트래픽 흐름 분석을 이용한 감염된 시스템 탐지 기법*

이재국° 김형식
충남대학교 대학원 컴퓨터공학과
{ empire°, hskim }@cs.cnu.ac.kr

An Infected System Detection Scheme to Use Traffic Flow Analysis

Jae-Kook Lee° Hyong-Shik Kim
Dept. of Computer Engineering, Chungnam National University

요 약

네트워크 환경의 발달과 더불어 DDoS 공격이나 웜 공격이 증대되고 있다. 다양한 공격의 증가뿐만 아니라 최근에는 공격이 발생하면 급속히 피해가 확산된다. 피해 속도가 빨라지는 이유 중의 하나는 피해 시스템이 공격자가 되기 때문이다. 그러나 만약 피해 시스템이 또 다른 공격 시스템이 되는 것을 차단할 수 있다면, 공격이 확산되는 속도를 늦출 수 있다.

본 논문에서는 감염된 시스템이 비정상적으로 많은 트래픽을 발생시키는 것을 탐지하기 위하여 특정 주소를 갖는 시스템으로 일정 기간 동안 들어오고 나간 인바운드 패킷과 아웃바운드 패킷의 양을 비율로 나타내어 트래픽 흐름을 분석한다. 그리고 B-클래스 네트워크에서 추출한 트래픽 샘플데이터를 이용하여 트래픽 흐름을 분석하여 감염된 시스템을 탐지할 수 있음을 보인다.

1. 서 론

네트워크 환경의 급속한 발달과 더불어 분산 서비스 거부(Distributed Denial of Service:DDoS) 공격이나 웜(Worm) 공격이 증대되고 있다. DDoS 공격이나 웜 공격은 인테넷에 개방되어 있으면서 동시에 한정된 자원(네트워크의 대역폭, 시스템의 패킷 처리 용량, 시스템에 도착한 패킷의 처리를 위하여 이용되는 시스템 자원 등)을 가진 모든 시스템을 쉽게 공격의 대상으로 하여 피해를 입힌다[1,2]. 이러한 공격이 다양화 되는 동시에 최근에는 공격이 발생하면 기하급수적으로 피해가 확산된다. 피해 확산이 점점 빨라지는 이유 중에 하나는 네트워크에 연결되어 있는 개별 시스템이 자신도 모르는 사이에 피해자가 되고, 동시에 다른 시스템을 공격하는 공격자로 돌변하기 때문이다. 공격시스템으로부터 자신의 시스템을 보호하는 것이 보안의 가장 중요한 이슈이다. 네트워크 환경에서 공격을 탐지하기 위하여 프로파일(Activity Profiling)[1,3]을 이용하거나 누적(Cumulative Sum:Cusum) 알고리즘[1,4,5,6]을 이용한 기법과 웨이블릿(Wavelet Approache)[1,7,8,9]을 이용한 많은 기법들이들이 제안되었다.

이처럼 공격으로부터 시스템을 보호하는 것이 중요하다 하지만, 공격을 받은 피해 시스템이 또 다른 공격 시스템이 되는 것을 차단할 수 있다면, 공격이 확산되는 속도를 늦출 수 있을 것이다.

본 논문에서는 소규모 네트워크에서 시스템의 취약성을 이용한 공격이나 DDoS 공격에 의하여 감염된 시스템이 비정상적으로 많은 트래픽을 발생시키는 것을 탐지하기 위하여 특정 주소의 인바운드 패킷과 아웃바운드 패킷의 트래픽 흐름을 분석한다. 그리고 실제 B-클래스 네트워크에서 'tcpdump'[10]를 이용하여 추출한 트래픽 샘플데이터에 트래픽 흐름 분석을 적용하여 나타나는 특징을 살펴보고, 감염된 시스템을 탐지하기 위한 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2절에서는 감염된 시스템의 특징을 간단히 살펴보고, 3절에서는 특정 주소에서의 트래픽 흐름 분석을 위한 알고리즘을 제안한다. 4절에서는 실제 B-클래스 네트워크의 트래픽 샘플데이터를 이용하여 트래픽 흐름 분석을 적용하고, 정상 시스템과 비정상 시스템이 갖는 트래픽 흐름 분석 결과를 확인한다. 5절에서는 감염된 시스템 탐지를 위한 수식을 정리고, 끝으로 6절에서는 결론을 내리고 향후 연구과제에 대하여 알아본다.

*본 연구는 "대학 IT 연구센터 육성/지원사업"의 지원을 받아 수행한 연구 결과임

2. 감염 형태 분석

침입이나 시스템 취약성을 이용한 공격에 의해 감염된 시스템은 다른 시스템을 대상으로 그림 1의 (a)와 같이 TCP SYN, UDP, ICMP와 같은 네트워크 취약성을 이용한 플로딩(Flooding) 공격과 같은 DDoS 공격을 일으키거나, 그림 1의 (b)와 같이 웜에 감염된 시스템이 또 다른 취약성 시스템을 찾기 위해 스캐닝 공격을 일으키게 된다. 이와 같이, 감염된 시스템은 피해자이면서 동시에 다른 시스템이나 네트워크를 공격하는 가해자가 되어 피해 확산 속도가 지수함수를 나타내게 된다. 그러나, 만약 네트워크 내부의 보안 어플라이언스나 라우터와 같이 네트워크 내부 시스템과 인터넷을 연결해 주는 게이트웨이와 같은 시스템이 감염된 시스템을 탐지하고 격리할 수 있다면 악순환을 최소화할 수 있을 것이다.

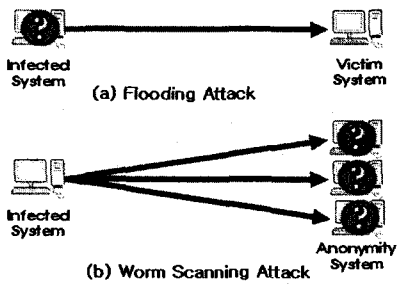


그림 1 감염된 시스템

3. 트래픽 흐름 분석

그림 1과 같이 감염된 시스템은 취약점이 있는 시스템을 공격하거나 취약성 있는 시스템을 찾기 위하여 스캐닝을 하거나 DDoS 공격을 일으키게 된다. 이렇게 되면 감염된 시스템에서 외부로 나가는 아웃바운드 트래픽은 증가하고 외부에서 들어오는 인바운드 트래픽은 상대적으로 작아질 것이다. 그림 2에서 보이는 것과 같이 임의의 시스템이 패킷을 받게 되면 그에 대응하는 응답 패킷을 보내게 된다[3].

그러나 DDoS나 웜과 같은 공격은 이러한 정상적인 네트워크의 흐름을 방해하거나 실제 존재하지 않는 시스템에 패킷을 보내게 되어 응답이 없는 패킷이 증가하게 된다.

Packet sent	Response from victim
TCP SYN (to open port)	TCP SYN ACK
TCP SYN (to closed port)	TCP RST (ACK)
TCP ACK	TCP RST (ACK)
TCP DATA	TCP RST (ACK)
TCP RST	no response
TCP NULL	TCP RST (ACK)
ICMP ECHO Request	ICMP Echo Reply
ICMP TS Request	ICMP TS Reply
UDP pkt (to open port)	protocol dependent
UDP pkt (to closed port)	ICMP Port Unreach
...	...

그림 2 응답패킷 예제

따라서, 수식(1)과 같이 정상적인 경우라면 일정 기간 동안 특정 시스템에 들어온 인바운드 패킷의 양과 나가는 아웃바운드 패킷의 양이 비슷할 것으로 추정된다. 그러나 감염된 시스템인 경우라면 수식(2)와 같이 인바운드 패킷이나 아웃바운드 패킷이 상대적으로 많아진다.

$$\sum_{\Delta t} V_{in} \approx \sum_{\Delta t} V_{out} \qquad \text{수식(1)}$$

$$\sum_{\Delta t} V_{in} \gg \sum_{\Delta t} V_{out}\ ,\ \sum_{\Delta t} V_{in} \ll \sum_{\Delta t} V_{out} \qquad \text{수식(2)}$$

본 논문에서는 네트워크 내부 시스템이 감염되어 비정상적인 트래픽을 생성하게 되므로 아웃바운드 패킷의 양이 비정상 적으로 많은 것을 이상으로 탐지할 수 있을 것이다.

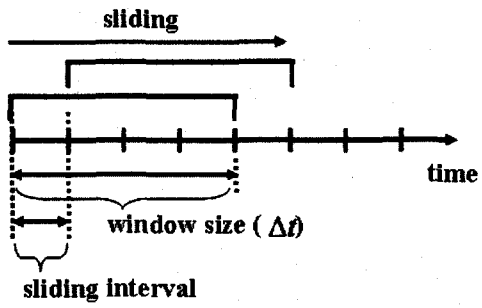


그림 3 비율을 구하기 위한 방법

특정 시스템의 인바운드 트래픽과 아웃바운드 트래픽의 흐름을 분석하기 위하여 그림 3에서와 같이 패킷량을 계산하기 위한 현시점에서 이전 시점까지의 일정 기간 동안을 윈도우 사이즈(window size(Δt))로 하고 슬라이

딩 간격(sliding interval) 만큼을 이동하면서 인바운드 패킷과 아웃바운드 패킷의 합계를 각각 계산한다. 이때 윈도우 사이즈와 슬라이딩 간격의 차이만큼을 중복되도록 한다.

보다 효과적으로 트래픽 흐름을 분석하기 위하여 그림 3과 같은 방식으로 인바운드 패킷과 아웃바운드 패킷의 양을 측정하고, 특정 IP 주소 addr에서의 트래픽 흐름을 수식(3)과 같이 비율로 나타낸다. 수식에 의하면 인바운드 트래픽이 많으면 '+'로 나타나고, 아웃바운드 트래픽이 많으면 '-'로 나타나게 된다.

$$r_{addr} = \begin{cases} 1 - \dfrac{MIN(V_{in}, V_{out})}{MAX(V_{in}, V_{out})} & , V_{in} \geq V_{out} \\ \dfrac{MIN(V_{in}, V_{out})}{MAX(V_{in}, V_{out})} - 1 & , V_{in} < V_{out} \end{cases}$$ 수식(3)

4. 트래픽 분석

본 절에서는 실제 B-클래스 네트워크에서 'tcpdump'를 이용하여 추출한 샘플데이터를 가지고 3절에서 제안한 기법을 적용하여 트래픽을 분석한다. 샘플데이터 분석을 위한 윈도우 사이즈는 20초로 하고, 슬라이딩 간격은 1초로 한다. 이때 그래프에 나타날 수 있는 노이즈를 없애기 위하여 인바운드 패킷과 아웃바운드 패킷의 개수가 100개가 되지 않는 경우는 0이 되도록 한다.

4.1 네트워크 서비스에 따른 트래픽 분석

트래픽 흐름 분석을 위해 사용된 샘플데이터는 2005년 3월 2일에 B-클래스 네트워크의 라우터에서 'tcpdump'를 이용하여 추출 데이터이다.

그림 4와 5의 (a)는 13시21분31초 (b)는 13시53분29초 (c)는 14시21분59초에 트래픽 흐름 분석을 수행한 결과를 나타낸다. 그림 4는 HTTP 응답 패킷을 보내는 웹 서버로 추정되는 시스템의 트래픽 흐름 분석을 수행한 것이다. 비율(r_{addr})이 -0.7보다 크고 0.5보다 작은 범위에서 정해진 것을 확인할 수 있다. 그림 5는 MS-SQL 포트를 통해 통신을 하는 시스템의 트래픽 흐름을 분석한 것이다. 비율이 ±0.4 사이에서 나타난 것을 확인할 수 있다. 이 이외에 FTP 서버로 추정되는 시스템, P2P 서비스 중에서 eDonkey에 의한 서비스를 수행중인 시스템, 그리고 SMTP 서비스를 해주고 있는 시스템의 트래픽 흐름 분석을 진행하였다. 전체적으로 정상 상태일 때의 시스템의 비율은 -0.8보다 크고 0.7보다 작은 범위에서 나타났다.

4.2 감염된 시스템을 포함한 트래픽 분석

감염된 시스템을 포함한 트래픽 흐름 분석을 위해 사용된 트래픽은 B-클래스 네트워크 내부에 감염된 시스템이 비정상 트래픽을 유발했던 2004년 5월19일 'tcpdump'를 이용하여 추출한 것이다. 그림 6의 (a)는 14시37분28초, (b)는 16시1분8초, (c)는 18시8분55초에 추출한 샘플이다.

비정상 트래픽이 포함된 샘플데이터의 트래픽 흐름 분석을 위하여 트래픽 양이 많은 상위 20개의 특정 주소를 선택하고, 트래픽을 분석하였다. 그림 6에서 보이는 것과 같이 20개 중에 17개의 트래픽 흐름 분석 결과는 -0.5보다 크고 0.7보다 작은 정상 상태일 때와 비슷한 그래프가 나타났다.

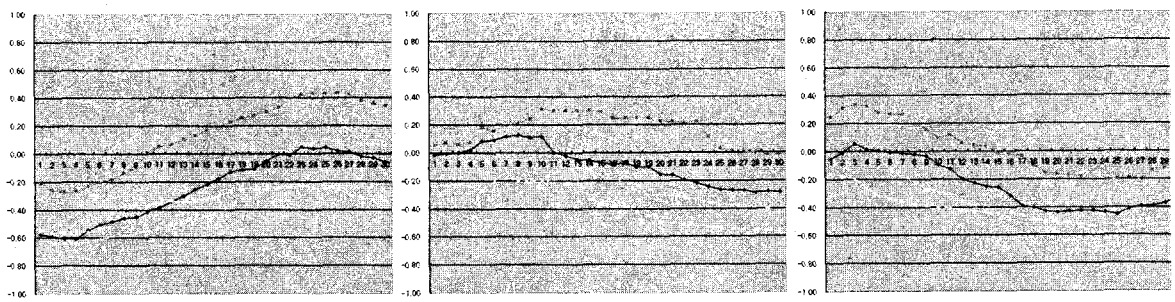
그러나, 하나의 그래프는 +1에 근접하여 나타났고, 2개는 -1에 근접하여 그래프가 나타난 것을 확인할 수 있다. 이들 그래프를 나타내는 IP 주소의 트래픽을 확인한 결과 -1에 가까운 두개의 그래프 중 하나 TCP SYN 플로딩 공격(-0.98 ~ -1)을 하는 시스템이었고 다른 하나는 MS-SQL 원 스캐닝 공격(-1)을 하는 시스템이었다.

이처럼 트래픽 흐름 분석을 이용하여 비정상적으로 많은 트래픽을 내보내는 감염된 시스템을 확인할 수 있었다. 반면에 인바운드 트래픽이 많아 +1에 나타났던 시스템은 MS THEATER 서비스를 받고 있는 정상 시스템이었다.

5. 감염된 시스템 탐지

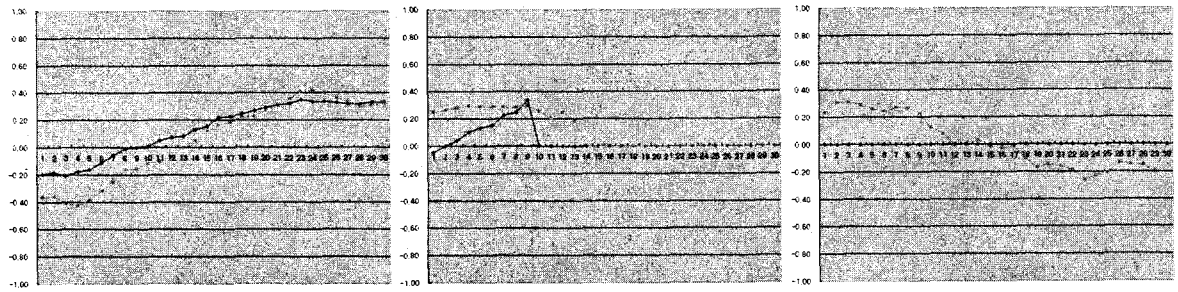
4절에서 분석한 것과 같이 트래픽 흐름 분석을 이용하면 네트워크 내부에서 감염된 시스템을 탐지할 수 있다. IP 주소 addr을 갖는 시스템의 인바운드 트래픽과 아웃바운드 트래픽의 비율이 수식(4)와 같이 비율이 임계치(α) 보다 큰 경우를 비정상으로 판단 한다.

$$|r_{addr}| > \alpha$$ 수식(4)



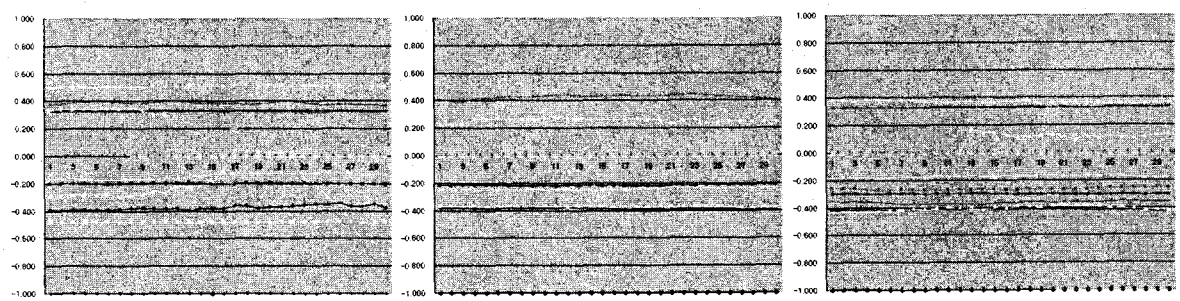
(a) (b) (c)

그림 4 HTTP 시스템의 트래픽 흐름 분석



(a) (b) (c)

그림 5 MS-SQL 시스템의 트래픽 흐름 분석



(a) (b) (c)

그림 6 감염된 시스템이 포함된 트래픽 데이터의 트래픽 흐름 분석

반대로 수식(5)와 같이 트래픽 흐름을 분석한 비율이 임계치보다 작은 경우 시스템이 정상 상태라고 할 수 있다.

$$|r_{addr}| \leq \alpha \qquad \text{수식(5)}$$

예를 들어 4절에서 트래픽 흐름 분석한 결과를 바탕으로 임계치를 0.9로 잡는다면 플로딩 공격을 하는 시스템과 웜 스캐닝 공격을 하는 두 시스템을 실제로 탐지할 수 있다.

임계치는 구현 단계에서 네트워크 환경의 특성을 반영하도록 유동적인 값을 갖는 변수로 한다면 보다 효과적으로 감염된 시스템을 탐지할 것이다. 그러나 이때 임계

치의 값을 너무 작게 잡는다면 오탐지율이 증가하게 되고 반대로 임계치를 너무 크게 잡으면 미탐지율이 증가하게 됨으로 주의해야 한다.

6. 결론 및 향후 연구과제

네트워크 내부 시스템이 웜 및 바이러스 공격에 의하여 감염되게 되면 또 다른 공격 시스템이 되어 비정상적으로 많은 트래픽을 발생하게 된다. 발생된 비정상 트래픽은 정상적인 연결을 맺고 통신이 제대로 이루어질 확률이 적다.

본 연구에서는 소규모 네트워크 환경에서 특정 주소에서의 트래픽 흐름을 분석을 이용하여 감염된 시스템을 탐지하기 위한 기법을 제안했으며, 실제 B-클래스를 갖는 네트워크 환경에서 정상상태일 때와 비정상상태일 때 각각 추출한 샘플 데이터의 인바운드와 아웃바운드 트래픽 흐름 분석하였다. 분석한 결과 내부 시스템에서 비정상 대규모 트래픽을 유발하는 시스템을 탐지할 수 있었다.

향후 트래픽 흐름 분석을 이용한 감염된 시스템 탐지 기법을 보안 어플라이언스와 같은 보안 장비에 구현하기 위한 연구를 계속할 것이며, 내부 시스템의 감염을 방지하기 위한 트래픽 흐름 분석 기법의 활용 방안도 모색할 것이다.

[참고문헌]

[1] 이철호, 최경희, 정기현, 노상욱, "웹 서버에 대한 DDoS 공격의 네트워크 트래픽 분석," 한국정보처리학회 논문지 C, 제10-c권, 제3호, June 2004.

[2] Glenn Carl, George Kesidis,Richard R. Brooks and Suresh Rai, "Denial-of-Service Attack-Detection Techniques," IEEE Internet Computing, Jan., Feb. 2006.

[3] D. Moore, G.M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," Proc. Usenix Security Symp.,Usenix Assoc., 2001.

[4] R.B. Blazek et al., "A Novel Approach to Detection of 'Denial-of-Service' Attacks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods," Proc.IEEE Workshop Information Assurance and Security, IEEE CS Press, 2001, pp. 220-226.

[5] H. Wang, D. Zhang, and K. Shin, "Detecting SYN Flooding Attacks," Proc. 21st Joint Conf. IEEE Computer and Comm. Societies (IEEE INFOCOM), IEEE Press, 2002, pp.1530-1539.

[6] L Feinstein et al., "Statistical Approaches to DDoS Attack Detection and Response," Proc. DARPA Information urvivability Conf. and Exposition, vol. 1, 2003, IEEE CS Press, pp. 303-314.

[7] Patrice Abry and Darryl Veitch, "Wavelet Analysis of Long Range Dependent Traffic," in IEEE Transactions on Information Theory, vol.44, Jan. 1998. pp 2-15.

[8] P. Barford et al., "A Signal Analysis of Network Traffic Anomalies," Proc. ACM SIGCOMM Internet Measurement Workshop, ACM Press, 2002, pp. 71-82.

[9] R.R. Brooks, Disruptive Security Technologies with Mobile Code and Peer-to-Peer Networks, CRC Press, 2005.

[10] tcpdump/libpcap: http://www.tcpdump.org