SNMP MIB 기반 트래픽 폭주공격 탐지

*박준상, 박대희, 김명섭
*고려대학교 컴퓨터정보학과
e-mail : {*runtoyou, dhpark, tmskim}@korea.ac.kr

Traffic Flooding Attack Detection using SNMP MIB

*Jun-Sang Park, Daihee Park, and Myung-Sup Kim
*Dept. of Computer and Information Science, Korea Univ

요 약

DoS/DDoS 공격과 웜 공격으로 대표되는 트래픽 폭주 공격은 그 특성상 사전 차단이 어렵기 때문에 빠르고 정확한 탐지는 공격 탐지 시스템이 갖추어야 할 필수요건이다. 기존의 SNMP MIB 기반 트래픽 폭주공격 탐지 방법은 1 분 이상의 탐지 시간을 요구하였다. 본 논문은 SNMP MIB 객체의 상관 관계를 이용한 빠른 트래픽 폭주 공격 탐지 알고리즘을 제안한다. 또한 빠른 탐지 시간으로 발생되는 시스템의 부하와 탐지 트래픽을 최소화하는 방안도 함께 제시한다. 공격 탐지 방법은 3 단계로 구성되는데, 1 단계에서는 MIB 정보의 갱신주기를 바탕으로 탐지 시점을 결정하고, 2 단계에서는 MIB 정보간의 상관 관계를 이용하여 공격의 징후를 판단하고, 3 단계에서는 프로토콜 별 상세 분석을 통하여 공격 탐지뿐만 아니라 공격 유형까지 판단한다. 따라서 빠르고 정확하게 공격을 탐지할 수 있고, 공격 유형을 분류해 낼 수 있어 신속한 대처가 가능해 질 수 있다.

Keywords: Traffic Flooding Attack, SNMP, MIB, Detection Algorithm, Detection Time

1. 서론

네트워크 기반 서비스에 대한 의존도가 증가하면서 유해 트래픽 탐지 및 차단은 안전한 서비스 제공을 위해 필수불가결한 요건이 되었다. 대표적인 유해 트래픽인 트래픽 폭주 공격이 발생하면 순식간에 컴퓨터 시스템은 물론 네트워크까지 마비시켜 막대한 피해를 주게 된다. 따라서 트래픽 폭주 공격 탐지에 대한 연구의 중요성이 날로 커지고 있다.

일반적인 공격탐지 방법인 패킷 수집을 통한 트래픽 폭주공격 탐지[1]는 상세한 분석이 가능하지만 고가의 고성능 분석시스템이 요구되고 설치 및 운영 확장성이 부족한 단점을 가지고 있다. 이를 보완하기 위한 방법으로 SNMP 를 이용한 탐지 방법[2,3,6,8]이 효과적으로 사용될 수 있다. SNMP MIB 정보를 이용한 공격탐지는 MIB 데이터 수집을 위한 시스템 및 네트워크 리소스의 사용이 적고, 계층과 프로토콜을 기준으로 표준화된 네트워크 성능 데이터를 제공 받을 수 있기 때문에 효과적인 탐지가 가능하다.[6,7]

본 논문에서는 SNMP MIB 객체의 상관관계 분석을 통한 새로운 트래픽 폭주 공격 탐지 방법을 제시한다. 공격 탐지 방법은 3 단계로 구성되는데, 1 단계에서는 MIB 정보의 갱신주기를 바탕으로 탐지 시점을 결정하고, 2 단계에서는 MIB 정보간의 상관관계를 이용하여 공격의 징후를 판단하고, 3 단계에는 프로토콜 별 상세 분석을 통하여 공격 탐지뿐만 아니라 공격 유형까지 판단한다.

본 논문은 다음과 같은 순서로 구성된다. 2 장에서는 트래픽 폭주 공격에 대한 기존 연구와 문제점을 기술하고, 3 장에서는 SNMP MIB 상관관계를 기초로 한 트래픽 폭주공격 탐지 방법을 제시한다. 4 장에서는 제안된 방법의 타당성을 실험으로 증명하고, 5 장에서는 결론과 향후 연구에 대하여 기술한다.

2. 관련 연구

SNMP MIB 을 기반으로 트래픽 폭주 공격을 탐지하는 방법은 3 가지 요소가 고려되어야 한다.

Ⅰ. 다양한 공격을 반영하는 MIB 의 선정
Ⅱ. 선정된 MIB 의 활용 방법
Ⅲ. 탐지 시점의 결정

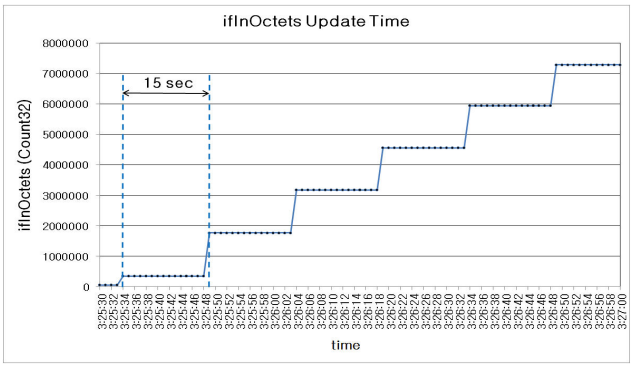
SNMP MIB 기반의 트래픽 폭주 공격 탐지 방법에서 가장 우선적으로 고려되어야 하는 부분은 공격을 탐지하기 위해 사용되는 MIB 의 선정이다. 선정된 MIB 은 다양한 유형의 공격 특성을 반영해야 한다. 기존의 탐지 방법[3]에서는 *tcpInErrs, udpNoPorts, icmpOutEchoReps* MIB 객체들을 사용하였지만, 각각의 공격을 실시 하였을 때 해당 객체가 반응하지 않았다. 이는 공격 도구가 견고해 짐에 따

* 이 논문은 2007 년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임.(KRF-2007-331-D00387)

라 사전에 Port Scanning 작업이 이루어 지고, 오류가 없는 패킷을 전송하기 때문이다. MIB 의 선정은 탐지 알고리즘의 탐지율을 좌우하는 중요한 요소이기 때문에 정확한 선정이 요구된다.

두 번째로 고려되는 부분은 선정된 MIB 의 활용방법이다. 기존의 탐지 방법에서 선정된 MIB 은 트래픽 추이를 통한 분석[3], 통계적 분석[8], 학습 알고리즘을 통한 분석[9] 등 다양한 방법으로 활용되고 있다. 이러한 접근 방법은 과거의 트래픽현황을 바탕으로 트래픽 폭주 공격을 탐지하게 된다. 하지만 유동적이고 변화가 많은 네트워크 트래픽의 특성상 위험 요소가 뒤따른다. 또한 관리 대상시스템의 사용 환경의 변화에 따라 탐지 알고리즘의 신뢰도가 떨어진다. 기준 트래픽을 재설정 하여 신뢰도를 증가 시킬 수 있지만 기준 트래픽을 설정하기 위해 많은 시간이 소비되는 문제점이 있다.

셋째, 트래픽 폭주 공격이 분산화되고, 대형화 됨에 따라 타깃 빠른 탐지가 우선되어야 한다. 따라서 탐지 시점의 결정은 탐지시스템을 평가하는 중요한 요소로 작용한다.



(그림 1) ifInOctets 의 갱신 주기

그림 1 은 공격 발생 시 interface 그룹의 *ifInOctets* 값을 매 1 초단위로 보여 주고 있다. 그림 1 을 통해 SNMP MIB 값은 특정한 주기를 기준으로 갱신됨을 알 수 있다. 기존의 탐지시스템의 탐지 시점은 최소 1 분의 특정 주기를 기준으로 수행되기 때문에 MIB 의 갱신 주기가 반영되지 않아 정확한 MIB 트래픽 수집이 어렵다. MIB 의 갱신 주기를 기준으로 탐지 시스템이 동작한다면 탐지의 정확성과 탐지 시간을 향상시킬 것으로 기대된다. 그러나 짧은 탐지 주기는 탐지를 위한 소비 트래픽과 시스템 부하를 증가 시키기 때문에 이를 고려한 효율적인 탐지 시점 결정 알고리즘이 필요하다.

3. 탐지 시간 향상 및 탐지 알고리즘

본 장에서는 SNMP MIB 객체의 상관관계를 통한 한 트래픽 폭주 공격을 탐지하는 알고리즘과 탐지시간을 향상 시키는 알고리즘에 대해 설명한다.

표 1 은 본 논문에서 사용된 MIB 객체들이다. 탐지 알고리즘에 사용된 MIB 객체들은 모든 SNMP agent 에서 공통으로 제공되는 RFC1213[4]에서 정의된 mib-2 그룹의 MIB 객체들로 구성되었다.

<표 1> 탐지 시스템에 사용된 MIB

MIB-2 Group	SNMP MIB objects
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*

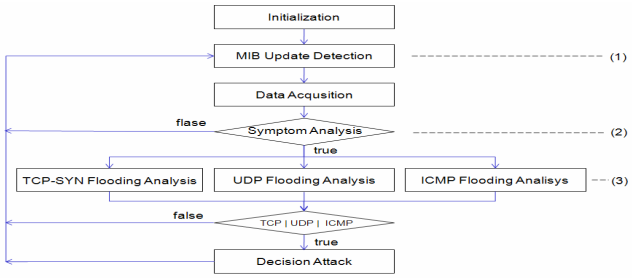
MIB 객체의 선택은 각 그룹의 MIB 객체의 의미와 트래픽 폭주공격 패킷과의 상관관계 및 실제 공격에 반응하는 MIB 객체들의 전수조사를 통하여 선정되었다.

표 2 는 본 논문에서 제안하는 탐지 알고리즘에 사용되는 기호 및 수식들을 정리한 것이다.

<표 2> 탐지 알고리즘에 사용되는 기호 및 수식

t	1 초 단위의 시간
t_n	n 번째 탐지 알고리즘 적용 시간
Mib(t,oid)	시간 t 에 수집한 SNMP oid 객체의 값
Diff(t_n, oid)	= Mib(t_n, oid) – Mib(t_{n-1}, oid)
bps(t_n)	= Diff(t_n, ifInOctets) / t_n - t_{n-1} * 8
pps(t_n)	= Diff(t_n, ifInUcastPkts) / t_n - t_{n-1}
DeliverRatio(t_n)	= Diff(t_n, ipInDelivers) / Diff(t_n, ipInReceives)
ResponseRatio(t_n)	= Diff(tn,ipOutRequests)/Diff(t_n, ipInReceves)

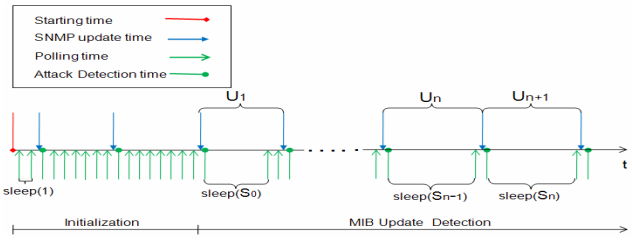
그림 2 는 SNMP MIB 기반 트래픽 폭주 공격 탐지 알고리즘의 흐름도이다. 크게 3 단계로 나누어지는데 MIB 의 갱신 주기 예측을 통한 탐지 시점 결정 단계 (1)와 공격 가능성 판단 단계(2)와 세부 공격 탐지 단계(3)로 나누어진다.



(그림 2) 트래픽 폭주 공격 탐지 흐름도

3.1 탐지 시점 결정 단계

SNMP MIB 의 갱신 시점을 기준으로 탐지 시점을 결정함으로써 탐지 시간을 향상시킨다.



(그림 3) 공격 탐지 시점 결정 알고리즘

그림 3 은 공격 탐지 시점 결정 알고리즘을 도

식화한 것이다. Initialization 단계에서 매 1 초 주기 Polling 을 통해서 SNMP MIB 값의 갱신 시점을 찾고, 갱신 시점에서 탐지 시스템을 가동한다. 이 단계에서 찾아낸 갱신 주기의 최소값(U_{min})과 마지막으로 탐지시스템을 적용한 시간(t_{old})을 초기 값으로 하여 탐지 시스템을 운영한다

MIB 갱신주기 탐지 단계에서는 MIB 값의 다음 갱신 시점을 예측하여 그 시간(S)만큼 시스템을 sleep 시킴으로써 시스템의 부하와 트래픽의 사용량을 줄인다. 예측을 위하여 현재까지 측정된 갱신주기들에 exponential average(A)로 적용함으로써 sleep 시간(S)을 아래의 수식과 같이 결정한다. 이때 α 값으로 0.5 를 사용한다

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

$$S_n = A_n - 1$$

3.2 공격 가능성 판단

트래픽 폭주 공격은 대량의 패킷을 전송함에 기초한다. 따라서 공격이 발생하면 BPS(Bits per Second)나 PPS(Packets per Second)값이 일정 수준 이상을 유지된다. 이러한 사실을 바탕으로 탐지 알고리즘의 수행 여부를 결정하고 ip 그룹의 MIB 간의 상관 관계를 나타내는 DeliverRatio(), ResponseRatio(), diff()값을 통해 공격의 가능성을 판단한다. 이 단계를 통해 탐지 시스템의 시스템 부하를 효율적으로 감소시킬 수 있다.

```
Boolean Symptom_analysis( … ){
    int weight = 0;
    if(bps < Th(bps) && pps < Th(pps)) return FALSE;
    if( DeliverRatio(t) < Th(DeliverRatio) )   weight++;
    if( ResponseRatio(t)<Th(ResponseRatio) ) weight++;
    if(Diff(t,ipOutDiscards) > Th(ipOutDiscards) ) weight++;
    if(weight > =1) return TRUE;
    return FALSE;
}
```

상위 계층 전달률(DeliverRatio)은 인테페이스를 통해 받은 패킷이 상위 계층에 전달되는 정도이다. 정상적인 트래픽의 경우 0.8 이상의 높은 전달률을 보이지만 폭주 공격이 발생하면 버퍼 공간의 부족과 오류로 인해 0.3 이하의 전달률을 보인다.

응답률(ResponseRatio)은 상위 계층으로 전달된 데이터그램의 개수 즉 *ipInDelivers* 대한 응답률이다. 정상적인 트래픽의 경우 0.5 이상의 응답률을 보이지만 공격이 발생하면 0.4 이하의 응답률을 보인다.

폐기된 패킷 개수를 나타내는 *ipOutDiscards* 객체는 인터페이스를 통해 받은 패킷을 버퍼공간의 부족으로 인해 버려지는 경우에 증가한다.

공격의 가능성 판단 단계에서 사용되는 임계치(Threshold)는 표 3 과 같이 결정된다. BPS, PPS 에 대한 임계치(Threshold)는 공격 트래픽의 최소 BPS, PPS 를 포함하며, 전달율과 응답율의 신뢰성을 보장하는 수준에서 결정되어진다. DeliverRatio, ResposeRatio, diff() 에 대한 임계치(Threshold)는 위의 설명에 따라 설정되었다.

<표 3> 공격 가능성 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 객체들이 반응을 보이기 때문에 탐지와 분류가 가능해진다. 실험을 통해 각 공격에 반응하는 SNMP MIB 객체를 전수 조사하여 표 1 과 같이 선별하였고, 이들 MIB 객체의 관계와 임계치(Threshold)이상의 변화 유무를 통해 공격을 탐지하고 공격 유형을 밝혀낸다. 이때 임계치(Threshold)의 설정은 정상 트래픽의 오탐지를 방지하기 위함이다. 각각의 공격 유형별 탐지 알고리즘은 아래와 같다.

3.3.1 TCP-SYN Flooding 공격 탐지

tcpAttamptFail 은 목적지 포트가 닫혀있는 경우와 RST 패킷을 받은 경우 연결이 실패되어 증가된다. *tcpOutRsts* 객체는 SYN 패킷을 받은 타깃 호스트의 해당 포트가 닫혀있는 경우 목적지 포트에 RST 패킷을 전송하기 때문에 증가된다.

```
Boolean TCP-SYN_Flooding_analysis( … )
{
    if( Diff(t, tcpAttamptFail) > Th(tcpAttamptFail)   ||
      Diff(t, tcpOutRsts )> Th(tcpOutRsts) ) )
        return TRUE;
    return FALSE;
}
```

3.3.2 UDP Flooding 공격 탐지

패킷에 대한 오류와 버퍼 공간의 부족으로 *udpInErrs* 객체가 증가된다. 또한 UDP Flooding 공격이 발생하면 상위 프로토콜에 전달되지 않거나 포트가 닫혀있는 등의 오류가 발생하면 오류보고 메시지를 송수신하기 때문에 *icmpOutDestUnreachs* 객체를 증가 시킨다.

```
Boolean UDP_Flooding_analysis( … )
{
    int weight = 0;
    if( Diff(t, udpInErrs ) > Th(udpInErrs)) weight ++;
     if(Diff(t, icmpOutDstUnreachs)==diff(t,icmpOutMsgs) &&
       Diff(t,icmpOutDstUnreachs) > 0 &&
       Diff(t, icmpOutMsgs) > 0 &&
       Diff(t, icmpInDstUnreachs) == 0) weight++;
    if(weight > =1) return TRUE;
    return FALSE;
}
```

3.3.3 ICMP Flooding 공격 탐지

IMCP Echo Request 에 의해 *icmpInEchos* 와 함께 *icmpInMsgs* 가 급격히 증가한다. ICMP Echo request 를 IP 계층에서는 모두 수용하기 때문에 *ipInDiscards* 는 증가하지 않지만 버퍼 공간의 부족으로 수신된 모든 패킷에 대해 응답이 이루어지지 않고 폐기되는 패킷에 의해 *ipOutDicards* 값을 증가시킨다. 또한 상위 프로토콜에 전달되지 않거나 포트가 닫혀있는 등의 오류가 발생하면 오류보고 메시지를 송수신하기 때문에 *icmpInDestUnreachs*, *icmpOutDestUnreachs* 객체를 증가 시킨다.

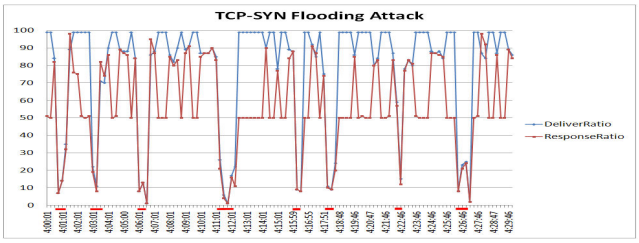
```
Boolean ICMP_Flooding_analysis( … )
{
    int weight=0;
    if( Diff(t, icmpInMsgs)   > Th(icmpInMsgs)   ||
      Diff(t, icmpinEchos) > Th(icmpInEchos) ) weight++;
    if( Diff(t, icmpInDstUnreachs)   > 0 &&
      Diff(t, icmpOutDstUnreachs) > 0 &&
      Diff(t, icmpOutEchoReps) > 0   &&
      Diff(t, ipOutDiscards > 0) ) weight++;

    if(weight > =1) return TRUE;
    return FALSE;
}
```

4. 실험 및 결과

본 논문에서 제안한 알고리즘의 평가를 위하여 공격 호스트 3 대, 타깃 호스트 1 대, 탐지 시스템 1 로 구성하였다. 실험은 TCP-SYN, UDP, ICMP Flooding 공격을 위해 Stacheldraht[5]를 이용하였다. 실험을 위하여 MIB 갱신 주기의 평균이 15 초인 시스템을 타깃 시스템으로 하여 10 일 동안 무작위로 TCP-SYN, UDP, ICMP Flooding 공격을 실시하였다. 실험은 특정시간에 하나의 공격만을 실시하였고, 공격 시점과 공격 지속시간은 무작위로 설정하였다.

지면상의 이유로 TCP-SYN Flooding 공격을 중심으로 알고리즘의 내용을 중점 기술하고, 성능 평가 내용을 기술한다.



(그림 4) DeliverRatio, ResponseRatio 변화

그림 4 는 TCP-SYN Flooding 공격 시 DeliverRatio 값과 ResponseRatio 값의 변화이다. 공격이 탐지된 시간에 DeliverRatio, ResponseRatio 값이 각각 0.8, 0.4 미만의 값을 가짐을 확인할 수 있다. ipOutDiscards 값은 증가를 보였지만 모든 공격 시간을 포함하지 못하기 때문에 TCP-SYN Flooding 공격이 발생하면 DeliverRatio 와 ResponseRatio 에 의해서 공격의 징후로 판단한다.

세부 분석 단계에서는 tcpAttamptFail, tcpOutRsts 값에 의해 TCP-SYN 공격으로 탐지하고 분류한다.

제안된 방법은 평균 8.23 초의 탐지시간을 보인다. 이는 관리 대상 시스템의 안전성을 보장하기에 충분한 시간이다. 또한 시스템 부하에 거의 영향을 미치지 않음을 확인할 수 있었고, 갱신 주기 및 공격 탐지를 위한 트래픽 부하 역시 거의 없음을 알 수 있었다. 이는 탐지 시간 향상 알고리즘에서 탐지 시점 결정을 위해 Exponential Average 를 적용한 결과이다. 또한 공격 징후 판단 단계에서 정상트래픽의 97.51%는 정상으로 분류할 수 있어 시스템의 부하를 줄일 수 있다. 이러한 결과는 제안된 방법론을 이용하여 다량의 시스템의 트래픽 폭주공격 탐지를 수행할 수 있어 확장성이 뛰어남을 나타낸다.

<표 4> 공격 탐지 성능 평가

실험 데이터	TCP-SYN	UDP	ICMP	Normal	Total
공격명령횟수	794	832	802		2428
TCP-SYN	100%	0%	0%		794
UDP	0%	100%	0%		832
ICMP	0%	0%	100%		802
탐지횟수	2526	2769	2613	49732	57640
TCP-SYN	97.34%	0%	0%	2.66%	2526
UDP	0%	99.2%	0%	0.8%	2769
ICMP	0%	0%	100%	0%	2613
Normal	0.03%	0.06%	0.9%	99.82%	49732

실험을 통하여 제시된 알고리즘은 모든 공격에 대하여 1~2 회의 탐지시도에서 100% 정확하게 탐지할 수 있었다. 탐지 시점을 기준으로 볼 때 표 4 와 같이 TCP-SYN, UDP 의 경우 약간의 FP, FN 을 나타냈다. 이는 공격 발생시점과 탐지시점의 차이가 1 초 이내인 경우에 생기는 것으로 무시해도 가능한 수치라고 판단된다.

5. 결론

본 논문에서는 SNMP MIB 의 갱신 주기를 기반으로 SNMP MIB 의 상관관계를 기초로 한 트래픽 폭주공격 탐지 알고리즘 및 탐지 시간 향상 알고리즘을 제안하였고, 실험을 통하여 탐지 시간 향상과 탐지 알고리즘의 타당성을 입증하였다. 본 논문에서 제시된 탐지 시간 향상 방법은 기존에 SNMP 기반의 트래픽 폭주공격 탐지 방법의 탐지 시간 문제를 해결하여 공격에 대해 빠른 대처가 가능해 졌다. 또한 탐지 시간 향상 과정에서 발생하는 트래픽과 시스템 부하 문제를 해결하는 효과적인 알고리즘이다.

향후 연구로는 SNMP MIB 의 갱신 주기가 매우 짧은 경우 탐지 시간과 시스템 부하 및 탐지 트래픽과의 관계를 파악하고 적절한 탐지 시간을 찾는 연구가 더 필요하다. 제안한 알고리즘을 확대하여 네트워크단에서 트래픽 폭주공격 탐지 방법에 대한 연구를 계획하고 있다.

참고 문헌

[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.

[3] 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.

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

[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, 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”, Machine Learning and Cybernetics, 2003 International Conference, Vol.4, 2-5 Nov. 2003, pp. 2305- 2308.