

IPv6 환경에서 지수 평활법을 이용한 공격 탐지 알고리즘

Attack Detection Algorithm Using Exponential Smoothing Method on the IPv6 Environment

구항욱*, 오창석**

충북대학교 컴퓨터공학과*, 충북대학교 전기전자컴퓨터공학부**

Hyang-Ohk Koo(ok999@hanmail.net)*, Chang-Suk Oh(csoh@nwork.chungbuk.ac.kr)**

요약

DDoS(Distributed Denial of Service) 공격에 사용되는 네트워크 트래픽과 정상적인 서비스를 위한 네트워크 트래픽을 구분해 내는 것은 쉽지 않다. 정상적인 패킷을 유해 트래픽으로 판단하고 유해 트래픽의 공격자의 의도대로 서비스를 못하는 경우가 발생하므로, DDoS 공격으로부터 시스템을 보호하기 위해서는 공격 트래픽에 대한 정확한 분석과 탐지가 우선되어야 한다. IPv6 환경으로 전환될 때 발생하는 유해 트래픽에 대한 연구가 미약한 상태이므로, 본 논문에서는 IPv6 환경에서 NETWOX로 공격을 수행하고 공격 트래픽을 모니터링한 후 MIB(Management Information Base)객체를 지수 평활법을 적용하여 예측치를 구한 후 임계치를 산정하여 공격을 판별하는 방법을 제안한다.

■ **중심어** : | 트래픽 플러딩 공격 | 지수 평활법 | NET-SNMP |

Abstract

Mistaking normal packets for harmful traffic may not offer service in conformity with the intention of attacker with harmful traffic, because it is not easy to classify network traffic for normal service and it for DDoS(Distributed Denial of Service) attack. And in the IPv6 environment these researches on harmful traffic are weak. In this dissertation, hosts in the IPv6 environment are attacked by NETWOX and their attack traffic is monitored, then the statistical information of the traffic is obtained from MIB(Management Information Base) objects used in the IPv6. By adapting the ESM(Exponential Smoothing Method) to this information, a normal traffic boundary, i.e., a threshold is determined. Input traffic over the threshold is thought of as attack traffic.

■ **keyword** : | Traffic Flooding Attack | Exponential Smoothing Method | NET-SNMP |

I. 서론

정보통신의 발달과 인터넷의 급속한 발전으로 인하

여 우리나라는 인터넷 이용인구와 초고속인터넷 가입자가 빠르게 증가하고 있으며 인터넷 이용이 보편화되고 서비스의 고도화에 대한 수요로 IPv4는 한계에 도

* 이 논문은 2005년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었습니다.

접수번호 : #051116-002

접수일자 : 2005년 11월 16일

심사완료일 : 2005년 12월 02일

교신저자 : 오창석, e-mail : csoh@nwork.chungbuk.ac.kr

로된 서비스를 할 수 없게 만드는 과정이다.

```

root@b2b:~# root@b2b:~# ./udp-flood.sh
IP
version: traffic class :          flow label
 6      0                  0
payload length : next header : hop limit
0x0008-8      0x11-1?      50
source
3fff:2209:1:2::1
destination
3fff:2209:1:2::2
UDP
source port      destination port
0x0402-1234      0x7467-31335
length          checksum
0x0008-8         0x8C8B-46267
    
```

그림 2. UDP 플러딩 공격

2.3 ICMP 플러딩 공격

ICMP 메시지는 다양한 공격의 표적이 될 수 있다. ICMP 메시지는 수신자로 하여금 메시지가 메시지를 처음 발송한 곳이 아닌 다른 곳에서 온 것이라 믿도록 하는 공격을 받을 수 있다. ICMP 메시지는 메시지 및 그에 대한 응답이 발신자의 의도와는 다른 곳으로 수송되도록 하는 공격을 받을 수 있다. 악의를 가지고 중간에 가로채어 메시지를 포함하고 있는 IP 패킷에서 목적 및 발신 주소를 변경하는 경우와 메시지 필드 및 payload 변경 공격, 계속해서 잘못된 IP 패킷을 수송하는 방식으로 서비스 거부 공격에 사용된다. [그림 3]은 ICMP 플러딩 패킷은 type이 ICMP echo request를 나타내도록 128로 설정하였다. 수신측에 계속적인 ICMP request를 보냄으로써 제대로 된 서비스를 할 수 없게 만드는 과정이다.

```

root@b2b:~# root@b2b:~# ./icmp-flood.sh
IP
version: traffic class :          flow label
 6      0                  0
payload length : next header : hop limit
0x0008-8      0x30-58      50
source
3fff:2209:1:2::1
destination
3fff:2209:1:2::2
ICMP_echo request
type      code      checksum
0x80-128  0x00-0      0x4971-18881
id        sequence
0x8245-33349      0xFFEC-61420
data:
    
```

그림 3. ICMP 플러딩 공격

2.4 기존의 공격 탐지

IPv6 환경 공격 탐지 기법으로 Libpcap을 이용한 탐지 모듈이 현재 다수 사용되고 있으며 탐지 모듈은 공격 모듈에서 공격을 수행하였을 경우 네트워크에서 트래픽을 수집한 결과를 이용하여 공격 여부를 탐지하는 알고리즘이다. 탐지 모듈의 전체 흐름도는 [그림 4]이며, [그림 5]는 Libpcap 이용한 트래픽 검출 알고리즘이다. 탐지 모듈은 네트워크 기반이므로 네트워크상의 모든 트래픽을 수집하게 된다. 수집된 트래픽들은 헤더 정보에서 분석을 위해 수집 시간, 프로토콜, 출발지 주소, 목적지 주소를 추출하여 탐지 알고리즘에 의해 공격 여부를 판단하게 된다[3].

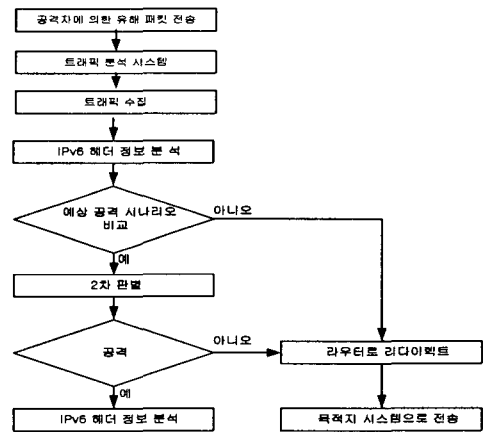


그림 4. Libpcap 이용한 트래픽 분석 흐름도

Libpcap을 이용한 방법은 공격 트래픽을 누적인 평균임계값을 구하여 그 수치를 기준으로 공격을 탐지하는 단순한 구조로 TCP, UDP, ICMP 플러딩을 구분하고 있다. 공격탐지 시간은 1분이 소요되며 Libpcap을 탐지 모듈 알고리즘은 먼저 Packetcapture 함수를 이용하여 네트워크 기반의 패킷을 수집한 후 ipv6save 구조체에 필요한 정보를 저장하게 된다. 다음으로는 Detectmodule 함수를 이용하여 공격 데이터베이스에 있는 공격 종류와 일치하게 되면 해당 차단 모듈을 호출하고 로그파일을 남기게 된다. 정상일 경우는 서비스를 내부 네트워크로 송수신이 허용되게 된다.

```

DAD-NA Attack detection:
if [ IPv6save->nh=58 && IPv6save->data[0]=135 ]
Increase DAD_NS_Count
if [ IPv6save->nh=58 && IPv6save->data[0]=136 ]
Increase DAD_NA_Count

TOP SYN Attack detection:
if [ IPv6save->nh=6 && (IPv6save->data[6])&0x02=2 ]
Increase TOP_SYN_Count
TOP_SYN_rate=TOP_SYN_Count/Total_Packet_Count-
(number of different attack)+1
if [TOP_SYN_rate>threshold B]
if [top_SYN-rate>threshold A]
Detect Attack level 2
else
Detect Attack level 1
DAD_NA_rate=DAD_NA_Count/Total_Packet_Count-
(number of different attack)+1
if [DAD_NA_rate > threshold A && DAD_NA_Count>DAD_NS_count]
{
if [DAD_NA_Count > threshold B=30]
Detect Attack level 2
else
Detect Attack level 1
}
UDP Flooding Attack detection:
if [ IPv6save->nh=17 &&
IPv6save->data[2]*256+IPv6save->data[3]=attack port ]
increase UDP_Flooding_Count
UDP_Flooding_rate=UDP_Flooding_Count/Total_Packet_Count
-(number of different attack)+1
if [UDP_Flooding_rate > threshold]
Detect Attack level 2
ICMP Flooding Attack detection:
if [ IPv6save->nh=58 && IPv6save->data[0]=128
&& IPv6save->da[0]=0xff ]
increase ICMP_Ping_Count
ICMP_Ping_rate=ICMP_Ping_Count/Total_Packet_Count -
(number of different attack)+1
if [ICMP_Ping_rate > threshold]
Detect Attack level 2
    
```

그림 5. Libpcap 이용한 트래픽 검출 알고리즘

III. 지수 평활법을 이용한 공격 탐지

3.1 트래픽 분석 모델

[그림 6]은 트래픽 분석모델을 관리대상시스템(피해 시스템)의 트래픽을 분석하는 모델로 수행 가능 개념도이다. 관리대상시스템에 공격이 유입되면 관리시스템에서 MIB을 30초 단위로 수집하여 공격 탐지 모듈에서 탐지를 하고 공격으로 판정이나면 관리자에게 경보를 발송하고 MRTG(MultiRouter Traffic Grapher)를 통해 트래픽의 통계 그래프를 TCP, UDP, ICMP별로 확인할 수 있다[4-6].

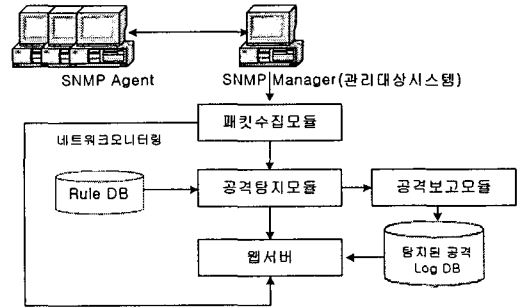


그림 6. 트래픽 분석 모델

MIB 객체 수집시간을 단축하기 위해 TCP, UDP, ICMP 트래픽 수치를 구할 수 있는 MIB와 ICMP 메시지의 수치를 정보 MIB을 사용하여 공격 판정에 사용하였으며, 포트를 이용한 공격인지 파악하기 위해 수신 Port번호 MIB을 추가로 사용하였다.

표 1. 공격 검출에 사용되는 MIB

MIB 객체	설명
ipv6IfStatsInReceives	수신한 데이터그램의 총수
ipv6TcpConnLocalPort	TCP로 connection 수신된 로컬 Port 번호
ipv6UdpLocalPort	UDP로 수신된 로컬 Port 번호
ipv6IfIcmpOutEchoReplies	송신된 ICMP 요청 메시지의 총 개수
tcpInSegs	수신된 세그먼트의 총 개수
udpInDatagrams	UDP 데이터그램의 총 개수
ipv6IfIcmpInNeighborSolicits	NeighborSolicit 수신 메시지수
ipv6IfIcmpInNeighborAdvertisements	Neighbor Advertisement 수신 메시지수
ipv6IfIcmpInRouterSolicits	RouterSolicit 수신 메시지수
ipv6IfIcmpInRouterAdvertisements	Router Advertisement 수신 메시지수
ipv6IfStatsInMcastPkts	송신 한 멀티 캐스트 Packet 수

3.2 공격 탐지 모듈

MIB 객체로 수집된 실제 트래픽을 이용하여 예측치와 평균절대오차를 구하여 정상트래픽 상한을 초과하면 공격으로 판정하는 모듈이다.

- 1) 초기화 : 네트워크에서 트래픽 발생하는 매 시점마다 공격을 판단하는 것은 어려운 일이다. 따라서 본 논문에서는 각 프로토콜별 트래픽의 양을 일정한 주기(30초마다) 초기의 MAD 또는 표준편차를 결정한다.
- 2) 실제치 입력 : 트래픽 측정 장치로부터 t시점에서 실

제로 측정된 트래픽 X_t 를 입력 받는다.

- 3) 예측치 존재 판단 : 예측을 처음 시작하였거나 이상 상태가 끝난 후, 지수평활법이 재실행되었을 경우에는 그 시점에 대한 예측치(Y_t)가 존재하지 않는다. 예측치가 존재하지 않을 경우에는 예측치가 실제치와 같다고 설정한다($Y_t = X_t$). 이 때 예측오차(E_t)는 0이 된다. 예측치가 존재하는 일반적인 경우에는 예측오차는 식 (1)과 같이 실제치와 예측치의 차이가 된다.

$$E_t = X_t - Y_t \quad (1)$$

- 4) 정상 범위 계산 :

$$\begin{aligned} UCL &= Y_{t+1} + z_\alpha \sigma_t \\ &= Y_{t+1} + 1.25 z_\alpha MAD_t \end{aligned} \quad (2)$$

식 (2)를 바탕으로 예측치(Y_t)와 MAD 또는 표준편차를 이용하여 관리한계의 상한을 결정한다.

식 (2)의 z_α 는 관리한계의 범위와 정상트래픽한계 내에 속할 확률을 결정하는 값으로, 표준정규분포표에서 값을 찾아 사용 하였다[8].

- 5) 이상 상태 파악 실제치(X_t)가 정상 트래픽 한계를 벗어날 경우 ($X_t > Y_t + z_\alpha \sigma_t$)에서는 네트워크에 이상이 발생하였다고 판단하여 지수평활법을 정지하고, 이상 상태처리 모듈을 수행한다. 실제치가 정상 범위에 있을 경우 ($X_t \leq Y_t + z_\alpha \sigma_t$)에는 다음 단계로 진행하여 지수평활법을 계속 수행한다.

- 6) 다음 시점 예측, 표준편차, MAD 업데이트 : 관리한계 내의 실제치(X_t)를 사용하여

$$Y_{t+1} = \alpha X_t + (1 - \alpha) Y_t, \quad 0 < \alpha \leq 1 \quad (3)$$

식 (3)에 따라, 다음 시점의 예측치(Y_{t+1})를 계산한다. 또한 현재의 예측 오차를 바탕으로

$$\sigma_t^2 = \delta E_t^2 + (1 - \delta) \sigma_{t-1}^2, \quad 0 < \delta \leq 1 \quad (4)$$

$$MAD_t = \delta |E_t| + (1 - \delta)MAD_{t-1} \quad (5)$$

식 (4)와 식(5)를 이용하여 σ_t 를 사용하여, 다음 시점의 정상 범위계산에 필요한 표준편차 또는 MAD를 업데이트한다. 그 후, 다음 시점의 실제치가 입력될 때까지 대기한다.

- 7) 이상 상태 처리 모듈 : 이상 상태파악단계에서 트래픽의 공격상태가 발견되면 지수평활에 의한 예측을 중지하고, 트래픽이 다시 정상 상태가 되기를 기다린다. 현재의 공격트래픽은 무시하고, 다음 시점의 트래픽을 입력 받는다. 현 시점의 트래픽이 정상적이라고 판단되면, 지수평활법을 다시 시작한다. 다시 시작되는 지수평활법에서, 표준편차 또는 MAD는 이상 발생 이전의 값이 그대로 사용되고, 그 시점에 대한 예측치가 없기 때문에 예측치의 초기값은 실제치와 같다고 설정된다. 입력된 트래픽이 여전히 공격상태로 판단되면 정상상태의 트래픽이 도착할 때까지 대기한다[7][8].

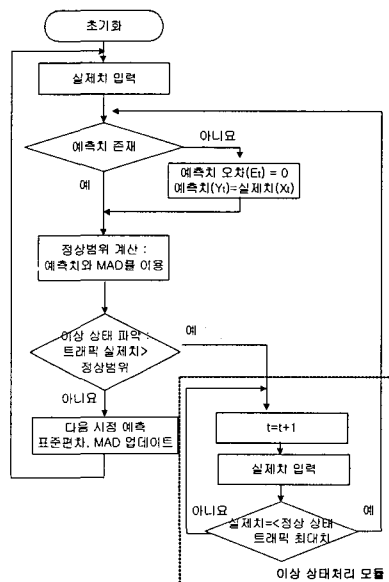


그림 7. 임계치 산출 흐름도

[그림 7]의 임계치 산출 흐름도에서 중요한 점은 이상 트래픽이 한번 발생하면 이상 트래픽의 계속된 유

입은 무시하며 정상상태의 트래픽이 도착할 때까지 대기한다.

공격 탐지 모듈은 [그림 8]과 같이 트래픽이 유입되면 ipv6IfStats InReceives의 MIB으로 임계치를 구하여 공격을 판정하게 된다.

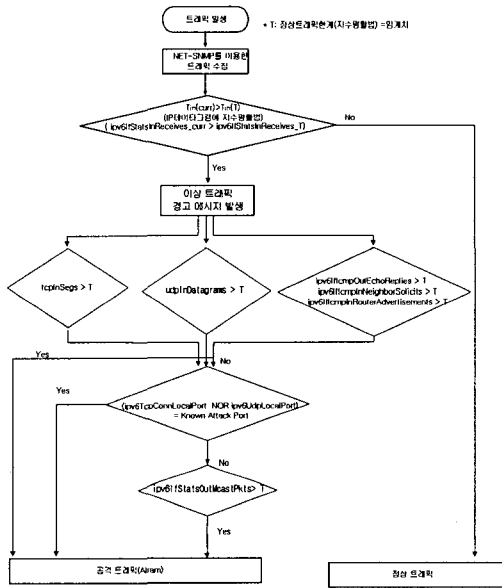


그림 8. 트래픽 분석 흐름도

```

Traffic analysis procedure
let Log() be a reading a log's value
let attack() be a attack traffic alarm

if ipv6IfStatsInReceives(curr) > ipv6IfStatsInReceives(T)
    return attack_IP_Datagram_Flooding : current log value

if tcpInSegs(curr) > tcpInSegs(T)
    return attack_TCP_Flooding
if udpInDatagrams(curr) > udpInDatagrams(T)
    return attack_UDP_Flooding
if ipv6IcmpOutEchoReplies(curr) > ipv6IcmpOutEchoReplies(T)
    return attack_ICMP_Flooding
if ipv6IcmpInNeighborSolicits(curr)>ipv6IcmpInNeighborSolicits(T)
    return attack_ICMP_Flooding
if ipv6IcmpInRouterAdvertisements(curr)>
    ipv6IcmpInRouterAdvertisements(T)
    return attack_ICMP_Flooding

if ipv6IfStatsOutMcastPkts(curr) > ipv6IfStatsOutMcastPkts(T)
    return attack_Multicast_Flooding
if ipv6UdpLocalPort = Known attack port
    return attack_Udp_Port_attack

if ipv6TcpConnLocalPort = Known
    return attack_Tcp_Port_attack
pass traffic analysis function
return Log()
    
```

그림 9. 트래픽 분석 알고리즘

TCP, UDP, ICMP별 지수평활법의 예측치와 임계치를 구하여 실제 수집된 MIB 실제치와 비교하여 임계치 초과일 경우 공격으로 판정한다. 포트별 판정 후 기존의 공격에 사용된 포트번호를 비교하여 TCP, UDP 공격으로 판정하며, 멀티캐스트로 입력되는 트래픽 총 패킷량을 수집하여 다시 한번 임계치를 구하여 마지막 공격을 검출한다. 트래픽분석 알고리즘은 [그림 9]와 같다.

공격트래픽의 포트번호를 TCP 포트와 UDP포트로 접근시 차단을 위해 기존 공격 포트를 리스트에 저장하여 검출한다. 사용되는 MIB은 ipv6TcpConnLocal Port (TCP로 수신된 connection 로컬 Port 번호), ipv6UdpLocalPort(UDP로 수신된 로컬 Port 번호)이다.

IV. 실험 및 결과 고찰

실험은 IPv6 환경에서 수행하였으며 PC 라우터를 통해 마스터에서 공격한 명령이 에이전트로 전달되고 감염된 여러 에이전트가 동시에 관리대상으로 NETWIB으로 만들어진 공격 패킷을 전송하여 관리대상시스템에서 서비스 거부 현상을 발생시킨다 [9][10].

4.1 공격 트래픽 검출

NET-SNMP를 이용한 트래픽 분석을 통해 지수평활계수를 $\alpha=0.3$ (평활상수), $\delta=0.05$ (분산평활상수) $\sigma=7$ (시그마계수)으로 설정하여 정상트래픽 한계를 구하여 공격을 검출한 구간은 실제치 부분이며 임계치를 구하고 지수평활계수를 사용하여 공격을 검출한다. 실험에 사용된 트래픽 기준으로는 일반적인 트래픽의 기준선으로 한달 평균의 일반 트래픽 수치를 표현하여 공격트래픽과 대조에 사용하였다. 실제 트래픽이 유입되면 예측치를 구하여 임계치가 산정된다. 임계치를 기준으로 정상트래픽 한계를 초과한 부분이 공격으로 판정되는 구간이다. TCP 플러딩 공격이 21:01:00~21:01:30에서 TCP 트래픽이 폭주하여 21:01:30에서 정상트래픽한계를 벗어나 공격이 검출된 공격트래픽

이 시작된다.

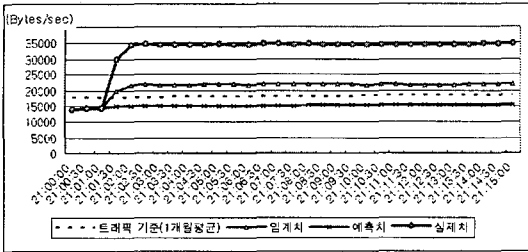


그림 10. TCP 플러딩 공격 트래픽($\alpha = 0.3, \delta = 0.05, \sigma = 7$)

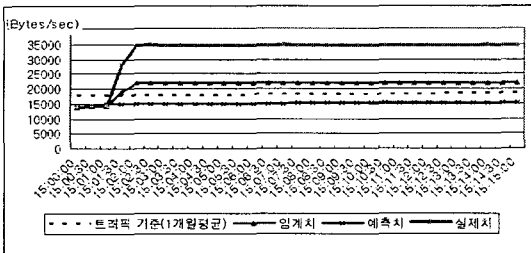


그림 11. UDP 플러딩 공격 트래픽($\alpha = 0.3, \delta = 0.05, \sigma = 7$)

UDP 플러딩 공격이 15:01:00~15:01:30에서 TCP 트래픽이 폭주하여 15:01:30에서 정상트래픽한계를 벗어나 공격이 검출된 공격트래픽 구간이 시작된다. 피해 시스템에 공격트래픽을 대량으로 전송하여 트래픽이 일시에 폭등하는 트래픽이 분포한다.

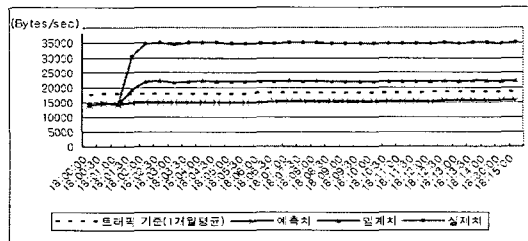


그림 12. ICMP 플러딩 공격 트래픽($\alpha = 0.3, \delta = 0.05, \sigma = 7$)

ICMP 플러딩 공격이 18:01:00~18:01:30에서 ICMP 트래픽이 폭주하여 18:01:30에서 정상트래픽한계를 벗어나 공격 검출된 공격트래픽 구간이 시작되

고 있으며 30초 간격으로 기존방식의 탐지보다 신속한 탐지가 이루어진다.

각각의 $\alpha=0.3, \alpha=0.15, \alpha=0.05$ 로 구분하여 예측치를 구하고 정상트래픽 한계와 같이 표현하면 같은 구간에서 다음과 같은 임계치와 공격 트래픽이 분포한다. α 의 값을 변경하며 실험한 결과 다음과 같은 임계치 이상의 공격 구간이 분포되었다.

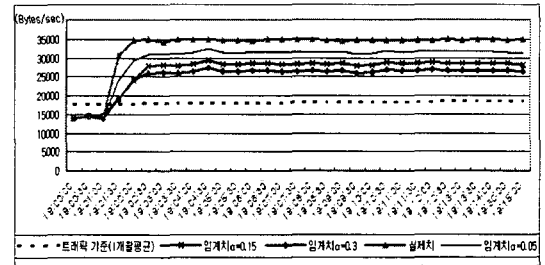


그림 13. $\alpha = 0.3, \alpha = 0.15, \alpha = 0.05$ 의 임계치와 공격 트래픽

지수 평활계수를 α 경우 0.3~0.05 사이의 값을 주면 임계치(정상트래픽한계)가 달라진다. 적합한 구간은 α 의 경우 0.05으로 값을 사용하는 것이 효율적이다.

표 2. 성능 분석 결과

평활상수(α)	분산평활상수(δ)	시그마계수(σ)	탐지율(%)
0.3	0.05	5	75
0.3	0.05	7	89
0.05	0.05	5	97
0.05	0.05	7	96
0.05	0.005	7	94

평균적인 평활계수를 구하기 위해 하루 50번 한달 동안 공격을 수행하여 적합한 계수의 수치를 표 2과 같이 구하였다. 평활상수(α)는 0.05, 분산평활상수(δ)는 0.05를 시그마계수(σ)는 5를 사용하는 것이 공격 탐지율 97%의 가장 높은 검출율을 갖고 있다.

$\alpha=0.05, \delta=0.05, \sigma=5$ 를 사용하여 침입을 검출하는 것이 가장 높은 공격 검출이 이루어지므로, 평활계수 수치를 고려하여 효율적으로 검출되어 공격 검출에 평활계수 수치를 고려하여 공격 검출에 사용하여야 효율적인 침입 판정이 이루어진다.

V. 결론

본 논문은 지수 평활법을 적용하여 실제 트래픽으로 구해진 예측치로 임계치를 초과하는 입력 트래픽은 공격으로 판별하고 정상트래픽 한계 즉 임계치를 각각의 MIB인 tcpInSegs, udpInDatagrams, ipv6IfIcmp OutEcho Replies을 30초 간격으로 트래픽을 수집하여 TCP 플러딩, UDP 플러딩, ICMP 플러딩 공격을 검출한다. 또한 이에 보다 정확한 검출을 위해 공격에 사용되는 기존공격 포트번호를 ipv6TcpConnLocalPort, ipv6UdpLocalPort을 이용하여 공격을 판별하고, 수신되는 멀티캐스트 메시지의 지수 평활법 임계치를 구하여 멀티캐스트 메시지 공격인지 판별하였다.

향후 침입탐지시스템이나 침입방지시스템에 적용하여 사용할 수 있을 것이다.

참고 문헌

[1] 한국전산원, IPv6 Status Report IPv6 동향, 2004.
 [2] 유대성, 오창석, “공격 탐지를 위한 트래픽 수집 및 분석 알고리즘”, 한국콘텐츠학회 논문지, 제4권, 제4호, pp.33-43, 2004.
 [3] 이홍규, 구향옥, 김선영, 김영기, 오창석, “IPv6 기반 자동화된 공격 대응도구”, 한국컴퓨터정보학회 논문지, 제10권, 제3호, pp.249-257, 2005.
 [4] 구향옥, 백순화, 오창석, “IPv6 환경에서의 유해 트래픽 분석”, 한국콘텐츠학회 2005 추계 종합 학술대회 논문집, 제3권, 제2호, pp.195-199, 2005.
 [5] <http://www.net-snmp.org/>
 [6] <http://ipv6.lghitachi.co.kr/manual/korean/gs4000/HTML/MIBREF/>
 [7] D. Risteski, A. Kulakov, and D. Davcev, “Single Exponential Smoothing Method and Neural Network in One Method for Time Series Prediction,” Cybernetics and Intelligent Systems, 2004 IEEE Conference

on, Vol.2, pp.741-745, 2004.

[8] 나중찬, 실시간 트래픽 분석에 의한 예측모형 기반 네트워크 이상징후 탐지 기법, 충남대학교 대학원 박사학위논문, 2003.
 [9] Bennett Todd, Distributed Denial of Service Attacks, http://www.linuxsecurity.com/resource_files/intrusion_detection/ddos-faq.html
 [10] 오창석, 생동하는 TCP/IP 인터넷, 내하출판사, 2004.

저자 소개

구향옥(Hyang-Ohk Koo)

중신회원



- 1999년 8월 : 한밭대학교 전자계산학과(이학사)
- 2002년 2월 : 충북대학교 컴퓨터공학과(공학석사)
- 2002년~현재 : 충북대학교 컴퓨터공학과 박사과정

• 2003년 8월~현재 : 백석대학 겸임

<관심분야> : 컴퓨터네트워크, 뉴로컴퓨터, 정보보호

오창석(Chang-Suk Oh)

중신회원



- 1978년 2월 : 연세대학교 전자공학과(공학사)
- 1980년 2월 : 연세대학교 전자공학과(공학석사)
- 1988년 8월 : 연세대학교 전자공학과(공학박사)

• 1985년~현재 : 충북대학교 전기전자컴퓨터공학부 교수

• 1982년~1984년 : 한국전자통신연구원 연구원

• 1990년~1991년 : Stanford대학교 객원교수

• 2001년~2004년 : 한국콘텐츠학회 논문지 편집위원장

• 2004년~현재 : 한국콘텐츠학회 상임고문

<관심분야> : 컴퓨터네트워크, 뉴로컴퓨터, 정보보호