ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 41, No. 9, pp. 617-624, 2014. 9
http://dx.doi.org/10.5626/JOK.2014.41.9.617

X-means 클러스터링을 이용한 악성 트래픽 탐지 방법

(A Malicious Traffic Detection Method Using X-means Clustering)

한 명 지 † (Myoungji Han) 임 지 혁 † (Jihyuk Lim) 최 준 용 †† (Junyong Choi) 김 현 준 †† (Hyunjoon Kim)
서 정 주 † (Jungjoo Seo) 유 철 †† (Cheol Yu) 김 성 렬 ††† (Sung-Ryul Kim) 박 근 수 †††† (Kunsoo Park)

요 약 악성 트래픽은 디도스 공격, 봇넷 통신 등의 인터넷 망을 교란시키거나 특정 네트워크, 서버, 혹은 호스트에 피해를 끼칠 의도를 가지고 발생시키는 트래픽을 지칭한다. 이와 같은 악성 트래픽은 인터넷이 발생한 이래 꾸준히 양과 질에서 진화하고 있고 이에 대한 대응 연구도 계속되고 있다. 이 논문에서는 악성 트래픽을 기존 X-means 클러스터링 알고리즘을 적용하여 효과적으로 탐지하는 방법을 제시하였다. 특히 악성 트래픽의 통계적 특징을 분석하고 클러스터링을 위한 메트릭을 정의하는 방법을 체계적으로 제시하였다. 또한 두 개의 공개된 트래픽 데이터에 대한 실험을 통해 실효성을 검증하였다.

키워드: 악성트래픽, 디도스공격, 봇넷, 클러스터링, 메트릭

Abstract Malicious traffic, such as DDoS attack and botnet communications, refers to traffic that is generated for the purpose of disturbing internet networks or harming certain networks, servers, or hosts. As malicious traffic has been constantly evolving in terms of both quality and quantity, there have been many researches fighting against it. In this paper, we propose an effective malicious traffic detection method that exploits the X-means clustering algorithm. We also suggest how to analyze statistical characteristics of malicious traffic and to define metrics that are used when clustering. Finally, we verify effectiveness of our method by experiments with two released traffic data.

Keywords: malicious traffic, DDoS attack, botnet, clustering, metrics

· 이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대 정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No.2011-0029924)

† 비 회 원 : 서울대학교 컴퓨터공학부
mjhan@theory.snu.ac.kr
jhlim@theory.snu.ac.kr
jjseo@theory.snu.ac.kr
†† 학생회원 : 서울대학교 컴퓨터공학부
jychoi@theory.snu.ac.kr
hjkim@theory.snu.ac.kr
rcheol@theory.snu.ac.kr
††† 종신회원 : 건국대학교 인터넷미디어공학부 교수
kimsr@konkuk.ac.kr
†††† 종신회원 : 서울대학교 컴퓨터공학부 교수(Seoul National Univ.)
kpark@theory.snu.ac.kr
(Corresponding author임)

논문접수 : 2014년 2월 28일
(Received 28 February 2014)
논문수정 : 2014년 5월 17일
(Revised 17 May 2014)
심사완료 : 2014년 5월 30일
(Accepted 30 May 2014)

정보과학회논문지 제41권 제9호(2014. 9)

1. 서 론

악성 트래픽은 정상적인 어플리케이션에 의해 발생한 트래픽이 아닌 전체 인터넷 망을 교란시키거나 특정 네트워크, 서버, 혹은 호스트에 피해를 끼칠 의도를 가지고 발생시키는 트래픽을 지칭한다. 악성 트래픽은 인터넷이 발생한 이래 꾸준히 양과 질에서 진화하고 있고 이에 대한 대응 연구도 계속되고 있다[1-9].

대표적인 악성 트래픽인 디도스(DDoS, Distributed Denial of Service) 공격은 악성코드에 감염된 다수의 호스트가 공격자의 명령에 따라 표적 시스템에 대량의 트래픽을 전송하여 표적 시스템의 자원을 고갈시키거나 전산망을 마비시키는 공격 방법이다[6]. 디도스 공격은 여러 대의 감염된 호스트들이 동시에 공격하기 때문에 표적 시스템에 막대한 피해를 줄 수 있다. 공격 패킷의 source IP를 위조하거나 정상 트래픽을 흉내내는 등 그 방법이 점점 교묘해지고 있기 때문에 디도스 공격의 준비부터 실행까지 전 단계에서 빠르게 탐지하고 대처하는 연구가 필요하다.

디도스 공격을 수행하는 다수의 호스트들의 네트워크를 봇넷(botnet)이라고 한다. 봇넷에서 봇마스터는 C&C (Command and Control) channel을 통하여 봇들에게 명령을 내리고 봇들의 정보를 수신한다. 중앙집중형 봇넷(centralized botnet) 은 모든 봇들이 하나의 C&C server와 통신한다. 이 방법은 설치가 쉽고 강력하여 널리 사용되지만 single-point-of-failure 문제가 있다. 최근 이러한 문제에 대응하기 위해 P2P 통신을 이용하여 봇 사이에 메시지를 전달하는 방식의 P2P 봇넷이 등장했다[10,11]. 봇넷은 디도스 공격 뿐만 아니라 스팸 메일, 악성코드 전파, 개인정보 유출, 피싱 등의 각종 사이버 범죄에 이용된다. 봇넷의 통신 트래픽 역시 악성 트래픽에 해당되며 이를 조기에 발견하여 차단하면 사이버 공격을 원천적으로 차단할 수 있다.

공격의 기법이 나날이 진화함에 따라 악성 트래픽을 탐지 및 역추적 하는 연구 또한 매우 많이 진행되었다. 그 중 본 연구와 직접적으로 관련 있는 연구는 다음과 같다. K. Lee 등은 일반적인 DDoS 공격의 각 단계별 특징을 고려한 메트릭을 정의한 뒤 계층적 응집 클러스터링(agglomerative hierarchical clustering)을 이용하여 트래픽을 DDoS 공격의 각 단계로 구별하였다[7]. S. Saad 등은 악성 트래픽을 탐지하기 위한 17개의 메트릭을 정의한 뒤 5개의 기계학습 기법을 사용하여 봇넷과 정상 트래픽으로 레이블링 된 데이터셋을 학습시키고 봇넷 트래픽을 탐지하였다[8]. H. Choi 등은 봇넷의 DNS 트래픽이 그룹 행동을 보인다는 점에 착안하여 각 도메인을 요청하는 DNS 쿼리들을 IP와 시간별로 나타내는 행렬을 생성한 뒤 X-means 클러스터링을 이용하여 C&C 서버를 탐지하였다[9].

이 논문에서는 이러한 알려진 악성 트래픽을 클러스터링 기법을 이용하여 효과적으로 탐지하는 방법을 제시한다. 특히 탐지하고자 하는 악성 트래픽이 주어지면 해당 트래픽의 통계적 특징을 분석하고 클러스터링을 위한 메트릭을 정의하는 방법을 체계적으로 제시한다. 또한 두 개의 공개된 트래픽 데이터에 대한 실험으로 이 방법의 실효성을 검증한다. 2장에서는 본 논문에서 사용한 X-means 클러스터링 알고리즘에 대하여 설명하고 3장에서 악성 트래픽 탐지 방법을 설명한다. 4장에서 실험에 사용한 두 개의 트래픽 데이터를 소개한 뒤 5장에서 두 데이터에 대하여 실험한 내용을 설명한다. 6장에서 실험 결과를 분석하고 7장에서 결론을 맺는다.

2. 클러스터링

일반적으로 많이 사용되는 클러스터링 기법으로는 K-means 알고리즘[12]과 계층적 클러스터링(hierarchical clustering)[13]이 있다. K-means 알고리즘은 각 클러스터의 포인트들로부터 그 중심점까지의 거리의 합을 최소화하는 K개의 중심점들을 찾는 최적화 알고리즘이다. 계층적 클러스터링은 클러스터들을 작은 것부터 응집(agglomeration)하거나 큰 것부터 분할(division)하는 방식을 통해 클러스터의 계층(hierarchy)을 만드는 방법이다. 클러스터의 계층은 덴드로그램(dendrogram)이라는 트리로 표현되는데 여기서 절단점(cut point)을 정하면 하나의 클러스터링 결과가 나온다. 두 방법은 모두 클러스터의 수가 주어져야 하기 때문에 사용자가 문제공간에 대한 충분한 사전지식을 가지고 있거나 경험적 분석을 통하여 적절한 값을 결정해야 한다. 클러스터 수의 최적값을 결정하는 여러 방법들이 있지만[14] 이 방법들은 대부분 큰 추가비용을 필요로 한다.

X-means 알고리즘[15]은 초기 K를 정해주면 K-means 클러스터링을 수행한 뒤, 각 클러스터를 두 개로 분할하는 split이라는 작업을 반복한다. 이때 split 전후의 BIC score를 비교하여 split 후의 BIC score가 개선되지 않으면 split을 멈춘다. 더 이상 split할 클러스터가 없으면 최종 클러스터를 반환한다. 이 방법은 K-means 알고리즘보다 효율적이며 좋은 결과를 낸다. 또한 알고리즘이 스스로 클러스터의 개수를 최적화한다.

3. 악성 트래픽 탐지 방법

본 논문의 악성 트래픽 탐지 방법은 크게 1)전처리 단계(preprocessing step)와 2)탐지 단계(detection step)로 나뉜다. 그림 1은 두 단계를 도표로 나타낸 것이다.

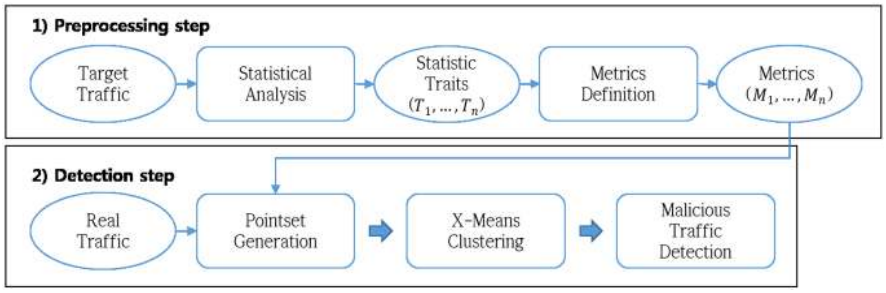

그림 1 악성 트래픽 탐지 방법
Fig. 1 Malicious traffic detection

전처리 단계에서는 먼저 탐지하고자 하는 표적 트래픽을 결정한다. 표적 트래픽은 스캐닝, 악성코드 설치, 봇넷 통신, 디도스 공격 등을 수행하는 악성 트래픽이 될 수 있다. 전처리 단계는 [Statistical Analysis] 단계와 [Metrics Definition] 단계로 나누어진다.

[Statistical Analysis] 단계에서는 표적 트래픽의 통계적 특징을 분석한다. 이 특징은 표적 트래픽만의 고유한 특징일 필요는 없고 표적 트래픽이 가지는 모든 특징을 찾으면 된다. 만약 어떤 특징이 우연히 정상 트래픽에서도 나타날 수 있다 하여도 다른 특징들에 의해 충분히 구별될 수 있다. 이렇게 찾은 통계적 특징들을 T_1, T_2, ..., T_n 이라고 하자.

[Metrics Definition] 단계에서는 각 통계적 특징들을 표현할 수 있는 메트릭 M_1, M_2, ..., M_n을 정의한다. 이 때 M_i (i=1,...,n)는 시간 간격, 플로우, 호스트 등의 단위에서 T_i (i=1,...,n)를 잘 표현할 수 있는 통계 식으로 정의한다. 이 때 표적 트래픽에 대하여 각 메트릭 M_i (i=1,...,n)을 계산한 값 m_i (i=1,...,n)는 정상 트래픽에 대하여 계산한 값에 비하여 상대적으로 매우 크거나 매우 작은 값이 나와야 한다.

탐지 단계에서는 탐지하고자 하는 실제 트래픽이 주어졌을 때 앞에서 결정한 표적 트래픽을 탐지한다. 탐지 단계는 [Pointset Generation] 단계, [X-Means Clustering] 단계, [Malicious Traffic Detection] 단계로 나누어진다.

[Pointset Generation] 단계에서는 [Metrics Definition] 단계에서 정의한 메트릭을 계산한 값들 m_1, m_2, ..., m_n을 계산하여 포인트 p=(m_1, m_2, ..., m_n)들을 생성한다. 트래픽으로부터 모든 포인트가 생성되면 포인트의 값을 각 축에 대하여 [0,1] 사이로 정규화 한다.

[X-Means Clustering] 단계에서는 전체 포인트셋을 X-means 알고리즘으로 클러스터링 한다. 앞에서 정의한 메트릭에 의하여 표적 트래픽을 포함하는 포인트 간의 거리는 상대적으로 가까워 동일 클러스터에 포함될 확률이 크다.

[Malicious Traffic Detection] 단계에서는 클러스터링 결과를 분석하여 표적 트래픽에 해당하는 클러스터를 찾는다. 이는 클러스터의 모든 포인트의 평균점을 계산하여 앞서 표적 트래픽이라면 크거나 작을 거라 예상했던 바에 가장 근접한 클러스터를 찾는 것이다.

4. 데이터셋

악성 트래픽 탐지 실험을 위하여 MIT Lincoln Lab의 Darpa 2000 Dataset[16]과 University of Victoria의 ISOT Research Lab의 ISOT Botnet Dataset[17]을 사용하였다.

Darpa Dataset은 미 국방성 DARPA가 침입탐지 시스템의 성능평가를 위해 만든 실험 데이터로서 특히 Darpa 2000 Dataset은 악성코드를 유포하고 디도스 공격을 수행하는 전체 시나리오를 포함하고 있다. 공격 시나리오는 총 5단계로 이루어졌으며 각 공격 단계에 해당하는 패킷들은 레이블링되어 있다.

ISOT Botnet Dataset은 빅토리아대의 ISOT Lab에서 공개한 데이터로 정상 트래픽과 P2P 봇넷의 트래픽을 병합해 만든 실험용 데이터이다. 이 데이터에서는 총 20,120개의 호스트가 1,515,185개의 플로우를 발생시킨다. 그 중 3개의 호스트가 봇이다. 각 봇의 역할은 표 1과 같다.

표 1 봇 호스트
Table 1 Bot hosts

Host	Description
172.16.2.11	Storm : UDP
172.16.0.11	Waledac : SMTP Spam
172.16.0.12	Storm : SMTP Spam

5. 실험

이 장에서는 본 논문의 기법을 적용하여 디도스 공격

과 P2P 봇넷 트래픽에 대하여 악성 트래픽을 탐지하는 과정에 대하여 설명한다.

5.1 디도스 공격 탐지 실험

Darpa Dataset에서 가정하는 디도스 공격 시나리오는 총 5단계로 이루어졌으며 표 2와 같다.

표 2 디도스 공격 시나리오
Table 2 DDoS attack scenario

Phase	Description
phase 1	IPsweep using ICMP
phase 2	trying remote access using portmap
phase 3	getting remote access
phase 4	DDoS software installation
phase 5	launching the DDoS

1단계에서 IP 탐색을 하기 위해 공격자는 여러 IP 주소로 ICMP 패킷인 ping을 전송한다. 공격자는 응답으로 오는 패킷으로 해당 IP 컴퓨터의 동작 여부 및 운영체제 기본 정보를 알 수 있다. 1단계의 특징은 ICMP 패킷 비율이 높고(T_1) 한 source IP가 접근하는 IP의 개수는 상당히 많다는 것이다(T_2). 2단계에서 공격자는 1단계에서 접근 가능하다고 판단된 IP 주소에 접근하여 원격연결을 시도한다. 이때 portmap을 사용하기 때문에 다른 단계에 비하여 portmap 접속이 빈번하며(T_3) 무작위적으로 접속을 시도하며 접속에 실패한 경우 여러번 재시도를 하기 때문에 portmap 접속 재시도 비율이 높다(T_4). 3, 4단계는 원격접속을 하고 디도스 공격 파일을 설치하는 단계이다. 이 단계들은 일반 트래픽과 성질이 비슷하여 통계적으로 구분하기 힘들다. 5단계는 공격을 수행하는 단계이므로 여러 감염된 호스트들은 표적 서버에 source IP 정보를 조작하여 대량의 패킷을 전송한다. 이때 트래픽의 특징은 source IP의 수가 많고(T_5) 트래픽의 양이 많다는 것이다(T_6).

위에서 분석한 각 단계의 통계적 특징들로부터 각 단계를 구분할 메트릭을 정의한다. 표 3은 6개의 메트릭을 나타낸 표이다. 각 메트릭은 정해진 시간 간격 T에서의 통계치를 계산하는 식이다.

M_1은 T 단위 동안 전체 패킷 중 ICMP 패킷이 차지하는 비율로 $\frac{ICMP\text{ 패킷수}}{\text{전체 패킷수}}$로 정의한다. Source IP 당 최대 destination IP 수, source IP 당 최대 portmap 패킷 수를 측정하는데, T 단위에 나타나는 source IP를 $src_1, src_2, \ldots, src_n$라 하자. Source IP src_i가 패킷을 보낸 destination IP의 집합의 크기를 d_i라 하자. 그리고 src_i가 portmap으로 보낸 패킷의 수를 $portmap_i$라 하자. 이때 M_2는 $\max(\{d_i | 1 \le i \le n\})$, M_3은 $\max(\{portmap_i | 1 \le i \le n\})$

표 3 디도스 공격 탐지를 위한 메트릭
Table 3 Metrics for DDoS attack detection

Metric	Description
M_1	rate of ICMP packets
M_2	maximum number of destination IPs for one source IP
M_3	maximum number of portmap packets for one source IP
M_4	rate of portmap access repeat
M_5	entropy of source IPs
M_6	number of packets

로 정의한다. 그 다음으로 portmap 접속 재시도 비율을 계산한다. 어떤 호스트가 T 단위 안에서 다른 호스트의 portmap 포트로 서비스요청을 보낸 패킷의 수를 $portmap_{total}$이라 하고 portmap 서비스 요청이 성공한 횟수를 $portmap_{succ}$라 하자. M_4는 $\frac{portmap_{total} - portmap_{succ}}{portmap_{total}}$로 정의한다. 이후 측정하는 통계는 source IP 엔트로피로, T 단위에서 나타나는 source IP $src_1, src_2, \ldots, src_n$에 대해서 src_i가 나타나는 비율을 p_i라 하자. M_5는 source IP들의 엔트로피(H)로서 $H = -\sum_{i=1}^{n} p_i \log_2 p_i$와 같이 계산한다. 마지막으로 M_6는 T 단위 동안 전송된 모든 패킷들의 수이다.

앞에서 분석한 바에 따르면 디도스 공격 1단계에 해당하는 포인트는 M_1, M_2의 값이 크고, 2단계에 해당하는 포인트는 M_3, M_4의 값이 크고, 5단계에 해당하는 포인트는 M_5, M_6의 값이 클 것이다.

Darpa 2000 Dataset의 전체 트래픽으로부터 T를 10초로 설정하고 위 메트릭을 사용하여 6차원의 포인트를 생성하니 총 1171개의 포인트셋이 생성되었다. 이 포인트셋을 입력으로 X-means 클러스터링을 수행하였다.

실험의 유효성을 확인하기 위해 각 포인트가 디도스 공격의 어느 단계를 대표하는지 결정하여 별도로 레이블링을 해주었다. T 단위에 공격 패킷이 하나라도 들어있으면, 각 공격 패킷의 단계 중 그 개수가 가장 많은 단계로 포인트를 레이블링 하였다. 만약 T 단위에 공격 패킷이 하나도 들어있지 않으면 'normal'로 레이블링해 주었다. 공격 3, 4단계는 'normal'이라고 가정했다. 대부분(약 94%)의 포인트가 'normal'로 레이블링 되었다.

5.2 P2P 봇넷 트래픽 탐지 실험

P2P 통신은 대개 TCP, UDP 프로토콜을 사용하여 이루어진다. 일반적으로 P2P 통신에서 control traffic, query, query reply 등은 UDP 프로토콜을 사용하고 실제 데이터 전송은 TCP 프로토콜을 사용한다[18].

P2P 봇넷이 TCP 프로토콜을 사용하는 경우(이하 TCP 봇넷)는 SMTP 프로토콜을 이용한 스팸메일 또는

컨트롤 메시지 및 악성 코드를 배포하는 경우가 있다. 전송되는 패킷 평균 길이가 크며(T_{T1}) 다양한 메시지 및 SMTP 명령어를 사용하기 때문에 패킷의 길이 종류가 많다(T_{T2}). 또한 수많은 스팸과 피어 사이에 메시지를 빈번하게 전송하기 때문에 플로우의 개수가 많으며(T_{T3}) 내용을 보내는 destination IP가 많다(T_{T4}).

P2P 봇넷이 UDP 프로토콜을 사용하는 경우(이하 UDP 봇넷)는 피어와 연결 여부를 확인하는 경우이다. 피어 사이의 봇넷 감염 시도 및 명령 전송, 봇 마스터와의 통신 등 다양한 통신 패턴을 보인다(T_{U1}). P2P 어플리케이션과 달리 서로 알려진 봇넷 사이에 지속적인 연결이 많기 때문에 각 플로우에 패킷 수가 많으며(T_{U2}) 전송한 destination IP에 비해 플로우 수가 많고(T_{U3}) 연결이 성공한 플로우의 비율이 높다(T_{U4}). 또한 전송 시 다양한 포트를 사용하기 때문에 마찬가지로 전송한 목적지 IP에 비해 source port의 개수가 많다(T_{U5}).

위 분석을 토대로 TCP 봇넷과 UDP 봇넷을 탐지하기 위한 메트릭을 정의한다. 봇넷에 감염된 호스트를 찾아내는 것이 목표이기 때문에 호스트 단위로 메트릭을 설정한다. 또한 TCP 프로토콜과 UDP 프로토콜의 역할이 다르기 때문에 프로토콜 별로 메트릭을 따로 설정한다. 결과적으로 각 호스트는 TCP 메트릭과 UDP 메트릭에 의하여 계산된 통계 값들을 가지게 된다.

동일한 P2P 어플리케이션이 발생시키는 트래픽은 같은 메시지와 프로토콜을 사용하기 때문에 플로우를 구성하는 패킷 수와 크기가 비슷하다. 그러므로 각 호스트에 대하여 플로우들을 (보낸 패킷수, 받은 패킷 수, 보낸 패킷 크기, 받은 패킷 크기)의 좌표로 변환하여 클러스터링하면 비슷한 통신 패턴을 가진 플로우들끼리 묶이게 된다. 단, 이때 각 호스트의 플로우들을 TCP 플로우 집합과 UDP 플로우 집합으로 나눈 뒤 따로 클러스터링 한다. 이를 플로우 클러스터링이라 한다.

플로우 클러스터링이 완료되면 하나의 호스트는 TCP, UDP 집합에 대하여 각각 복수개의 클러스터를 가질 수 있다. 표 4는 하나의 클러스터로부터 얻을 수 있는 메트릭들이다.

표 4 클러스터를 나타내는 메트릭
Table 4 Metrics representing clusters

Metric	Description	Protocol
APS	average packet size	TCP
NDPS	number of distinct packet sizes	TCP
NF	number of flows	TCP
ND	number of distinct destination IPs	TCP
ANPF	average number of packets in a flow	UDP
NDNF	number of destination IPs / number of flows	UDP

표 5 호스트를 나타내는 메트릭
Table 5 Metrics representing hosts

Metric	Description	Protocol
M_{T1}	maximum APS	TCP
M_{T2}	maximum NDPS	TCP
M_{T3}	maximum NF	TCP
M_{T4}	maximum ND	TCP
M_{U1}	number of clusters	UDP
M_{U2}	maximum ANPF	UDP
M_{U3}	minimum NDNF	UDP
M_{U4}	number of successful flows / number of destination IPs	UDP
M_{U5}	number of destination IPs / number of source ports	UDP

모든 클러스터에 대하여 표 4의 메트릭을 계산하고 이 값들을 이용하여 호스트를 대표하는 메트릭을 정의한다. 표 5는 호스트의 통계적 특징을 나타내는 메트릭들이다.

예를 들어 n개의 호스트 H_1, ..., H_n 중 한 호스트 H_i로부터 TCP 메트릭 중 하나인 M_{T1}을 계산해 보자. M_{T1}~M_{U5}는 하나의 호스트를 나타내는 메트릭들이다. 먼저 이 호스트의 TCP 플로우 집합으로부터 플로우 클러스터링을 수행한다. 이 때 총 m개의 클러스터가 생성되면 각 클러스터로부터 APS를 계산한다. APS를 포함한 표 4의 메트릭들은 하나의 클러스터를 나타내는 메트릭들이다. 이렇게 계산한 값들을 APS_1, ..., APS_m이라고 했을 때 M_{T1}은 $\max(\{APS_i | 1 \le i \le m\})$가 된다. 같은 방식으로 나머지 메트릭 M_{T2}~M_{U5}도 계산한다. 이로부터 (M_{T1}, ..., M_{T4})의 TCP 포인트와 (M_{U1}, ..., M_{U5})의 UDP 포인트를 생성한다.

앞에서 분석한 바에 따르면 TCP 봇넷의 포인트는 M_{T1}~M_{T4}가 모두 높을 것이고 UDP 봇넷의 포인트는 M_{U1}~M_{U4}가 높고 M_{U5}가 낮을 것이다.

ISOT Botnet Dataset으로부터 트래픽 데이터를 플로우로 변환한 뒤 호스트별로 위의 메트릭을 사용하여 4차원의 TCP 포인트셋과 5차원의 UDP 포인트셋을 생성하였다. 그 후에 각 포인트셋에 대하여 X-means 클러스터링을 수행하였다.

5.3 P2P 봇넷 트래픽 필터링

P2P 봇넷 탐지 실험에서 효율성을 위하여 추가적으로 필터링을 하였다. 우선 호스트의 경우 성공한 플로우가 없거나 실패한 플로우가 10개 이하이면 P2P 통신을 하지 않는다고 판단하여 해당 호스트를 제외했다. 또한 플로우 클러스터링의 결과로부터 클러스터 내의 Destination IP의 BGP prefix 개수를 조사하여 50개 이하이면 해당 클러스터를 제외했다. 이 과정에서 총 20,120개

의 호스트 중 2,309개의 호스트만 남았다. 결과적으로 필터링을 통하여 대략 90%의 포인트를 줄일 수 있었다.

6. 실험 결과

이 장에서는 각 데이터에 대하여 최종적으로 클러스터링한 결과를 분석하여 악성 트래픽을 탐지하고 검증한다.

6.1 디도스 공격 탐지 결과

그림 2는 Darpa dataset으로부터 변환한 포인트셋의 클러스터링 결과를 나타낸다. x, y, z축은 각각 M_1, M_4, M_6를 의미한다.

X-means 클러스터링 수행 결과 총 4개의 클러스터로 나뉘었다. 표 6은 각 클러스터의 평균점을 나타낸다. 4개의 클러스터 중 M_1, M_2의 값이 큰 cluster 2를 공격 1단계로, M_3, M_4의 값이 큰 cluster 1을 2단계로, M_5, M_6의 값이 큰 cluster 3를 5단계로 분류하였다. 그리고 남은 cluster 4를 정상으로 분류하였다.

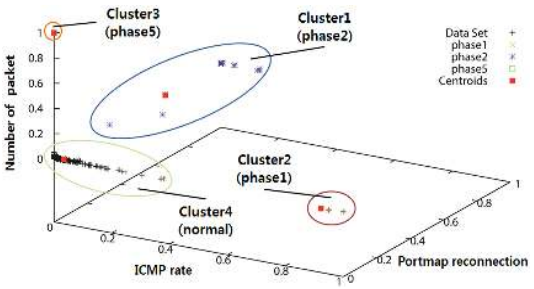

그림 2 클러스터링 결과
Fig. 2 Clustering result

표 6 각 클러스터의 평균점
Table 6 Centroid for each cluster

	cluster 1	cluster 2	cluster 3	cluster 4
M_1	0	0.923	0	0.034
M_2	0.009	0.729	0.005	0.009
M_3	0.5	0	0	0
M_4	0.667	0	0	0
M_5	0.164	0.441	1	0.193
M_6	0.004	0.007	1	0.012

표 7 각 클러스터의 레이블링 분포
Table 7 Distribution of labelings in each cluster

	cluster 1	cluster 2	cluster 3	cluster 4
phase 1	0	3	0	1
phase 2	62	0	0	0
phase 5	0	0	1	0
normal	0	0	0	1104
size	62	3	1	1105

표 7은 위의 탐지 결과를 검증하기 위해 실제 cluster마다 포인트들의 레이블링의 분포를 나타낸 표이다. 이로부터 클러스터링이 포인트들을 공격 단계별로 거의 완벽하게 구분한다는 것을 확인할 수 있다.

6.2 P2P 봇넷 탐지 결과

그림 3과 그림 4는 각각 ISOT dataset으로부터 변환한 TCP, UDP 포인트셋의 클러스터링 결과를 나타낸다. 결과는 PCA(Principle Component Analysis)[19]를 사용하여 3차원 좌표로 변환하여 도시하였다.

그림 3을 보면 X-means 클러스터링 수행 결과 2개의 클러스터로 나뉘었다. 2개의 클러스터 중 모든 좌표 값이 큰 클러스터를 봇으로 탐지했다. 해당 클러스터에

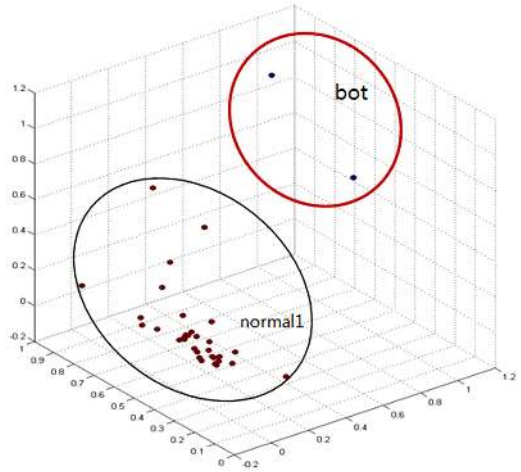

그림 3 호스트 클러스터링 결과 (TCP)
Fig. 3 Host clustering result (TCP)

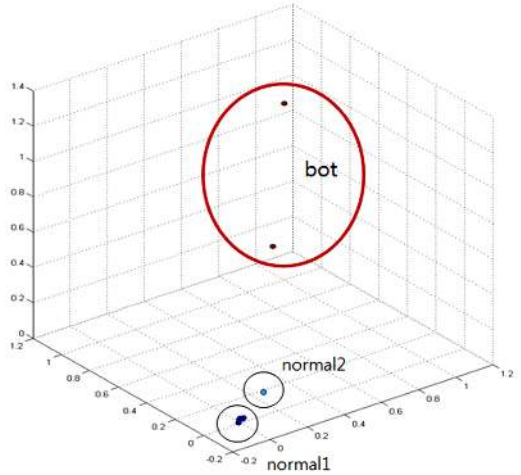

그림 4 호스트 클러스터링 결과 (UDP)
Fig. 4 Host clustering result (UDP)

표 8 각 클러스터의 평균점 (TCP)
Table 8 Centroid for each cluster (TCP)

	M_{T1}	M_{T2}	M_{T3}	M_{T4}
normal	0.038	0.422	0.139	0.026
botnet	0.766	0.525	0.966	0.963

표 9 각 클러스터의 평균점 (UDP)
Table 9 Centroid for each cluster (UDP)

	M_{U1}	M_{U2}	M_{U3}	M_{U4}	M_{U5}
normal 1	0.00	0.003	0.019	0.024	0.096
normal 2	0.14	0.022	0.114	0.011	1.000
botnet	0.71	0.667	0.849	0.853	0.000

속한 두 호스트는 172.16.0.11, 172.16.0.12로 각각 Waledac과 Storm의 스팸메일을 전송하는 봇이었다. 표 8은 각 클러스터의 평균점을 나타낸다.

그림 4를 보면 X-means 클러스터링 수행 결과 3개의 클러스터로 나뉘었다. 3개의 클러스터 중 M_{T1}, M_{T2}, M_{T3}, M_{T4}가 높고 M_{T5}가 낮은 클러스터를 봇으로 탐지했다. 해당 클러스터에 속한 두 호스트는 172.16.2.11, 172.16.0.11로 각각 Storm 봇과 Waledac 봇이었다. 표 9는 각 클러스터의 평균점을 나타낸다.

6.3 기존 연구와의 비교

K. Lee 등은 계층적 응집 클러스터링을 통하여 Darpa 2000 Dataset의 DDoS 공격을 모두 탐지하였다[7]. 계층적 클러스터링은 일반적으로 계산복잡도가 크고 클러스터링이 끝난 뒤 클러스터의 수를 결정해야 하는 문제가 있다. 반면 본 논문에서 악성 트래픽 탐지에 사용한 X-means 알고리즘은 클러스터링 중에 최적 클러스터 수를 결정하고 멈추기 때문에 큰 데이터에 대하여 더욱 효율적이다.

S. Saad 등은 ISOT Botnet Dataset에서 기계학습 기법을 이용하여 공격 호스트를 탐지하였다[8]. 이 방법은 학습 과정이 필요하기 때문에 사전에 공격이 들어있는 트래픽 데이터가 존재해야 한다. 반면 본 논문의 기법은 공격의 특성을 통하여 메트릭을 정의하고 공격 데이터의 메트릭 값들을 미리 추정하기 때문에 바로 실제 트래픽 데이터에 적용하여 공격을 탐지할 수 있다.

H. Choi 등은 X-means 클러스터링을 이용하여 봇넷을 탐지하였지만[9] 이들은 봇넷의 C&C 서버와 봇 간의 DNS 트래픽의 그룹 행동을 가정하였기 때문에 DNS 트래픽을 사용하지 않는 P2P 봇넷 탐지에 취약하고 공격자가 그룹 행동이 나타나지 않도록 DNS 트래픽을 정교하게 랜덤화 하는 등의 역공격을 할 수 있다.

7. 결 론

이 논문에서는 알려진 악성 트래픽의 통계적 특징을 분석하고 메트릭을 정의한 뒤 클러스터링을 통하여 악성 트래픽을 탐지하는 방법을 제시하였다. 또한 두 가지 공개된 데이터에 이 방법을 적용하여 악성 트래픽을 정확하게 탐지하였다. 본 논문에서 제시한 기법은 공격에 대한 사전 분석이 필요하기 때문에 새로운 공격에 대응이 늦을 수 있다. 하지만 대부분의 공격이 반복적으로 이루어지고 또한 대부분의 공격들의 내재된 특성이 서로 유사하므로 조금 변형된 공격에 대하여도 기존 공격들을 데이터베이스화 해 놓으면 대부분의 공격에 대응할 수 있을 것으로 예상된다. 이러한 점에 미루어 향후 전체 악성 트래픽을 포괄하면서 새로운 공격에도 대응할 수 있는 일반적인 메트릭에 대한 연구를 진행할 필요가 있다.

References

[1] Y. Xie, S. Z. Yu, "Monitoring the application-layer DDoS attacks for popular websites," *IEEE/ACM Transactions on Networking*, Vol. 17, No. 1, pp. 15-25, 2009.

[2] G. Gu, R. Perdisci, J. Zhang, W. Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection," *SS'08 Proceedings of the 17th conference on Security symposium*, pp. 139-154, 2008.

[3] J. Zhang, R. Perdisci, W. Lee, U. Sarfraz, X. Luo, "Detecting stealthy P2P botnets using statistical traffic fingerprints," *Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*, pp. 121-132, 2011.

[4] H. R. Zeidanloo, M. J. Z. Shooshtari, P. V. Amoli, M. Safari, M. Zamani, "A taxonomy of Botnet detection techniques," *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, Vol. 2, pp. 158-162, 2010.

[5] T. Thapngam, S. Yu, W. Zhou, G. Beliakov, "Discriminating DDoS attack traffic from flash crowd through packet arrival patterns," *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pp. 952-957, 2011.

[6] J. Mirkovic, P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 2, pp. 39-53, 2004.

[7] K. Lee, J. Kim, K. H. Kwon, Y. Han, S. Kim, "DDoS attack detection method using cluster analysis," *Expert Systems with Applications*, Vol. 34, No. 3, pp. 1659-1665, 2008.

[8] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, L. Wei, J. Felix, P. Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*, pp.174-

180, 2011.
[9] H. Choi, H. Lee, "Identifying botnets by capturing group activities in DNS traffic," *Computer Networks*, Vol. 56, No. 1, pp. 20-33, 2012.
[10] S. Stover, D. Dittrich, J. Hernandez, S. Dietrich, "Analysis of the Storm and Nugache trojans: P2P is here," *;login:*, Vol. 32, No. 6, pp. 18-27, 2007.
[11] G. Sinclair, C. Nunnery, B. H. Kang, "The waledac protocol: The how and why," *Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on*, pp. 69-77, 2009.
[12] J. A. Hartigan, M. A. Wong, "A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, Vol. 28, pp. 100-108, 1979.
[13] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, Vol. 32, No. 3, pp. 241-254, 1967.
[14] G. W. Milligan, M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, Vol. 50, No. 2, pp. 159-179, 1985.
[15] D. Pelleg, A. Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," In *Proceedings of the 17th International Conf. on Machine Learning*, pp. 727-734, 2000.
[16] DARPA Dataset [Online]. Available: http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/2000data.html
[17] ISOT Dataset [Online]. Available: http://www.uvic.ca/engineering/ece/isot/datasets/index.php
[18] T. Karagiannis, A. Broido, M. Faloutsos, K. claffy, "Transport layer identification of P2P traffic," *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 121-134, 2004.

한 명 지

2010년 서울대학교 컴퓨터공학부(학사) 2010년~현재 서울대학교 컴퓨터공학부(석박사 통합과정). 관심분야는 알고리즘, 빅데이터

임 지 혁

2006년~2010년 한양대학교 전자통신컴퓨터공학부(학사). 2010년~2012년 한양대학교 전자컴퓨터통신공학과(석사). 2013년~현재 서울대학교 컴퓨터공학부(박사과정). 관심분야는 알고리즘, 보안, GPGPU

최 준 용

2013년 중앙대학교 컴퓨터공학부(학사) 2013년~현재 서울대학교 컴퓨터공학부(석박사 통합과정). 관심분야는 알고리즘, 빅데이터

김 현 준

2013년 서울대학교 전기컴퓨터공학부(학사). 2013년~현재 서울대학교 전기·컴퓨터공학부(석박사 통합과정). 관심분야는 빅데이터, 바이오인포매틱스

서 정 주

2009년 성균관대학교 컴퓨터공학과(학사) 2009년~현재 서울대학교 전기컴퓨터공학부(석박사 통합과정). 관심분야는 알고리즘, 컴퓨터이론, 암호 및 보안

유 철

2007년~2012년 연세대학교 전기전자공학부(학사). 2012년~현재 서울대학교 전기·컴퓨터공학부(석박사 통합과정). 관심분야는 스트링 매칭

김 성 렬

1993년 서울대학교 컴퓨터공학과(학사) 1995년 서울대학교 컴퓨터공학과(석사) 2000년 서울대학교 컴퓨터공학과(박사) 2000년~2002년 WiseNut Inc. Engineering Manager. 2002년~현재 건국대학교 인터넷미디어공학부 교수. 관심분야는 암호학 및 시스템 보안, 병렬 알고리즘, 계산 복잡도, 정보 검색

박 근 수

1983년 서울대학교 컴퓨터공학과 학사 1985년 서울대학교 컴퓨터공학과 석사 1992년 미국 Columbia 대학교 전산학박사. 1991년 11월~1993년 8월 영국 런던대학교 King's College 조교수. 1993년 8월~현재 서울대학교 컴퓨터공학부 교수. 관심분야는 컴퓨터이론, 생물정보학, 암호학, 웹 검색