된 임계치 이상의 패킷 증가량이 탐지 되었을 경우 1차 경보를 TUI Master로 보고하며 프로토콜별 증가량 분석과 임계치를 비교하여 TCP, UDP, ICMP에서 발생하는 Flooding을 감시한다.

TAMS 모듈의 각 프로토콜에 대한 평균값 계산식과 임계치 계산식은 다음과 같다. 여기서, TAMS는 TCP, UDP, ICMP 각 프로토콜을 의미한다.

$$TAMS_{average} = (TAMS_{traffic} + TAMS_{average})/2$$

$$TAMS_{criterion} = (TAMS_{traffic} + TAMS_{criterion})/2 + util_{TAMS}$$

그림 13은 TAMS모듈의 임계치 계산 알고리즘이다[10].

```
TAMS(char packets[], int agent[]){
  switch(agent){
    case A: tcp_traffic = ∑Packet_tcp /∑Packet_IP;
            if( tcp_traffic>= tcp_criterion)
                    sendMaster(alertLog);
            u_tcp = (tcp_traffic + tcp_criterion )/2;
            tcp_criterian=calMax(u_tcp, tcp_traffic);
            break;
    case B: udp_traffic = ∑Packet_udp /∑Packet_IP;
            if(udp_traffic >= udp_criterion)
                    sendMaster(alertLog);
            u_udp = (udp_traffic + udp_criterion )/2;
            udp_criterion=calMax(u_udp, udp_traffic);
            break;
    case C: icmp_traffic = ∑Packet_icmp /∑Packet_IP;
            if(icmp_traffic >= icmp_criterion)
                    sendMaster(alertLog);
            u_icmp = (icmp_traffic + icmp_criterion )/2;
            icmp_criterion=calMax(u_icmp, icmp_traffic);
            break;
  }
  calMax(u_t, p_traffic){
    t_util = (p_traffic/Total_traffic);
    t_criterion = u_t + t_util;
    return(t_criterion);
  }
}
```

(그림 13) TAMS 임계치 계산 알고리즘

3.2.2 PCSLB 설계

PCSLB(Proxy Cash Server Load Balancing) 모듈은 클라이언트의 요청에 대한 프록시 캐쉬 서버의 처리율을 높이고 cash 메모리의 효율적인 사용을 위해 설계되었으며, PLB와 동일한 역할을 한다. 즉, 가중치 최소 알고리즘을 이용해 쇄도하는 클라이언트들에 대하여 프록시 캐서 서버의 메모리를 최적으로 분배함으로써 누락되는 패킷을 줄이고, 처리속도를 향상시키도록 설계되었다. 즉, TUI Server에서 전송받은 패킷들을 PCSLB에 의하여 부하가 가장 적은 프록시 서버로 분배된다.

```
PCSLB(char packets[], int cash[]){
  int i, W[4], wlc[4], min_value, select_apds;
  for(i=1; i<=3; i++) W[i]=i;
  for (i=1; i<=3; i++)  wlc[i]=cash[i]/W[i];
  if(wlc[1]<=wlc[2]) {
     min_value=wlc[1];
     select_apds=1;
  }
  else {
     min_value=wlc[2];
     select_apds=2;
  }
  if(wlc[3]<=min_value){
     min_value=wlc[3];
     select_apds=3;
  }
    sendPacket(packets, select_apds);
}
```

(그림 14) PCSLB 알고리즘

3.2.3 APDS 설계

APDS(Application Proxy Data Segment)에서 E-mail, Telnet, FTP, Web 등과 같은 응용 서비스 수준에서 트래픽을 분석하여 외부 네트워크로부터 내부 네트워크로의 진입을 허용 또는 거절할지 결정한다. 즉, APDS에서는 응용서비스 단계에서 Hit ACL 기법을 이용해 액세스 제어를 제공할 수 있고, 응용 서비스의 사용에 대한 로그(log)를 유지하여 감사추적 기능을 제공하며, Hit ACL 기법을 이용해 패킷 필터링에서 허락된 패킷들이 이 APDS에서 다시 필터링 되어 안전한 패킷을 서버로 전달하게 된다.

```
APDS(int slect_apds, char packets[], int cash[];){
  if(check(HitACL_apds, packets){
    switch(select_apds){
      case 1 : cash[1]=cash[1]+1;
               sendProxyServer(packets, select_apds);
               break;
      case 2 : cash[2]=cash[2]+1;
               sendProxyServer(packets, select_apds);
               break;
      case 3 : cash[3]=cash[3]+1;
               sendProxyServer(packets, select_apds);
               break;
    }
  }
  sendProxyServer(char packets[], int select_apds){
    checkTraffic(packets);
  }
}
```

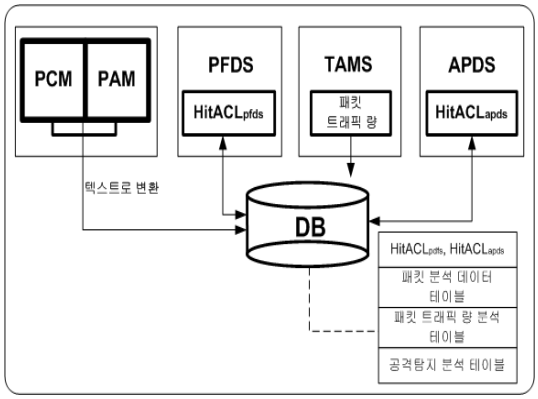
(그림 15) APDS 알고리즘

4. HWbF DB 설계

HWbF DB는 모든 패킷에 대한 자료를 저장할 수 있는 테이블, 로그 변환 모듈에 의해 텍스트 파일로 변경되어 저장되는 테이블, Hit ACL이 저장되어 있는 테이블, 트래픽 분석에 이용되는 패킷들의 헤더 정보를 저장하는 테이블 그리고 로그기록들은 저장하는 테이블이 있다. 또한, TAMS 모듈에 의하여 차단된 공격으로 판단되는 패킷들도 HWbF DB에 저장한다.

HWbF DB는 패킷 분석 엔진과 직접적으로 연동되도록 하여, 탐지와 업데이트 시간을 단축시키도록 설계되었다.

그림 16은 HWbF와 HWbF DB의 데이터 흐름 관계를 설명한 것이다.



(그림 16) HWbF DB

TUI Agent에서 공격을 탐지하기 위하여 패킷량 분석 로그와 패킷 분석 로그를 바탕으로 프로토콜별 트래픽 평균값을 계산할 수 있도록 표 2의 로그 포맷 형식으로 저장되도록 변환하여 저장한다.

(표 2) 패킷 분석 로그 포맷 형식

속성	정의	기타
Agent	각 TUI Agent 식별자	TUI Master로 전송시 프로토콜 식별
Time	패킷 도달시간과 로그 포맷 시간	일정시간동안 증가량과 시간별 데이터 구성에 사용
Source Address	패킷 Soruce Address	
Destination Address	패킷 Destination Address	
Protocol	패킷 종류	
Protocol_ sub	플래그, 타입, 코드	Flooding 패턴 검사에 사용
Soruce Port	소스 포트 번호	
Destination Port	목적지 포트 번호	
Length	패킷 길이	일정한 크기의 패킷에 대한 길이 정보

패킷량 분석 로그는 DoS 공격 발생 시 가장 우선되는 로그이며, 표 3은 패킷량 분석 로그 형식이다.

(표 3) 패킷량 분석 로그 형식

속성	정의	기타
Time	로그 구성 시간	시간대별 패킷의 이동량 검사에 이용
Protocol	패킷의 종류	TCP, UDP, ICMP
Packet 이동량	일정시간 동안 패킷 이동량	

표 4는 공격 탐지 분석 로그의 형식으로, TUI Master는 TUI Agent가 DB에 기록한 패킷량 분석 로그를 받는다. TUI Agent에서 정해진 임계값 이상의 공격 트래픽이 발생하였을 경우 TUI Agent의 식별자, Source Address, Port에 대한 정보 그리고 임계값을 넘어 증가한 패킷량에 대한 정보를 TUI Master에 보고하기 위한 로그 포맷 이다.

(표 4) 공격 탐지 분석 로그 형식

속성	정의	기타
Agent	Agent 식별자	TUI Master로 전송 프로토콜 식별
Time	패킷량 증가를 발견한 시간	패킷 증가 탐지 시간
Analytical Info	패킷 증가 탐지 시간, 증가하는 캐시의 프로토콜 타입과 소스 주소, 포트, 길이 정보	임계치를 초과한 패킷 이동량에 대한 정보를 TUI Master로 보고하는데 사용

표 5는 Hit기법을 이용하는 PFDS 모듈과 APDS 모듈에서 패킷 필터링을 할 때 사용하는 Hit ACL이 저장되어 있는 테이블이다.

(표 5) Hit ACL 형식

속성	정의	기타
ServiceType	서비스 유형	Telnet, FTP, HTTP등의 서비스 종류 지정
UsedServerPort	서버가 사용하는 포트 번호	
UsedClientPort	클라이언트가 사용하는 포트번호	

5. HWbF 시뮬레이션

5.1 로드밸런싱 시뮬레이션

로드 밸런싱은 트래픽 분산 기능을 가지고 있는 기술로서 각 PFDS에 작업을 균등하게 분산하여 PFDS의 성능에 맞게 요청을 처리할 수 있게 하는 목적으로 사용된다.

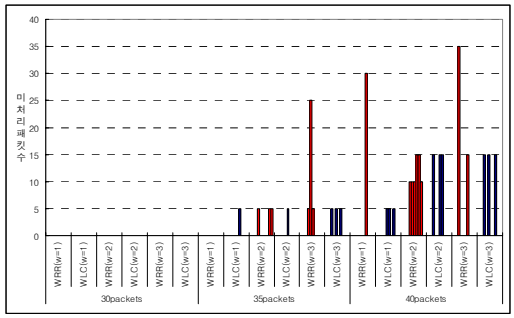
본 논문에서는 트래픽 분산을 가능하게 하는 로드 밸런싱 중 가중 라운드 로빈 알고리즘과 가중치 최소 연결 알고리즘을 기반으로 아래와 같은 가정 하에 시뮬레이션 하여 성능 평가하였다 [11].

가정 1. PFDS 1, PFDS 2, PFDS 3의 초당 패킷 처리 능력을 각각 5, 10, 15패킷이라 하고, PFDS의 가중치는 각각 1, 2, 3이라 한다.

가정 2. TIME_WAIT 시간을 5초로 하여, 사용자가 요청패킷을 보내고 5초 이내에 응답 패킷을 보내지 못하면 패킷을 폐기한다.

가정 3. 초당 요청 패킷들의 범위는 30, 35, 40packet으로 하여 10회 이상 반복 실험 하였다.

그림 17은 가중 라운드 로빈(WRR)과 가중치 최소 연결(WLC) 알고리즘의 미처리 패킷의 수를 그래프로 나타낸 것이다.



(그림 17) WRR, WLC 알고리즘 미처리 패킷 수 비교

그래프의 결과에서처럼 요청 패킷이 많을수록 더욱 미처리 패킷이 많이 발생하게 된다는 것을

알 수 있으며, 같은 조건으로 두 알고리즘의 성능을 실험한 결과 가중 최소할당 알고리즘이 훨씬 많은 요청 패킷을 처리한다는 것을 알 수 있다. 위의 실험 결과에서 가중치 최소 연결 알고리즘을 사용하는 PFDS가 가중 라운드 로빈 알고리즘을 사용하는 PFDS에 비해 최대 1.7배의 패킷 처리능력을 보였다.

따라서 본 논문에서는 가중치 최소 연결 알고리즘을 사용함으로써 기존의 방화벽의 병목현상을 줄이고 좀 더 효율적인 트래픽 분산으로 미처리 패킷의 수를 줄였다.

5.2 TAMS 모듈 시뮬레이션

TAMS 모듈은 libpcap 라이브러리와 tcpdump 프로그램을 응용하여 설계하였으며, DoS 공격은 SYN Flooding, UDP Flooding, ICMP Flooding으로 하였다. TCP, UDP, ICMP 평균 트래픽은 3회 반복하여 수집한 트래픽의 평균값을 계산하였고, 고정임계치는 3회 반복한 평균값 중에서 제일 큰 값으로 지정하였다.

TAMS 모듈의 평균값은 식(1)를 이용해 계산하였고, TAMS 모듈의 임계치는 식(2)를 이용해 계산하였으며, 트래픽 양에 따라 값이 변하는 유동적인 값이다.

$$TAMS_{average} = (TAMS_{traffic} + TAMS_{average})/2 \quad \text{.....식(1)}$$

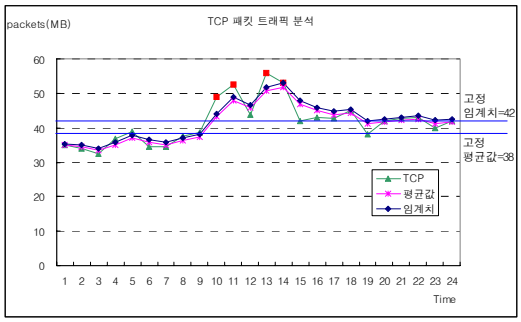
$$TAMS_{criterion} = (TAMS_{traffic} + TAMS_{criterion})/2 + util_{TAMS} \text{...식(2)}$$

DoS 공격 탐지는 트래픽 량이 고정임계치, 고정평균값, 트래픽량에 따라 변하는 TAMS 모듈의 평균값, 임계치를 초과했을 경우를 DoS 공격을 탐지하도록 하였으며, 실험 중에는 총 4번의 DoS 공격을 시행하였다.

5.2.1 TCP

그림 18은 TUI Agent A(TCP)의 트래픽 탐지 결과를 설명한 그래프이다.

TUI Agent A(TCP)의 평균 트래픽이 38MB일때, DoS 공격 탐지율은 100%이지만 오탐지율이 50%(=(12/24)×100)이고 고정임계치가 42MB일때 DoS 공격 탐지율은 100%이지만 DoS 공격 오탐지율이 25%(=(7/24)×100)이다.

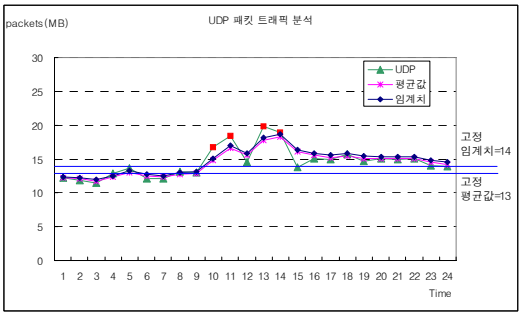


▲ 정상 트래픽 ■ DoS 공격 트래픽,
(그림 18) TUI Agent A의 공격탐지 그래프

하지만, 본 논문에서 설계한 트래픽 양에 따라 변하는 TAMS 모듈의 평균값은 DoS 공격 탐지율이 100%이고 DoS 공격 오탐지율이 38%(=(9/24)×100)이고, TAMS모듈의 임계치는 DoS 공격 탐지율이 100%이고 DoS 공격 오탐지율이 17%(=(4/24)×100)이므로 기존의 고정 평균값과 고정 임계치를 이용한 DoS 공격 탐지 기법에 비해 오탐지율을 감소시켰다.

5.2.2 UDP

그림 19는 TUI Agent B(UDP)의 트래픽 탐지 결과를 설명한 그래프이다.



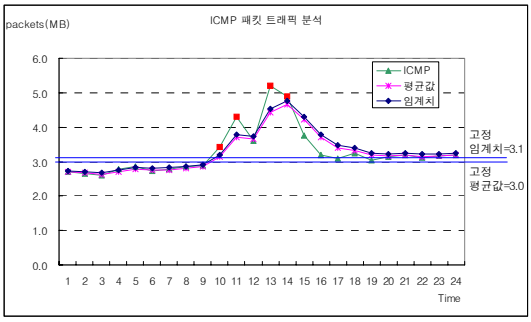
▲ 정상 트래픽 ■ DoS 공격 트래픽,
(그림 19) TUI Agent B의 공격탐지 그래프

TUI Agent B(UDP)의 평균 트래픽이 13MB일때 DoS 공격 탐지율은 100%이지만 DoS 공격 오탐지율은 54%(=(13/24)×100)이고, 고정 임계치가 14MB일때 DoS 공격 탐지율은 100%이지만 DoS 공격 오탐지율은 33%(=(8/24)×100)이다.

하지만, 본 논문에서 설계한 트래픽 양에 따라 변하는 TAMS 모듈의 평균값은 DoS 공격 탐지율이 100%이고 DoS 공격 오탐지율이 25%(=(6/24)×100)이고, TAMS 모듈의 임계치는 DoS 공격 탐지율이 100%이고, DoS 공격 오탐지율은 13%(=(3/24)×100)이므로 기존의 고정 평균값과 고정 임계치를 이용한 DoS 공격 탐지 기법에 비해 오탐지율을 감소시켰다.

5.2.3 ICMP

그림 20은 TUI Agent C(ICMP)의 트래픽 탐지 결과를 설명한 그래프이다.



▲ 정상 트래픽 ■ DoS 공격 트래픽,
(그림 20) TUI Agent C의 공격탐지 그래프

TUI Agent C(ICMP)의 평균 트래픽이 3.0MB일 때 DoS 공격 탐지율은 100%이지만 DoS 공격 오탐지율은 46%(=(11/24)×100)이고, 고정 임계치가 3.1MB일때 DoS 공격 탐지율은 100%이지만 DoS 공격 오탐지율은 38%(=(9/24)×100)이다.

하지만, 본 논문에서 설계한 트래픽 양에 따라 변하는 TAMS 모듈의 평균값은 DoS 공격 탐지율이 100%이고 DoS 공격 오탐지율이 33%(=(8/24)×100)이고, TAMS 모듈의 임계치는 DoS 공격 탐지율이 100%이고 DoS 공격 오탐지율은 8%(=(3/24)×100)이므로 기존의 고정 평균값과 고정 임계치를 이용한 DoS 공격 탐지 기법에 비해 오탐지율을 감소시켰다.

물론, 평균값과 고정 임계치를 좀 더 높게 책정하여 DoS 공격 오탐지율을 낮출 수는 있지만, DoS 공격 탐지율이 낮아지는 문제점이 발생할 수 있다. 따라서 본 논문에서 제안하는 TAMS 모듈의 평균값과 임계치를 사용해 기존의 평균값과 고정 임계치를 사용하는 DoS 공격 탐지의 오탐지율을 TCP 경우 12%, 8%를 감소시켰고, UDP 경우 29%, 20%를 감소시켰고, ICMP 경우 13%, 30%를 감소시켜, DoS 공격 탐지 신뢰도를 향상시켰다.

6. 결 론

본 논문에서는 Hit 기법과 WLC(Weight Least Connection) 알고리즘을 이용해 외부에서의 접근 제어와 트래픽 공격을 차단하는 로드 밸런싱 기반 방화벽인 HWbF을 설계하였다. HWbF는 PLB(Packet Load Balancing) 모듈을 이용해 강력한 패킷 필터로 진화하여 처리속도를 향상시키는 PFS(Packet Filter Station)과 PCSLB(Proxy Cash Server Load Balancing) 모듈을 이용해 프록시 캐쉬 서버를 연동하여 서버에 걸리는 부하를 감소시키고, 패킷 트래픽 량으로 DoS 트래픽을 감지하고, DoS 공격을 차단하는 APS(Application Proxy Station)로 구성된다.

HWbF 시스템은 능동형 네트워크 정보보호시스템으로써 네 가지 측면에서 성능과 보안이 향상되었다.

첫째, PFS(Packet Filter Station)에서의 패킷 캡처 모듈과 분석 모듈을 통한 모니터링 시스템을 설계하고 패킷 필터기능에서 Hit 항목의 업데이트로 인하여 강력한 패킷 필터 기능을 한다.

둘째, 로드 밸런싱 기법을 이용한 PLB를 이용해 패킷 전송 지연을 방지하고 병목 현상을 줄여 패킷 처리 속도를 향상 시켰다.

셋째, APS의 TAMS 모듈을 이용하여 기존의

평균값과 고정 임계값을 이용하는 DoS 공격 탐지 기법 보다 DoS 공격에 대한 오탐지율을 감소시키고, 탐지된 DoS 공격을 Hit 기법으로 차단하였다.

넷째, APS 방화벽의 PCSLB 모듈로 프록시 캐쉬 서버의 연동으로 빠른 시간 내에 데이터를 처리함으로써 서버의 부하를 감소시키고 자원의 원활한 활용이 가능하다.

참 고 문 헌

[1] 이병관, 정은희, “인터넷 보안”, 남두도서, 2004

[2] 강유, 정수현, “강유의 해킹&보안 노하우”, 에이콘, 2003

[3] W. Zhang, “Linux Virtual Sever for Scalable Network Services”, Ottawa Linux Symposium, 2000. http://www.linuxvirtualserver.org/lvs.pdf

[4] Y. M. Teo, R. Ayani, “Comparison of Load Balancing Strategies on Cluster-based Web Server”, Transaction of the Society for Modeling and Simulation, 2001.

[5] 한국정보보호진흥원, “IPv6 보안 기술 해설서”, Oct. 2005

[6] 이종엽, 윤미선, 이훈, “DoS 공격의 유형 분석 및 탐지 방법”, KNOM Review, Vol. 6, No.2, pp21-32, Feb. 2004

[7] 송병학, 홍충선, “향상된 통계기반 분산 서비스 거부(DDoS) 공격 탐지 시스템”, 한국정보처리학회 춘계학술발표대회논문집, 제13권 제1호, 2006.5.

[8] 노광민, “리눅스에서 pcap library를 사용하여 패킷을 잡아보기 v.03″, 리눅스 한글 문서 프로젝트, 2000.9.14

[9] 전용희, “홈 네트워크 보안 관련 기술”, 한국통신학회, 2004.3

[10] 윤미진, “네트워크 이용율에 기반한 DoS 트래픽 탐지 기법”, 석사 학위 논문, 조선대학교, 2004

[11] 정은희, 최은실, 이병관, “IPSec 프로토콜 성능 향상과 LSNAPT 설계에 관한 연구”, 한국인터넷정보학회논문지, 제5권 6호, pp.45-58, 2004.12.

[12] L. Graber, “Denial-of-Service Attacks Rip the Internet”, IEEE Computer(p.12-17), April 2000.

[13] CERT Advisory CA-96.21, “TCP SYN Flooding and IP Spoofing Attacks”, September 24. 1996

[14] William Cheswick & Steven Bellovin, ″Firewalls and Internet Security″, 1994

[15] 권동혁, 장재열, 이병관, “DoS 공격의 효율적인 탐지와 처리속도 향상을 위한 CLB(Cashing and Load Balancing)”, 한국인터넷정보학회 추계학술발표대회, 제 7권 제 2호, pp.149-153, 2006.11.

[16] 이철호, 최경희, 정기현, 노상욱, “웹서버에 대한 DDoS 공격의 네트워크 트래픽 분석”, 한국 정보처리학회 논문지, 10-C(3), pp.253-263, 2003.06.

◐ 저 자 소 개 ◑

이 병 관

1975년 부산대학교 기계설계학과 졸업(이학사)
1986년 중앙대학교 대학원 전자계산공학과 졸업(석사)
1990년 중앙대학교 대학원 전자계산공학과 졸업(박사)
1988년 3월~현재 관동대학교 컴퓨터학과 교수
관심분야 : 네트워크 보안, 컴퓨터 네트워크, 전자상거래, etc.
E-mail : bklee@kwandong.ac.kr

권 동 혁

2005년 강릉대학교 컴퓨터공학과 졸업(공학사)
2007년 관동대학교 대학원 전자계산공학과 졸업(석사)
2007년 3월 ~ 현재 관동대학교 대학원 전자계산공학과 재학 중(박사과정)
관심분야 : 네트워크 보안, 컴퓨터 네트워크. 전자상거래, etc.
E-mail : kdh0108@kwandong.ac.kr

정 은 희(교신저자)

1991년 강릉대학교 통계학과 졸업(이학사)
1998년 관동대학교 대학원 전자계산공학과 졸업(석사)
2003년 관동대학교 대학원 전자계산공학과 졸업(박사)
2003년 9월~현재 강원대학교 지역경제학과 조교수
관심분야 : 네트워크 보안, 전자상거래. 웹 프로그래밍, etc.
E-mail : jeongeh@kangwon.ac.kr