DOI: 10.3745/KIPSTC.2009.16-C.1.13

SNMP 기반의 실시간 트래픽 폭주 공격 탐지 시스템 설계 및 구현

박 준 상[†] · 김 성 윤[†] · 박 대 희[††] · 최 미 정[†††] · 김 명 섭[††††]

요 약

DoS/DDoS공격과 웜 공격으로 대표되는 트래픽 폭주 공격은 그 특성상 사전 차단이 어렵기 때문에 정확하고 빠른 탐지에 의한 대처는 공격 탐지 시스템이 갖추어야 할 필수요건이다. 본 논문에서는 SNMP MIB의 다양한 상관관계 분석을 통해 빠르고 정확한 탐지 알고리즘을 제안하고, 이를 적용한 실시간 탐지 시스템을 구현하였다. 공격 탐지 방법은 SNMP MIB의 갱신 주기를 이용하여 공격 탐지 시점을 결정하는 단계와 수신된 패킷의 상위 계층 전달률, 수신된 패킷에 대한 응답률, 그리고 폐기된 패킷 개수와 같은MIB 정보간의 상관 관계를 이용하여 공격의 징후를 판단하는 단계, 프로토콜 별 상세 분석을 통하여 공격 유무 탐지 및 공격 유형 분류를 수행하는 단계로 구성된다. 제안한 탐지 방법은 빠른 탐지로 발생되는 시스템 부하와 관리를 위한 소비 트래픽의 증가 문제를 효율적으로 해결하여 다수의 탐지 대상 시스템의 관리가 가능하며, 빠르고 정확하게 공격의 유무를 탐지하고 공격 유형을 분류해 낼 수 있어 공격에 대한 신속한 대처가 가능해 질 수 있다.

키워드 : 트래픽 폭주 공격, DoS/DDoS, SNMP, MIB, 탐지 알고리즘, 탐지 시간, 탐지 시스템

Design and Implementation of an SNMP-Based Traffic Flooding Attack Detection System

Jun-Sang Park[†] · Sung-Yun Kim[†] · Daihee Park[††] · Mi-Jung Choi[†††] · Myung-Sup Kim[††††]

ABSTRACT

Recently, as traffic flooding attacks such as DoS/DDoS and Internet Worm have posed devastating threats to network services, rapid detection and proper response mechanisms are the major concern for secure and reliable network services. However, most of the current Intrusion Detection Systems (IDSs) focus on detail analysis of packet data, which results in late detection and a high system burden to cope with high-speed network traffic. In this paper we propose an SNMP-based lightweight and fast detection algorithm for traffic flooding attacks, which minimizes the processing and network overhead of the detection system, minimizes the detection time, and provides high detection rate. The attack detection algorithm consists of three consecutive stages. The first stage determines the detection timing using the update interval of SNMP MIB. The second stage analyzes attack symptoms based on correlations of MIB data. The third stage determines whether an attack occurs or not and figure out the attack type in case of attack.

Keywords : Traffic Flooding Attack, DoS/DDoS, SNMP, MIB, Detection Algorithm, Detection Time, Detection System

1. 서 론

인터넷의 발달은 사용자에게 네트워크를 통한 다양하고 빠른 서비스를 제공받을 수 있는 환경을 제공하였다. 이와 같이 네트워크 기반 서비스에 대한 의존도가 증가하면서 유해 트래픽 탐지 및 차단은 안전한 서비스 제공을 위해 필수 불가결한 요건이 되었다. 대표적인 유해 트래픽인 트래픽 폭주 공격이 발생하면 컴퓨터 시스템은 물론 네트워크까지 마비시켜 업무에 엄청난 영향을 미치고 막대한 피해를 주게 된다. 따라서 트래픽 폭주 공격탐지에 대한 연구의 중요성이 날로 커지고 있다.

일반적인 공격탐지 방법인 패킷 수집을 통한 트래픽 폭주 공격 탐지[1]는 상세한 분석이 가능하지만 고가의 고성능 분석시스템이 요구되고 설치 및 운영의 확장성이 부족한 단점을 가지고 있다. 이를 보완하기 위한 방법으로 SNMP를 이용한 탐지 방법[2,4,6,7,8]이 효과적으로 사용될 수 있다.

※ 이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2007-331-D00387)
† 준 회 원 : 고려대학교 컴퓨터정보학과 석사과정
†† 정 회 원 : 고려대학교 컴퓨터정보학과 교수
††† 정 회 원 : 강원대학교 컴퓨터과학과 조교수
†††† 정 회 원 : 고려대학교 컴퓨터정보학과 조교수
논문접수 : 2008년 7월28일
수 정 일 : 1차 2008년 10월 17일
심사완료 : 2008년 10월20일

SNMP MIB정보를 이용한 공격탐지는 MIB 데이터 수집을 위한 시스템 및 네트워크 리소스의 사용이 적고, 계층과 프로토콜을 기준으로 표준화된 네트워크 성능 관리 데이터를 제공 받을 수 있기 때문에 효과적인 탐지가 가능하다.[6,7]

본 논문은 SNMP MIB 객체의 상관관계 분석을 통한 트래픽 폭주 공격 탐지 방법 및 탐지 시간을 향상 시키는 방법을 제안한다. 탐지 시간의 향상에 따른 짧은 탐지 주기와 탐지 시간 향상 과정에서 관리 시스템의 부하와 관리를 위한 소비 트래픽이 증가하는 문제점이 발생한다. 이와 같은 문제를 해결하기 위해 계층적 구조의 탐지 방법과 탐지 시스템 동작 시점을 결정하는 과정에서MIB의 다음 갱신 시점을 예측하는 방법을 적용하였다.

본 논문은 다음과 같은 순서로 구성된다. 2장에서는 트래픽 폭주 공격에 대한 기존 연구와 문제점을 기술하고, 3장에서는 SNMP MIB 상관관계를 기초로 한 트래픽 폭주공격 탐지 알고리즘을 제시한다. 4장에서는 제안된 알고리즘을 바탕으로 구축한 탐지 시스템의 구조와 사용자 인터페이스를 설명하며, 5장에서는 실험을 통해 제안한 알고리즘의 타당성을 증명한다. 6장에서는 결론과 향후 연구에 대하여 기술한다.

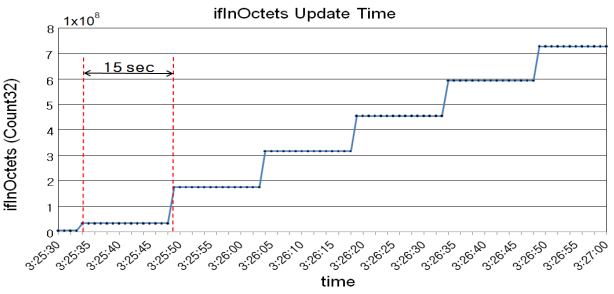
2. 관련 연구

SNMP기반 탐지 시스템의 탐지율과 탐지 시간을 향상시키고, 시스템 및 네트워크의 부하를 줄이는 효율적인 탐지 알고리즘을 설계하고 탐지 시스템의 구축을 위해서는 다음과 같은 4 가지 요소가 고려되어야 한다.

첫째, 공격을 탐지하기 위해 사용하는 MIB의 선정이다. MIB의 선정은 탐지 알고리즘의 탐지율을 결정짓는 중요한 요소이다. 기존의 탐지 방법[4]에서는 *tcpInErrs*와 *udpNoPorts* 객체들을 사용하여 공격을 탐지하였으나 실험 결과 *tcpInErrs*와 *udpNoPorts*객체가 변화하지 않았다. 이는 공격 도구가 견고해 짐에 따라 사전에 Port Scanning 작업이 이루어 지고, 오류가 없는 패킷을 전송하기 때문으로 판단된다. 또한 *icmpOutEchoReps*는 정상적인 ICMP Echo요청에도 반응한다. 이와 같이 MIB의 선정은 탐지 알고리즘의 정확성을 좌우하게 된다.

둘째, 선정된 MIB의 활용방법이다. 기존의 탐지 방법에서 선정된MIB은 트래픽 추이를 통한 분석[4], 통계적 분석[6], 학습 알고리즘을 통한 분석[9] 등 다양한 방법으로 활용되고 있다. 이러한 접근 방법은 과거의 트래픽 현황을 바탕으로 트래픽 폭주 공격을 탐지하게 된다. 하지만 유동적이고 변화가 많은 네트워크 트래픽의 특성상 사용 환경에 따라 탐지 알고리즘의 탐지율이 저하된다.

셋째, 트래픽 폭주 공격이 분산화되고, 고속화됨에 따라 빠른 탐지는 탐지 시스템의 필수요소가 되었다. (그림 1)은 공격 발생 시 interface 그룹의 *ifInfOctets*값을 매 1초단위로 보여 준다. 그림에서 알 수 있듯이 SNMP MIB 값은 선형적인 증가를 보이지 않고, 특정 주기를 기준으로 갱신됨을



(그림 1) *ifInOctets*의 갱신 주기

알 수 있다.[10] 이러한 특징으로 인해 Timer를 기준으로 탐지 시스템이 수행되는 기존의 탐지 방법[4]은 MIB의 갱신 주기를 반영하지 않기 때문에 수집하는 MIB 정보의 정확성을 보장할 수 없다.

넷째, 1대의 관리 시스템은 다수의 타깃 호스트를 관리하기 때문에 관리 시스템의 부하를 최소화하고, 해당 네트워크에 영향을 미치지 않도록 관리를 위한 소비 트래픽을 효율적으로 줄이는 탐지 알고리즘이 요구된다.

본 논문에서는 SNMP MIB의 갱신 주기를 이용하여 탐지 알고리즘의 탐지 시점을 결정함으로써 탐지 시간과 탐지율을 향상시키고, 다양한 MIB간의 상관관계를 이용하여 공격의 징후를 판단하며 프로토콜 별로 제공되는 MIB의 변화 유무를 통해 트래픽 폭주 공격을 탐지하는 알고리즘을 제안한다.

3. 탐지 시간 향상 및 탐지 알고리즘

본 장에서는 SNMP MIB 객체의 상관관계를 통한 트래픽 폭주 공격을 탐지하는 알고리즘과 탐지 시간을 향상하기 위한 방법을 설명한다.

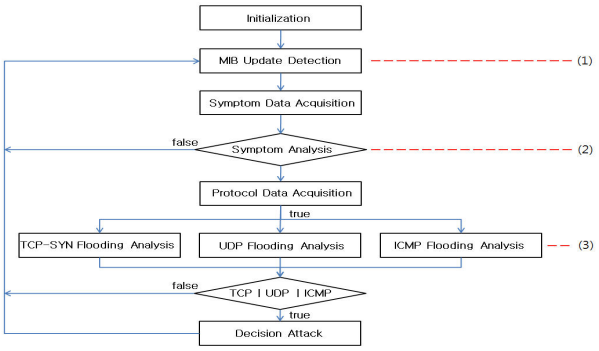
<표 1>은 본 논문에서 사용된 MIB 객체들이다. 탐지 알고리즘에 사용된 MIB 객체들은 모든 SNMP agent에서 공통으로 제공되는RFC1213[3]에서 정의된 MIB-II그룹의 MIB

〈표 1〉 탐지 알고리즘에 사용된 MIB

MIB-2 Group	SNMP MIB objects
system	system.sysUpTime
interface	interface.ifTable.ifEntry.*ifInOctets* interface.ifTable.ifEntry.ifInUcastPkts
IP	ip.*ipInReceives* ip.*ipInDelivers* ip.*ipOutRequests* ip.ipOutDiscards
TCP	tcp.*tcpAttemptFails* tcp.tcpOutRsts
UDP	udp.udpInErrors
ICMP	icmp.*icmpInMsgs* icmp.*icmpInDestUnreachs* icmp.*icmpInEchos* icmp.*icmpOutMsgs* icmp.*icmpOutDestUnreachs* icmp.*icmpOutEchoReps*

〈표 2〉 탐지 알고리즘에 사용되는 기호 및 수식

t	1초 단위의 시간
t_n	n 번째 탐지 알고리즘 적용 시간
mib(t_n,oid)	시간 tn 에 수집한 SNMP oid 객체의 값
diff(t_n, oid)	= mib(tn,oid) − mib(tn−1,oid)
bps(t_n)	= 800 * diff(tn,ifInOctets) / diff(tn, sysUpTime)
pps(t_n)	= 100 * diff(tn,ifInUcastPkts) / diff(tn, sysUpTime)
DeliverRatio(t_n)	= diff(tn, ipInDelivers) / diff(tn, ipInReceives)
ResponseRatio(t_n)	= diff(tn,*ipOutRequests*) / diff(t_n, *ipInReceves*)



(그림 2) 트래픽 폭주 공격 탐지 흐름도

객체들로 구성되었다. MIB 객체의 선택은 각 그룹의 MIB 객체의 의미와 트래픽 폭주공격 패킷과의 상관관계 및 실제 공격에 반응하는 MIB 객체들의 전수조사를 통하여 선정되었다.

<표 2>는 본 논문에서 제안하는 탐지 알고리즘에 사용되는 기호 및 수식들을 정리한 것이다. 시간 t는1초 단위의 시간이고, oid는 <표 1>에서 정의한 SNMP MIB 객체를 대표한다.

(그림 2)는 공격 탐지 알고리즘의 흐름도를 나타내고 있다.

Initialization 단계에서는 탐지 시스템을 운영하기 위해 사용되는 SNMP MIB 갱신주기의 초기값을 설정한다. 탐지 시점 결정 단계(1)에서는 Exponential average를 적용하여 현재까지의 갱신주기를 바탕으로 다음 갱신시점을 예측하여 해당 타겟 시스템에 대한 시스템의 동작을 정지하고, 다음 갱신시점에 탐지시스템을 동작시킨다. 공격 징후 판단 단계(2)에서는 <표2>와 같은 MIB의 상관 관계를 파악하여 공격의 가능성을 판단한다. 이 단계에서 대부분의 정상 트래픽을 분류하고, 세부분석 단계의 실행을 줄임으로써 관리 시스템의 부하를 감소시킨다. 공격의 가능성이 보이면 프로토콜 별로 세부 분석(3)이 이루어진다. 이때 tcp, udp, icmp 그룹으로부터 수집된 데이터를 기준으로 공격 유무를 판단한다. 공격 트래픽으로 판단되면 최종적으로 각 프로토콜 별

〈표 3〉 탐지 시점 결정 알고리즘에 사용되는 수식

U_n	n 번째 *ifInOctets* MIB의 갱신 주기
U_{min}	최소 갱신 주기
A_n	n 번째 갱신 주기의 평균 값
α	= 1/2 (상수)
S_n	n 번째 sleep 시간 (1초 단위)

MIB을 바탕으로 공격의 유형을 분류하고, 다음 탐지 시점을 결정하기 위해 MIB의 갱신 시점을 탐지하게 된다.

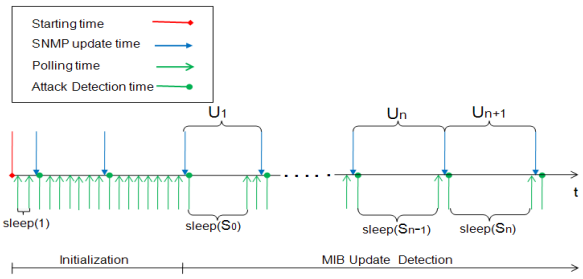
3.1 탐지 시점 결정 알고리즘

탐지 시점 결정 단계에서는 탐지 시간 향상을 위해서 SNMP MIB의 갱신 시점을 찾아내고, 갱신 시점을 기준으로 공격의 탐지 시점을 결정한다. <표 3>은 탐지 시간 향상을 위한 알고리즘에 사용되는 수식들을 정리한 것이다.

탐지 시점 결정 알고리즘의 핵심은 SNMP MIB의 갱신 시점 직후 공격 탐지 시스템을 실행하는 것이다. 하지만 정확한 갱신 시점을 탐지하기 과정에서 관리 시스템의 부하가 커지고, 공격 탐지를 위한 소비 트래픽이 증가하는 문제가 발생한다. 또한SNMP Agent 에 따라 SNMP MIB의 갱신 주기가 유동적이기 때문에 정적인 탐지 시점의 결정방법은 다수의 타깃 시스템에 적용하기 어려운 문제점이 발생한다. 이와 같은 문제점들을 해결하고 정확한 갱신 시점을 찾는 탐지 시점 결정 알고리즘을 제시한다.

(그림 3)은 공격 탐지 시점 결정 알고리즘을 도식화한 것이다. Initialization 단계에서 매 1초 주기 Polling 을 통해서 SNMP MIB값의 갱신 시점을 찾고, 갱신 시점에서 탐지 시스템을 가동한다. 이 단계에서 찾아낸 갱신 주기의 최소값(U_{min})과 마지막으로 탐지시스템을 적용한 시간(t_{old})을 초기값으로 하여 탐지 시스템을 운영한다. MIB 갱신시점 탐지 단계에서는 다음 갱신 시점을 예측하여 그 시간(S)만큼 시스템을 sleep시킴으로써 시스템의 부하와 트래픽의 사용량을 줄인다. 예측을 위하여 현재까지 측정된 갱신주기들의 exponential average(A)를 이용하며, sleep시간(S)을 아래의 수식과 같이 결정한다.

$$A_n = \alpha \times U_n + (1-\alpha) \times A_{n-1}$$
$$S_n = A_n - 1$$



(그림 3) 탐지 시점 결정 알고리즘

3.2 공격 징후 판단

트래픽 폭주 공격은 대량의 패킷을 전송함에 기초한다. 따라서 공격이 발생하면 BPS(Bits per Second)나 PPS(Packets per Second)값이 일정 수준 이상 유지된다. 이러한 사실을 바탕으로 탐지 알고리즘의 수행 여부를 결정한 후 ip 그룹의 MIB 간의 상관 관계를 나타내는 DeliverRatio(), ResponseRatio(), diff()값을 통해 공격의 가능성을 판단한다. 알고리즘 1과 같이 공격 징후를 판단한다.

Algorithm1. Symptom_analysis

```
1: Boolean Symptom_analysis( … ){
2:     int weight = 0;
3:   if(bps < Th(bps) && pps < Th(pps)) return FALSE;
4:   if( DeliverRatio(t) < Th(DeliverRatio) )  weight++;
5:   if( ResponseRatio(t)<Th(ResponseRatio) ) weight++;
6:     if(Diff(t,ipOutDiscards) > Th(ipOutDiscards) ) weight++;
7:   if(weight > =1) return TRUE;
8:   return FALSE;
9:  }
```

상위 계층 전달률(DeliverRatio)은 인터페이스를 통해 받은 패킷이 상위 계층에 전달되는 정도를 나타낸다. 정상적인 트래픽의 경우 전달률은 0.8 이상의 전달률을 보였지만 공격이 발생하면 버퍼 공간의 부족과 오류로 인해 0.3 이하의 전달률을 보였다.

응답률(ResponseRatio)은 상위 계층으로 전달된

데이터그램의 개수 즉*ipInDelivers*대한 응답률을 나타낸다. 정상적인 트래픽의 경우 응답률이 0.5 이상의 응답률을 보이지만 폭주 공격이 발생하면 0.4이하의 응답률을 보인다.

폐기된 패킷 개수를 나타내는*ipOutDiscards* 객체는 인터페이스를 통해 받은 패킷을 버퍼공간의 부족으로 인해 버려지는 경우에 증가한다. 폭주 공격은 오류가 없는 대량의 패킷을 전송하기 때문에 거의 대부분의 패킷이 수신되지만 응답되는 과정에서 버퍼공간의 부족으로 패킷을 폐기시킨다.

공격의 징후 판단 단계에서 사용되는 임계치(Threshold)는 <표 4>와 같이 결정된다.

<표 4> 공격 징후 판단 단계 임계값 설정

Th(bps)	1M	Th(pps)	20
Th(DeliverRatio)	0.8	Th(ResponseRatio)	0.4
Th(ipOutDiscards)	0		

3.3 세부 프로토콜 탐지 및 유형 분석

특정 공격 트래픽이 발생하면 tcp, udp, icmp MIB 그룹의 객체가 반응을 보이기 때문에 공격의 유무에 대한 탐지와 공격 유형에 대한 분류가 가능해진다. 세부 프로토콜 별 탐지를 위해 선정된 MIB은 공격 트래픽에 의해 급격히 증가하지만 정상 트래픽의 경우에도 오류에 의해 변화가 발생할 수 있다. 따라서 각 프로토콜 분석 알고리즘에 임계치을 적용하여 이러한 문제를 해결했다. 이때 사용된 임계치는 10일간의 정상 트래픽에 대한 통계치를 기준으로 아래의 <표 5>와 같이 설정하였다.

<표 5> 세부 프로토콜 탐지에 사용된 임계치 값

TCP	Th(*tcpAttamptFail*) = 10	Th(*tcpOutRsts*) = 10
UDP	Th(*udpInErrs*) = 10	
ICMP	Th(*icmpInMsgs*) = 100	Th(*icmpInEchos*) = 100

3.3.1 TCP-SYN Flooding 공격 탐지

TCP SYN Flooding 트래픽을 발생 시켰을 경우 *tcpAttamptFail*과 *tcpOutRsts*객체가 반응을 보였다. *tcpAttamptFail*은 목적지 포트가 닫혀있는 경우와 RST패킷을 받은 경우 연결이 실패되어 증가된다. *tcpOutRsts*객체는 SYN 패킷을 받은 타깃 호스트의 해당 포트가 닫혀있는 경우 목적지 포트에 RST패킷을 전송하기 때문에 증가된다. TCP Flooding 공격을 탐지하는 단계에서는 알고리즘2와 같이 *tcpAttampFail*과 *tcpOutRsts* 객체의 반응 유무를 통해 탐지할 수 있다.

Algorithm2. TCP SYN_Flooding_analysis

```
1: Boolean TCP-SYN_Flooding_analysis( … ){
2:   if( Diff(t, tcpAttamptFail) > Th(tcpAttamptFail)  ||
       Diff(t, tcpOutRsts )> Th(tcpOutRsts) ) )
3:       return TRUE;
4:   return FALSE;
5:  }
```

3.3.2 UDP Flooding 공격 탐지

UDP Flooding 공격이 발생하면 타깃 호스트는 대량의 패킷을 전송받기 때문에 버퍼 공간의 부족으로 *udpInErrs* 객체가 증가된다. 또한 *udpInErrs*는 체크섬의 오류에 의해서도 증가된다. 또한 UDP Flooding 공격이 발생하면 상위 프로토콜에 전달되지 않거나 포트가 닫혀있는 등의 오류가 발생하여 오류보고 메시지를 전달하고, 전송받기 때문에 *icmpOutDestUnreachs*객체를 증가 시킨다. 따라서 *icmpOutDstUnreachs*와 *icmpOutMsgs*는 같은 변화량을 보이게 되므로 알고리즘 3과 같이 UDP Flooding 공격을 탐지한다.

Algorithm 3. UDP_Flooding_analysis

```
1:  Boolean UDP_Flooding_analysis( … ) {
2:    int weight = 0;
3:    if( Diff(t, udpInErrs ) > Th(udpInErrs)) weight ++;
4:    if(Diff(t, icmpOutDstUnreachs)==diff(t,icmpOutMsgs) &&
        Diff(t,icmpOutDstUnreachs) > 0 &&
        Diff(t, icmpOutMsgs) > 0 &&
        Diff(t, icmpInDstUnreachs) == 0) weight++;
5:
6:    if(weight > =1) return TRUE;
7:    return FALSE;
8:  }
```

3.3.3 ICMP Flooding 공격 탐지

ICMP Flooding 공격이 발생하면 ICMP Echo Request 에 의해*icmpInEchos*와 함께*icmpInMsgs*가 급격히 증가한다. ICMP Echo request를 IP 계층에서는 모두 수용하기 때문에 *ipInDiscards*는 증가하지 않지만 버퍼 공간의 부족으로 수신된 모든 패킷에 대해 응답이 이루어지지 않고 폐기되는 패킷이 발생한다. 이는 *ipOutDicards* 값을 증가 시키며, 정상적으로 응답 패킷이 보내진 경우에는 *i*cmpOutmsgs, *icmpOutEchoReps* 객체가 증가된다. 또한 상위 프로토콜에 전달되지 않거나 포트가 닫혀있는 등의 오류가 발생하여 오류보고 메시지를 송수신하기 때문에 *icmpInDestUnreachs*, *icmpOutDestUnreachs*객체를 증가 시킨다. 따라서 알고리즘 4와 같은 알고리즘으로 ICMP Flooding 공격을 탐지한다.

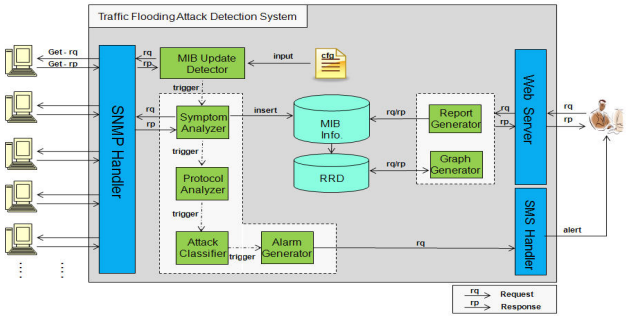
Algorithm 4. ICMP_Flooding_analysis

```
1:  Boolean ICMP_Flooding_analysis( … ){
2:      int weight=0;
3:      if( Diff(t, icmpInMsgs)  > Th(icmpInMsgs)  ||
          Diff(t, icmpinEchos) > Th(icmpInEchos) ) weight++;
4:      if( Diff(t, icmpInDstUnreachs)  > 0 &&
          Diff(t, icmpOutDstUnreachs) > 0 &&
          Diff(t, icmpOutEchoReps) > 0  &&
          Diff(t, ipOutDiscards > 0) ) weight++;
5:      if(weight > =1) return TRUE;
6:      return FALSE;
7:  }
```

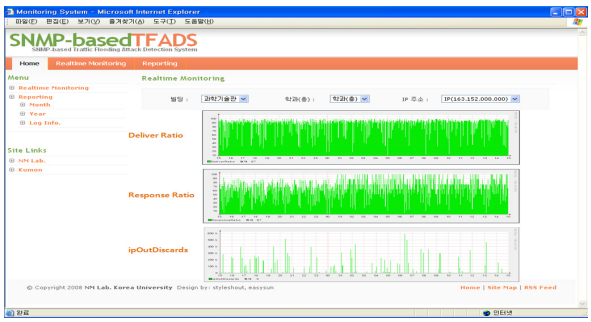
4. 탐지 시스템 구현

본 논문에서 제안한 알고리즘을 바탕으로 트래픽 폭주 공격 탐지 시스템을 구축하였다. 탐지 시스템은 각 타깃 시스템의 SNMP Agent로부터 수집된 데이터를 통해 공격을 탐지하고 Web 기반으로 관리자에게 탐지 정보 현황을 제공한다. 공격 발생 시 SMS를 통해 관리자에게 보고된다.

(그림 4)는 탐지 시스템의 전체 구조를 표현한다. 탐지 시스템은 탐지 시점 결정 모듈, 탐지 모듈, 알람 생성 모듈, MIB Info.저장 모듈, RRD 저장 모듈, Graph generator, 웹 페이지 생성 모듈로 구성된다. 탐지 시스템은 각 타깃 시스템을 기준으로 프로세스를 할당하고 해당 프로세스는 탐지



(그림 4) 시스템 구조



(그림 5) 탐지 시스템 페이지 구성

에 필요한 SNMP MIB 정보의 수집, MIB 정보 저장, 공격 탐지까지의 일련의 과정을 관장한다. 수집된 MIB 정보는 타깃 시스템 단위로 관리되며 정보를 제공한다. 그래프를 생성하기 위해 MIB Info. DB는 RRD DB 형태로 다시 저장된다.

(그림 5)는 공격 탐지 시스템에서 웹 페이지로 보여주는 정보이다. 웹 서버를 통해 각 타깃 시스템에 대한 관리자의 요청이 있으면 Report Generator는 DB에 저장된 정보를 읽어 들여 보여준다. RRD를 통해 전달률(DeliverRatio), 응답률(ReponseRatio), ipOutDiscards() 결과를 그래프 형태로 제공한다. 또한 공격에 대한 로그 정보와 통계적 정보를 관리자의 요청에 의해 제공한다. 공격이 탐지되면 타깃 시스템 정보, 공격 발생 시각, 공격 유형에 대한 정보를 관리자에게 SMS를 통해 실시간으로 제공한다.

5. 실험 및 결과

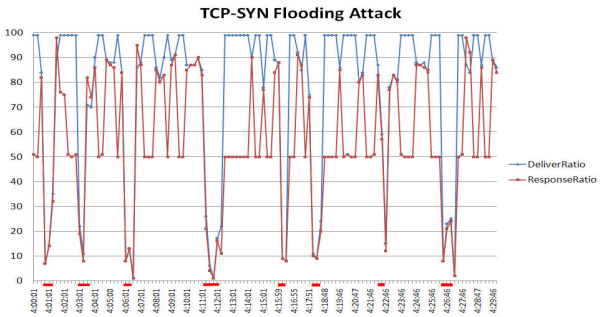
본 논문에서 제안한 알고리즘의 실험을 위하여 공격 호스트 3대, 타깃 호스트 1대, 관리 시스템 1대, 탐지 시스템 1로 구성하였다.

실험은 TCP-SYN, UDP, ICMP Flooding공격을 Stacheldraht[5]를 이용하여 공격그룹 호스트에서 타깃 호스트로 각각 실시하였다. MIB 갱신 주기의 평균이 15초인 시스템을 타깃 시스템으로 하여 10일 동안 무작위로 TCP-SYN, UDP, ICMP Flooding 공격을 실시했으며, 특정 시간에 하나의 공격만을 실시하였다. 공격 시점과 공격 지속시간은 무작위로 설정하였다. 알고리즘의 성능 평가는 TCP-SYN Flooding 공격을 중심으로 기술한다.

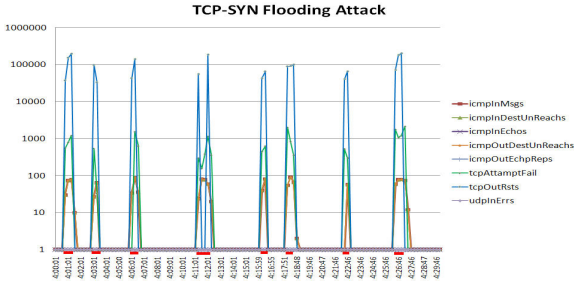
5.1 TCP-SYN Flooding 공격 탐지

TCP-SYN Flooding 공격 발생시 각 단계에서 사용된 MIB 값들의 변화를 통해서 탐지 알고리즘의 타당성을 설명한다.

(그림 6)은 TCP-SYN Flooding 공격 시 DeliverRatio 값과 ResponseRatio값의 변화를 나타낸다. 공격이 이루어진 시점에서 DeliverRatio 값이 0.8미만의 값을 가지고, Response Ratio값은 0.4 미만의 값을 가짐을 확인할 수 있다. TCP-SYN Flooding공격이 발생하면 DeliverRatio와



(그림 6) DeliverRatio, ResponseRatio 변화



(그림7) 공격 탐지를 위한MIB값의 변화

ResponseRatio에 의해서 공격의 징후로 판단하고 세부 분석 단계에 진입한다.

(그림 7)은 TCP-SYN Flooding 공격 시 *icmpInMsgs*, *icmpInDestUnReachs*, *icmpInEchos*, *icmpOutDestUnReachs*, *icmpOutEchoRep*, *tcpAttamptFail*, *tcpOutRsts*, *udpInErrs*값의 변화를 나타낸 것이다. 각 공격이 이루어진 시점에서 *tcpAttamptFail*과 *tcpOutRsts* 객체 값이 급격히 증가함을 알 수 있다. 따라서 TCP-SYN Flooding analysis 알고리즘에 의해 공격으로 판단하고, TCP SYN Flooding공격으로 분류한다. 반면 UDP Flooding 분석과 ICMP Flooding분석에서는 *udpInErr*, *icmpOutMsgs*, *icmpOutEchoReps*의 값이 변화가 없기 때문에 UDP 및 ICMP 공격으로 탐지되지는 않는다.

실험 결과 TCP-SYN Flooding 공격의 경우, DeliverRatio와 ResponseRatio의 값에 의해 공격징후로 판단되며 세부 분석단계에서 *tcpAttamptFail*과 *tcpOutRsts* 값에 의해서 TCP-SYN Flooding 공격으로 탐지되고 분류됨을 알 수 있다

5.2 관리 시스템 성능 평가

트래픽 폭주 공격 발생 시 제안된 탐지시간 향상 알고리즘은 최악의 경우 18초의 탐지 시간을 보였고 평균 탐지 시간은 8.23초가 요구되었다. 이는 관리 대상 시스템의 안전성을 보장하기에 충분한 시간으로 판단된다.

<표 6>은 공격 탐지 주기가 짧아짐에 따라 발생되는 관리 시스템의 부하와 탐지를 위한 소비 트래픽의 부하에 대한 성능평가를 보여준다.

제안된 방법은 시스템 부하에 거의 영향을 미치지 않음을 확인할 수 있고, 갱신 주기 및 공격 탐지를 위한 SNMP 메시지에 의한 트래픽 부하 역시 거의 없음을 알 수 있다. 이

〈표 6〉 시스템 및 네트워크 부하 평가

평가 항목		Proposed Method	
CPU Usage (Pentium D 3.40GHz)		< 0.1%	
Memory Usage (512 MB)		< 0.3%	
Network Overhead	Inbound	122 bps	0.2 pps
	Outbound	113 bps	0.2 pps

러한 결과는 제안된 방법론을 이용하여 다수의 네트워크 장비 및 시스템에 대해 탐지가 가능하다는 것을 보여준다.

5.3 탐지 알고리즘 성능 평가

<표 7>은 알고리즘의 성능 평가를 위해 수집된 실험데이터의 내용을 보여주고 있다.

공격 징후 판단 단계에서는 49732의 정상데이터 탐지 횟수 중 48441회를 정상 트래픽으로 분류하여 97.51%의 정상 트래픽에 대해서는 세부 분석단계 분석을 위한 데이터의 수집을 줄일 수 있었다. 공격 징후 판단 단계에서 3가지 공격의 경우를 모두 포함하여 9199회를 공격 징후로 판단하였다. 공격 징후 판단 단계에서 공격 트래픽은 모두 공격의 징후로 판단하였고 정상 트래픽은 전체 정상 트래픽 중 1291회만을 포함하였다. 48441회의 정상 트래픽은 세부 분석 단계를 생략할 수 있었다.

공격 횟수를 기준으로 탐지 성능을 평가하면 제시된 알고리즘은 <표 7>과 같이 TCP-SYN, UDP, ICMP 공격을 100% 정확하게 탐지하고 유형을 분류하였다. 탐지는 공격이 시작되고 나서 1~2회의 탐지시점 내에 모두 이루어졌는데, 최초 탐지에서 오탐지가 발생한 경우는 공격시작시점과 탐지시점의 시간차가 3초 이내인 경우로 분석되었다.

<표 7>에 나타난 바와 같이 탐지시점을 기준으로 볼 때, 총 7908 번의 공격이 이루어진 시점 중에 86회 (TCP-SYN 공격의 12회, UDP공격의 26회, ICMP 공격의 48회)가 정상으로 인식되었고, 정상의 경우는 공격 징후 판단 단계를 통과한 1291탐지 횟수 중에 89회가 공격(TCP-SYN 공격 67회, UDP 공격 22회)으로 오탐지되었다. 이와 같이 탐지시점에서 오탐지가 발생하는 경우는 예외 없이 공격이 시작되는 시점과 공격이 끝나는 시점에 MIB값의 갱신이 이루어지는 경우였다.

〈표 7〉 탐지 횟수 및 성능 평가

실험 데이터	TCP-SYN	UDP	ICMP	Normal	Total
공격명령횟수	794	832	802	0	2428
TCP-SYN	794	0	0	0	794
UDP	0	832	0	0	832
ICMP	0	0	802	0	802
탐지횟수	2526	2769	2613	49732	57640
TCP-SYN	2459	0	0	67	2526
UDP	0	2747	0	22	2769
ICMP	0	0	2613	0	2613
Normal	12	26	48	49646	49732

6. 결론 및 향후 과제

본 논문에서는 SNMP MIB의 갱신 주기를 기반으로 SNMP MIB의 상관관계를 기초로 한 계층적 구조의 트래픽 폭주공격 탐지 알고리즘 및 탐지 시간 향상 알고리즘을 제안하였다. 탐지 알고리즘을 적용한 탐지 시스템을 구축하였고, 실험을 통하여 탐지 알고리즘의 타당성을 입증하였다. 또한 제안한탐지 시간 향상 방법은 기존에 SNMP기반의 트래픽 폭주공격 탐지 방법의 탐지 시간 문제를 해결하고 탐지의 정확성을 향상 시켰다. 트래픽 폭주 공격에 대해 빠른 탐지를 통한 대처가 가능해 졌다.

향후 연구로는 SNMP MIB의 갱신 주기가 매우 짧은 경우 본 알고리즘을 적용하면 탐지 트래픽과 시스템 부하가 증가한다. 따라서 탐지 시간과 시스템 부하 및 탐지 트래픽과의 관계를 파악하고 적절한 탐지 시점을 찾는 연구가 더 필요하다. 그리고 제안한 알고리즘을 확대하여 실제 공격을 수행하는 Agent를 탐지하기 위한 방법에 대한 연구를 계획하고 있다.

참 고 문 헌

[1] Myung-Sup Kim, Hun-Jeong Kang, Seong-Cheol Hong, Seung-Hwa Chung, and James W. Hong, “A Flow-based Method for Abnormal Network Traffic Detection,” Proc. of NOMS 2004, Seoul, Korea, Apr. 19-23, pp.559-612, 2004.

[2] E. Duarte, Jr., A. L. dos Santos, “Network Fault Management Based on SNMP Agent Groups,” Proc. of ICDCSW 2001, pp.51.

[3] IETF RFC 1213 “Management Information Base for Network Management of TCP/Ip-Based Internets: MIB-II,” http://www.rfc-editor.org/rfc/rfc1213.txt

[4] Dae-Sung Yoo, Chang-Suk Oh, “Traffic Gathering and Analysis Algorithm for Attack Detection,” KoCon 2004 Spring Integrated conference, Vol.4, 2004, pp.33-43.

[5] “Distributed Denial of Service (DDoS) Attacks/tools”, http://staff.washington.edu/dittrich/misc/ddos/

[6] Jun Li, Constantine Manikopoulos, “Early Statistical Anomaly Intrusion Detection of DoS Attacks Using MIB Traffic Parameters,” Proc. of the IEEE WIA 2003, West Point, NY, Jun. 2003, pp53-59.

[7] GasparyL.P, Sanchez.R.N, Antunes.D.W, Meneghetti.E “A SNMP-based platform for distributed stateful intrusion detection in enterprise network,” IEEE Journal on Selected Areas in Communications, Oct. 2005, Vol.23, pp.1973-1982.

[8] Cabrera. J.B.D. Lewis.L, Xinzhou. Qin, Wenke.Lee, Prasanth.R.K. Ravichandran.B, Mehra.R.K, “Proactive detection of distributed denial of service attacks using MIB traffic variables-a feasibility study” Integrated Network Management Proceedings IEEE/IFIP International Symposium 2001, pp606-622

[9] Qiang Xue, Lin-Lin Guo, Ji-Zhou Sun “The design of a distributed network intrusion detection system IA-NIDS,” Proc. of the International Conference on Machine Learning and Cybernetics 2003, Vol.4, Nov. 2-5, 2003, pp.2305-2308.

[10] Patric carlsson, Markus Fiedler, Kurt Tutschku, Stefan Chevul and Arne A. Nilsson “Obtaining Reliable Bit Rate measurements in SNMP-Managed Networks,” ITC Specialist Seminar, Würzburg, pp.114-123, 2002.

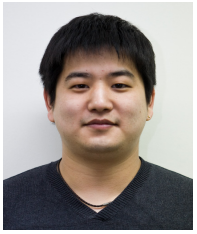
박 준 상

e-mail : runtoyou@korea.ac.kr

2008년 고려대학교 컴퓨터정보학과(학사)

2008년~현 재 고려대학교 컴퓨터정보학과 석사과정

관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석



김 성 윤

e-mail : adayslife@korea.ac.kr

2008년 고려대학교 컴퓨터정보학과(학사)

2008년~현 재 고려대학교 컴퓨터정보학과 석사과정

관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

박 대 희

e-mail : dhpark@korea.ac.kr

1982년 고려대학교 수학과(학사)

1984년 고려대학교 수학과(석사)

1989년 플로리다 주립대학 전산학과(석사)

1992년 플로리다 주립대학 전산학과(박사)

1993년~현 재 고려대학교 컴퓨터정보학과 교수

관심분야 : 멀티미디어마이닝, 기계학습, 인공지능, 지능 데이터베이스, 침입탐지

최 미 정

e-mail : mjchoi@kangwon.ac.kr

1998년 이화여자대학교 컴퓨터공학과 (학사)

2000년 포항공과대학교 컴퓨터공학과 (석사)

2004년 포항공과대학교 컴퓨터공학(박사)

2004년 3월~2004년 9월 포항공대 컴퓨터공학과 박사후 연구원

2004년 10월~2005년 9월 프랑스 INRIA 연구소 박사후 연구원

2005년 11월~2006년 10월 캐나다 워터루대학 컴퓨터과학부 박사후 연구원

2006년 11월~2008년 8월 포항공대 컴퓨터공학과 연구조교수

2008년 8월~현 재 강원대학교 컴퓨터과학과 조교수

관심분야 : 네트워크 및 서비스 관리, XML 및 웹서비스 기반의 네트워크 관리, 미래 인터넷, 자율 관리

김 명 섭

e-mail : tmskim@korea.ac.kr

1998년 포항공과대학교 전자계산학과(학사)

1998년~2000년 포항공과대학교 컴퓨터공학과 (석사)

2000년~2004년 포항공과대학교 컴퓨터공학과 (박사)

2004년~2006년 Post-Doc., Dept. of ECE, Univ. of Toronto, Canada.

2006년~현 재 고려대학교 컴퓨터정보학과 조교수

관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크