情報保護學會論文誌
第 22 卷 第 3 號, 2012. 6

반가상화 환경 Guest OS 보호를 위한 효율적인 서비스 거부 공격 탐지 방법에 관한 연구

신 승 훈,† 정 만 현, 문 종 섭‡
고려대학교 정보보호대학원

A study on the effective method of detecting denial of service attack to protect Guest OS in paravirtualization

Seunghun Shin,† Manhyun Jung, Jongsub Moon‡
Graduate School of Information Security, Korea University

요 약

최근에 자원의 효율적 사용과 비용 절감을 위해 클라우드 컴퓨팅 서비스가 크게 이슈화 되고 있다. 하지만, 클라우드 컴퓨팅 서비스에 대한 보안 안전성이 제대로 검증이 되지 않아 대중적으로 사용하기엔 한계가 있다. 특히, Guest OS에 대해 보안 취약점이 그대로 드러나 있어 좀비 PC로 활용되어 서비스 거부 공격을 유발시킬 가능성이 점차 증대되고 있다. 본 논문에서는 Xen으로 구현된 클라우드 시스템에서 Guest OS 취약점으로 인해 좀비PC로 사용되어 발생될 수 있는 내부 서비스 거부 공격에 대해 Xen에서 발생되는 하이퍼콜 빈도수를 이용한 침입 탐지 방안에 대해 제안 한다. 실험을 통해서 K-means와 EM을 사용하여 제안된 방법이 2분, 5분, 10분, 30분 동안 수집한 두 데이터가 2분, 5분일 때 90%이상 10분 이상일 때 100% 분류율을 보이며 성공적으로 분류가 가능함을 보였다.

ABSTRACT

Recently, cloud computing service has become a rising issue in terms of utilizing sources more efficiently and saving costs. However, the service still has some limitations to be popularized because it lacks the verification towards security safety. In particular, the possibility to induce Denial of service is increasing as it is used as Zombie PC with exposure to security weakness of Guest OS's. This paper suggests how cloud system, which is implemented by Xen, detects intrusion caused by Denial of service using hypercall. Through the experiment, the method suggested by K-means and EM shows that two data ,collected for 2 mins, 5 mins, 10mins and 20mins each, are distinguished 90% when collected for 2mins and 5mins while collected over 10mins are distinguished 100% successfully.

Keywords: Cloud Computing, Intrusion detection, Virtual Machine, Xen, K-Means, EM

I. 서 론

클라우드 컴퓨팅은 서로 다른 물리적 위치에 존재하는 다양한 종류의 컴퓨팅 및 스토리지를 통합하여 가상화된 고성능 컴퓨팅 자원 집합체를 구축하고 다수의 고객들에게 높은 수준의 확장성을 가진 IT자원들을 온-디멘드(on-demand)방식으로 제공하여 자원

접수일(20012년 3월 23일), 수정일(2012년 4월 25일), 게재확정일(2012년 4월 26일)
† 주저자, shin0623@korea.ac.kr
‡ 교신저자, jsmoon@korea.ac.kr

효율성 극대화와 관리의 최소화라는 장점을 가지는 새로운 컴퓨팅 패러다임이다[1]. 이러한 클라우드 컴퓨팅 서비스는 물리적 자원을 적재적소에 필요한 사람에게 원하는 만큼 제공하고 또 반환하는 비용 효율적인 서비스로서 크게 각광받고 있다. 하지만, 클라우드 컴퓨팅 서비스 이면에 많은 보안 문제점을 내포하고 있어 보안 취약점에 대해 제대로 분석되지 않은 상태에서 서비스의 진행은 사용자들에게 큰 위협요인[2]이 되고 있다. 특히 클라우드 서비스는 언제 어디든 원하는 서비스를 제공받아야 하는 가용성 측면이 가장 중요 요소로 꼽히고 있다. 하지만, 사용자들 내부에서 발생되는 문제점에 대해 자원의 모니터링에 국한되어 있어 위협을 사전에 효과적으로 대처할 수 없는 문제점이 발생한다. 특히 최근에 큰 이슈화 되었던 DDoS 공격은 유휴 자원을 고갈시켜 서비스를 마비시키는 시스템에 심각한 악영향을 가져온다. 클라우드 환경 또한 사용자에게 제공한 Guest OS에 대한 보호가 되지 않은 상태에서 언제든지 공격자에 의해서 좀비 PC로 활용 될 수 있으며, 이는 또한 내부의 DDoS공격을 위협하는 요인으로 작용될 수 있다[3]. 본 논문에서는 오픈소스 클라우드 컴퓨팅을 지원하는 가상머신 Xen 환경을 구축하고, 웹서버를 서비스하고 있는 Guest OS와 웹서버를 공격하는 Guest OS를 구축하고 실제 내부에서도 DDoS공격을 통해 가용하고 있는 서비스가 다운됨을 보이며, 이를 탐지하기 위한 탐지 방안을 제안 한다. 본 논문에서는 서비스거부 공격에 대해 사전에 방지할 수 있도록, 하이퍼바이저의 하이퍼 콜 이벤트를 통계적으로 분류하여 검증하는 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 클라우드 가상화 기법와 하이퍼 콜 이벤트 그리고 침입탐지 방법에 대해 설명한다. 3장에서는 본 논문에서 제안하는 방법을 설명한다. 4장에서는 실험을 통해 제안된 방법의 결과를 분석해 보이고, 마지막 5장에서는 결론을 맺으며 본 연구를 토대로 앞으로 더욱 연구하고 개선할 내용을 기술한다.

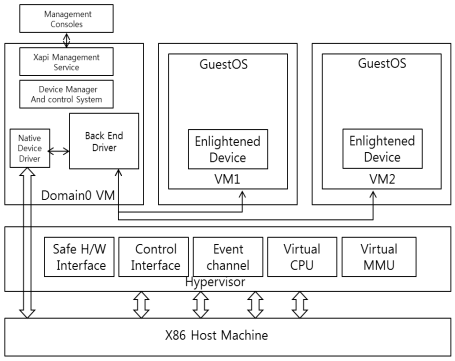
II. 관련연구

2.1 클라우드 컴퓨팅

클라우드 컴퓨팅을 구성하는 물리적인 컴퓨팅 자원들에 가상화 기술을 적용하여 다수의 사용자에게 서버, 스토리지, 네트워크 등의 자원을 제공하기 위한 가상화 기술은 여러 종류의 운영체제를 사용할 수 있게 하기 위하여 각 운영체제가 하드웨어를 공유할 수 있도록 기능을 제공하는 소프트웨어 기술로 [그림 1]과 같이 다수의 서비스 플랫폼을 하나의 플랫폼으로 통합하는 것을 가능하게 한다[4]. 기존 데이터 센터 환경에서는 개별적으로 운영되는 서버들의 컴퓨팅 자원 활용률이 하드웨어 성능에 비해 크게 떨어져 자원 낭비가 발생하였다. 그러나 서버 통합 환경에서 공간 및 컴퓨팅 자원의 낭비를 방지할 수 있는 가상화 기술은 데이터 센터 환경에서 주목을 받고 있다. 가상화 기술은 현재 여러 벤더에서 연구 및 상품화를 진행하고 있다. 그 중 시스템 가상화 도구로 보편적으로 사용되는 것으로는 VMware와 Xen을 들 수 있으며, 가상화의 형태에 따라 전가상화(Full Virtualization)와반가상화(Para Virtualization)로 나뉜다.

2.1.1 반가상화 기법

반가상화는 운영체제를 가상화 지원이 가능할 수 있도록 수정하여 시스템 부팅 시 적용하는 방법으로서 CPU와 메모리 등 일부 장치만을 가상화하며, Xen의 가상화 방법이 이에 속한다. 본 논문에서 클라우드 시스템 구축을 통한 물리적인 자원의 가상화를 위하여 사용한 가상화 도구는 Xen이며, Xen의 구조는 VMM(Virtual Machine Monitor)으로도 불리는 하이퍼바이저(Hypervisor)와 Host OS(Dom-0), Guest OS(Dom-U)로 구성된다. 하이퍼바이저는 하드웨어 상위에서 CPU와 메모리의 자원을 Guest OS에 할당해 주는 역할을 한다. Host OS는 Dom-0로 불리며, CPU와 메모리 외에 Guest OS가 디바이스를 접근하기 위해서는 이를 거쳐야 한다. 또한 시스템 부팅 시 먼저 시작하여 게스트 도메인의 생성 및 제거 등 관리 작업을 수행하며, Guest OS는 가상머신을 의미한다. Xen Linux는 Linux를 수정한 버전으로, 다른 Guest OS를 변경시키거나 시스템 권한을 가질 수 있는 Privileged level을 가진 명령어들을 하이퍼 콜 interface로 대치시킨 후 빌드된다. 커널 일부가 수정이 되며 User level Linux legacy 응용프로그램은 수정 없이 동작이 된다. 수행 중 Privileged level을 가진 명령어들의 호출이 발생하면 Guest OS에서 수행되지 않고 미리 정의된 하이퍼 콜을 통해 하이퍼바이저에서 수행이 이루어진다. 디바이스 드라이버는 Front-end Device driver를 가지게 되는데 이는 Dom-0 측 Back-end



〔그림 1〕 Xen 가상화 아키텍쳐

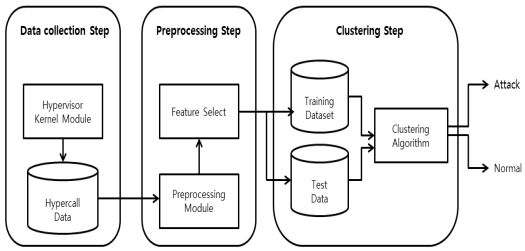
Device driver 와 통신을 하여 실제 디바이스 제어는 Dom-0에서 이루어지게 된다. 〔그림 1〕는 Xen의 구조를 나타낸다[5].

2.1.2 하이퍼콜과 이벤트

Xen 과 Guest OS 도메인 간의 상호 작용을 제어하기 위해 두 가지 메커니즘이 존재한다[6]. Xen으로부터 Guest OS로의 통신은 비동기 이벤트 메커니즘을 사용하며, Guest OS로부터 Xen에서의 통신은 하이퍼콜을 사용하여 동기식 호출을 수행한다. 하이퍼콜 인터페이스는 특권적인 활동을 실행하기 위해서 하이퍼바이저에 동기적인 소프트웨어 트랩을 실행시키기 위한 도메인을 수행한다. Guest OS를 대신해서 특수 권한 명령을 수행 하는 하이퍼콜은 실제 OS의 시스템 콜과 비슷하지만 반가상화의 Xen에서 발생되는 이벤트 함수이다[7]. 하이퍼콜은 새로운 도메인의 생성, 주소 매핑 생성, 도메인 간 버퍼 전송, CPU 인터럽트 전송 등의 기능들을 가지고 Guest OS에서의 특수 권한 명령에 대한 호출에 대해서 해당 API를 처리하여 준다. Ring 3에 Guest 응용 프로그램이 몇 가지 프로세스를 실행시키고 Ring 0의 호스트 커널의 권한을 필요로 하는 하이퍼콜 요청을 들어, Dom-0와 하이퍼바이저를 통해 라우팅 된다. 하이퍼바이저는 요청된 콜 들의 진위를 확인하고 실행을 가능하게 한다. Xen으로부터의 Guest OS의 통신은 비동기 이벤트 메커니즘을 통해 제공되는데, 이는 디바이스 인터럽트를 위해서 전달메커니즘을 치환하고, 도메인-종료 요청과 같이 중요한 이벤트의 신속한 통지를 수락한다. 전통적인 Unix 시그널과 같이, 소수의 이벤트만이 있으며, 각각의 신호마다 한 종류의 특정사건을 수행한다. 이벤트는 새로운 데이터가 네트워크를 통해 수신된 것이나 가상 디스크 요청이 완료된 것을 나타내 보이는데 이용된다.

2.2 침입 탐지 연구

침입 탐지 시스템은 네트워크 기반 침입탐지 시스템과 호스트 기반 침입탐지 시스템 두 가지로 분류 된다. 네트워크 기반의 침입 탐지 시스템은 네트워크 패킷을 모니터링하거나 트래픽의 변화량을 분석 하는 방향으로 연구가 진행되고 있으며 호스트 기반 침입 탐지 시스템은 시스템에서 발생 하는 커널 기반 데이터나 시스템 로그 기록시스템의 메모리, cpu등의 사용내역을 이용한 연구들이 진행되고 있다. 특히 커널 기반 데이터를 이용한 연구가 활발히 진행되고 있다. 커널 기반 데이터의 경우 시스템에서 발생하는 행위를 표현할 수 있는 데이터를 말하는데 대표적으로 사용자와 프로세스들의 행위를 나타내는 시스템 호출을 많이 이용한다. 시스템 호출을 이용한 침입탐지 시스템의 대표적 연구로 Liao and Vemuri[8], Wenjie Hu and Liao and Vemuri[9], Seung-Hyun Paek[10]의 연구가 있다. Liao and Vemuri는 MIT에서 제공하는 1998 DARPA Basic Security Module(BSM)[11]데이터의 일반 행위 데이터와 공격 행위 데이터에서 시스템 호출을 프로세스 별로 추출하여 시스템 호출빈도를 계산하고 가장 많은 빈도를 가진 시스템 호출 49개를 추출하고 나머지를 기타로 하여 총 50개의 시스템 호출을 추출하는 방법으로 전처리를 하고 각 프로세스 별 시스템 호출의 빈도를 이용하여 정보 검색 분야에서 많이 사용되는 TF-IDF 가중치 방법을 적용하여 시스템 호출의 값을 결정하였다. TF-IDF는 문서에서 각 단어의 가중치를 해당 문서에서 각 단어의 빈도와 역 문헌빈도(IDF)의 곱으로 나타내는 방식이다. 문서를 프로세스로 문서안의 단어를 시스템 호출로 하여 계산된 결과 값을 K-Nearest neighbor 알고리즘에 적용하였고 공격과 일반 행위의 거리 값을 구하기 위하여 Cosine Similarity를 사용하여 침입 탐지를 하는 방법을 제안하였다. 하지만, 다수의 Guest OS가 서비스 되는 클라우드 시스템에서 시스템 호출 값을 통한 DDoS탐지시스템은 시스템에 많은 Overhead를 발생시키고, 실제 메모리 점유율도 높아 하드웨어 Resource도 많이 차지하게 되어 효과적이지 않다.



〔그림 2〕 서비스 거부 공격 분류 시스템 구조도

III. 제안하는 기법

전체적인 구성 방법은 첫 번째 하이퍼바이저의 커널에서 하이퍼콜 데이터를 수집하고, 두 번째 수집된 데이터를 전처리 후 하이퍼콜의 feature를 선정 한 다음 세 번째 공격데이터와 일반데이터를 모델링하고 분류하는 군집화 과정으로 크게 세부분으로 구성 되며 마지막으로 군집의 유효성 검증을 통해 군집의 효과성을 검증한다. 구성은 다음 〔그림 2〕와 같다.

3.1 하이퍼콜 수집 과정

하이퍼콜은 Linux 2.6.22 커널에서 324개의 시스템 콜과 비교할 때 오직 35개의 하이퍼콜을 가지고 있다〔12〕. 본 논문에서는 35개의 하이퍼콜 중 웹서버로 서비스를 요청할 때 발생되는 I/O연산과 연관 있는 event_channel, grand-table 그리고 메모리, cpu 사용과 관련 있는 mmuext_op, hvm_op와 vcpu_op 등을 5개를 특징으로 추출하여 하이퍼콜을 수집하였다. 각 하이퍼콜의 기능 및 특징은 〔표 1〕과 같다. 각각의 특징 벡터를 다음 식(1) 와 같이 표현한다.

$$CV_F = \{V_{F,1}, V_{F,2}, V_{F,3}, V_{F,4}, V_{F,5}\} \quad (1)$$

〔표 1〕각 하이퍼콜의 기능 및 특징

특징벡터	하이퍼콜	기능
$V_{F,1}$	event_ channel_op	내부 도메인 간 이벤트 요청 수
$V_{F,2}$	grand_ table	메모리 writable page 요청 수
$V_{F,3}$	mmuext_op	확장된 메모리 연산 수행 수
$V_{F,4}$	hvm_op	hardware virtual machine I/O요청 수
$V_{F,5}$	vcpu_op	GuestOS cpu 동작 발생 콜 수

수집 방법은 Xenperf〔13〕라는 Xen 하이퍼바이저에서 발생되는 하이퍼콜 성능을 수집 해주는 시스템 툴을 사용하여 수집 하였다. 수집된 대용량의 하이퍼콜 데이터를 추출한 특징들을 사용하여 일반 행위와 공격 행위를 기본적인 통계 값을 갖기 위해 하이퍼콜 호출이 발생하는 시간 t로 하고 단위를 초 단위로 하여 시간당 반복적으로 수집하고 계산하여 통계적 수치를 구한다.

3.2 데이터 전처리

이렇게 수집된 5개 하이퍼콜 Feature 값은 군집을 위한 입력 값으로 사용된다. Feature Scaling은 Feature space가 굉장히 크고, Feature Value가 〔1,500000〕라서 다량의 데이터를 군집화 하는 알고리즘에는 적합하지 않아 〔0,1〕사이로 정규화 했다. Scaling 은 다음 식(2)와 같이 정의된다.

$$Normalized(e_i) = \frac{e_i - E_{\min}}{E_{\max} - E_{\min}} \quad (2)$$

3.3 데이터 분류

하이퍼콜 호출 수 측정을 통해 구한 값을 반복적인 모델 정제 과정을 통해 최적의 군집을 찾는 알고리즘으로 효과적인 K-means와 EM 알고리즘을 이용하여 분류하였다. K-means 알고리즘이 데이터간의 거리를 계산할 때 유클리디언(Euclidean) 거리 계산 방법을 사용하는 것과는 달리 EM은 통계적인 방법을 사용하므로 EM알고리즘이 더욱 효과적인 기능을 제공하여 많이 사용된다.

3.3.1 K-means 알고리즘

K-means은 주어진 데이터 집합을 데이터가 가진 속성의 유사도에 따라서 K개의 클러스터로 묶는 알고리즘으로서, 각 클러스터와 거리차이의 분산을 최소화하는 방식으로 동작한다〔14〕. K-means 알고리즘〔표 2〕은 입력 값으로 K를 취하고 군집 내 유사성은 높고 군집끼리의 유사성은 낮도록 n개 객체들의 집합을 k개의 군집으로 분할한다. 이 기법은 전체 데이터가 n이고, k가 군집의 수, t가 반복수일 때 알고리즘의 복잡도가 O(nkt)이기 때문에 큰 데이터 집합을 다루는데 상대적으로 효율적이고 확장적용이 가능하다.

〔표 2〕 K-means 알고리즘

Algorithm KMEANS(dataset X, initial centers C_{init})
X : a set of N data points C_{init} : initial centers of k clusters C : cluster centers of k clusters P = {p(i) | I=1,....,N} is the cluster label of X 1: C $\leftarrow$ C_{init} 2: while(C$\neq C_{prev}$)do { 3: $C_{prev} \leftarrow$C 4: for each centers x_i in X do 5: p(i)$\leftarrow$ $\arg min_{c \in C} d(x_i, c)$ 6: for each centers C_j C do 7: $C_j \leftarrow$Average of x_i , whose p(i) = j 8: } 9: return (C,P)

3.3.2 EM 알고리즘

EM은 데이터를 그룹으로 분리하는데, 어느 그룹에 속할지 모를 때, 전체 확률을 최대한 한다. 이는 정보가 직접적으로 얻어지지 않고, 다른 관측 가능한 변수를 통하여 획득할 수 있는 경우이므로, 관심의 대상이 되는 정보를 관측 가능한 변수의 공간을 통하여 추정하는 통계적 방법이다〔15〕.

EM(Expectation-maximization)알고리즘〔표 3〕 수행 전에 군집의 개수, k와 종료 조건으로 $\varepsilon(>0)$ 을 입력받는다. EM에서의 모델은 정규분포를 가정하기 때문에 이 과정은 새로 배정된 레코드들의 평균과 표준편차를 변경한다.

〔표 3〕 EM 알고리즘

Algorithm EM
Input : Cluster number k, a database, Stopping tolerance $\varepsilon(>0)$ Output : A set of k clusters with weight that maximize Log-likelihood function 1: Expectation Step For each database record x, Compute the membership probability of x in each cluster h=1,...,k 2: Maximization Step Update mixture model parameter(probability weight) 3: Stopping criteria If stop criteria is satisfied stop Else set j = j+1 and goto 1

3.4 군집화 유효성 검증

위 두 알고리즘은 클러스터의 개수를 지정하고 수행되게 되므로 지정한 클러스터의 개수가 데이터에 적합한지를 검증해야 한다. 가장 많이 사용하는 검증 방법으로 Partition Coefficient(PC)와 Partition Entropy coefficient(PE)가 있다〔16〕. PC는 클러스터 내의 데이터의 상관관계를 보는 것으로서 얼마나 조밀하게 모여 있는지를 나타낸다. PE는 PC와 유사하나 클러터 내의 데이터가 얼마나 확산되어있는지를 나타낸다. PC는 1에 가까울수록 PE는 0에 가까울수록 분류가 잘 되었음을 증명한다. 식(3)(4)는 각각의 검증방법에 수행되는 수식이다.

$$PC(c) = \frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^2 \quad (3)$$

$$PE(c) = -\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}u_{ij}\log(u_{ij}) \quad (4)$$

IV. 실험 및 결과

4.1 표본조사

4.1.1 실험환경

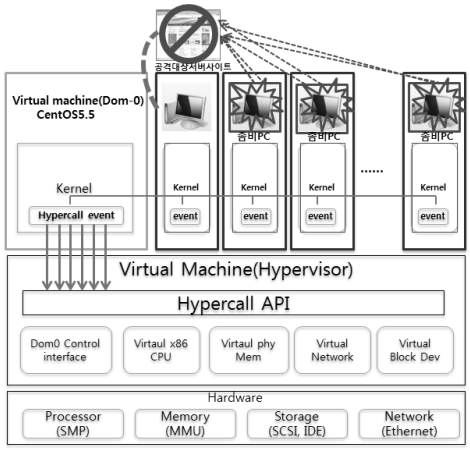
실험에 사용된 장비는 다음 〔표 4〕와 같다. Xen 가상머신은 1대의 서버에 구축하였으며, 각각 실제 서비스를 위한 Guest OS 1대와 공격을 감행하는 공격 PC용 Guest OS 5대를 구축하여 총 6대의 Guest OS를 생성하였다. 본 실험은 내부의 Guest OS의 좀비로 인한 실제 서비스 중인 웹서버를 공격하기 위해 내부 공격으로 실험을 제한한다.

〔표 4〕 실험장비

분류	장비	세부정보	갯수
웹서버	intel L5640	CentOS5.5	1
좀비Guest OS	intel L5640	windowXP	2
일반PC	intel L5640	CentOS5.5	3

4.1.2 실험 내용

일반적으로 사용되는 1대의 Xen 가상머신은 총 6대의 서비스가 제공되고 있다. 〔그림 3〕과 같이 1대의 웹서버에서 정상적인 서비스를 제공되고 있으며, 5대



〔그림 3〕 Xen cloud 실험 환경

는 일반적인 사용자에 의해 사용되고, 이 사용되는 5대 PC에서 DDoS공격 툴을 사용하여 웹서버로 공격을 수행하였다. 일반 행위를 수집하기 위해서 본 웹서비스를 일반적으로 운영되는 홈페이지를 사용하여 데이터를 하루 동안 수집하였고 각 웹서비스는 일반적인 포털접속, 게시판, 자료 업로드 등의 행위를 수행시키고 정상행위를 수집하였다. 공격 행위는 실제 7.7 DDoS공격이 발생 했던 DDoS공격 툴을 사용하여 5대를 좀비 PC로 구성한 후 웹서버를 공격을 감행하였다. TCP, UDP, HTTP Get flooding 공격을 감행하여 시스템이 마비되는 상황까지의 데이터를 초당 2분, 5분, 10분, 30분 단위로 하루에 걸쳐 수집하였다.

4.2 결과 및 분석

본 실험에서는 각 초당 수집된 데이터 에서 발생한 하이퍼콜의 호출 빈도를 계산하여 각각 scaling을 통해 데이터 값을〔0,1〕로 변환한 후 비교 한다. 〔표 5〕은 스케일링으로 변환한 값이다. 다음〔표 5〕같이 계산된 데이터를 이용하여 기대도수를 구한다. 분류 성능을 측정하기 위하여 전체 데이터 중 80%를 학습데이터로 정하고 20%를 실험데이터로 분류

〔표 5〕 하이퍼콜 scaling data

t	$V_{F,1}$	$V_{F,2}$	$V_{F,3}$	$V_{F,4}$	$V_{F,5}$
1	0.0038	0.1432	0.0441	0.0388	0.0177
2	0.0049	0.1321	0.0545	0.0252	0.0125
3	0.0049	0.1479	0.0529	0.0222	0.0140
4	0.0106	0.1658	0.0577	0.0286	0.0110
…	…	…	…	…	…

〔표 6〕 Feature 에 따른 실험 망에서의 데이터 결과

V_F	Class 1(Normal)		Class 2(Attack)	
	평균	표준편차	평균	표준편차
$V_{F,1}$	0.009	0.0037	0.487	0.1126
$V_{F,2}$	0.194	0.1121	0.454	0.2870
$V_{F,3}$	0.048	0.0197	0.259	0.1420
$V_{F,4}$	0.013	0.0107	0.153	0.1678
$V_{F,5}$	0.014	0.0037	0.209	0.1544

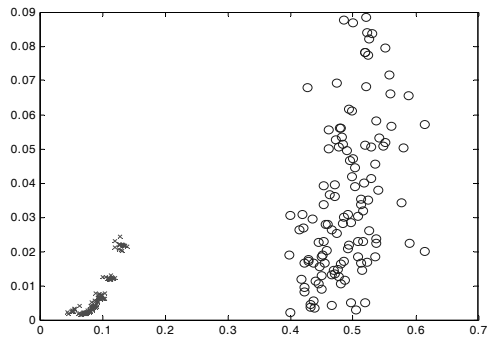
〔표 7〕 각 알고리즘 시간에 따른 분류율

t	K-means	EM
120	98.2%	99.1%
300	99.4%	99.89%
600	100%	100%
1800	100%	100%

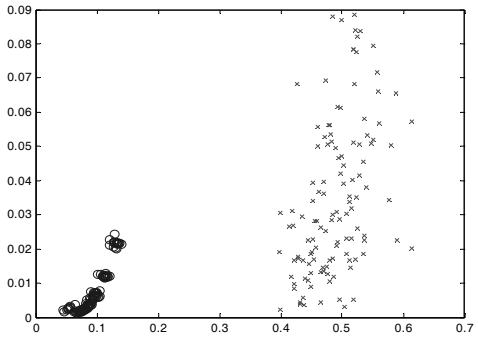
를 수행하였다.

또한 K-means 알고리즘에서 사용하는 K값은 분류하고자 하는 집단이 일반적인 상황과 공격적인 상황인 두 집단이므로 2로 정하고 군집화를 수행하였다. 분류를 수행한 결과는 〔표 6〕의 내용과 같다. 표의 내용에서는 10분 동안 수집된 데이터를 각각 하이퍼콜 호출수를 계산한 결과의 평균, 표준편차를 나타내며 〔표 7〕는 검증 데이터를 통한 시간에 따른 분류율을 나타냈다. 결과를 통해 알 수 있듯이, t값이 120일 경우 98.2%와 99.1%의 분류율 보여주고 있으며, t값이 600과 1800일 경우100%의 분류율을 보여주고 있다. 이러한 결과는 본 논문에서 제안한 알고리즘과 이를 통해 수집한 표본 데이터가 일반적인 상황과 공격적인 상황을 충분히 분류해 낼 수 있음을 보여준다.

다음 〔그림 4〕〔그림 5〕는 t=600일 때 1초당 각5개의 특징벡터를 사용하여 이를 시각적으로 표현하기 위해 평균과 표준편차를 사용하여 특징의 차원을 축소



〔그림 4〕 K-means 결과화면



〔그림 5〕 EM 결과화면

해서 나타내었다. 각 알고리즘의 선점도를 보았을 때 K-means를 사용했을 때와 EM 알고리즘을 사용했을 때, 두 알고리즘 모두 정확한 군집화 결과를 나타내주는 것으로 볼 수 있다.

4.3 검증

다음 〔표 8〕은 군집화 결과 데이터를 PC와 PE로 검증한 결과를 나타낸다. 〔표 8〕에서 보는 바와 같이 K-menas와 EM 군집화는 클러스터 개수가 2일 때, PC에서 0.95~0.97대의 높은 클러스터 밀집도를 보이고 있다. PE에서도 0.061~0.059으로 낮은 엔트로피를 보이고 있어 2일 때 최적의 클러스터 분할 되었음을 검증할 수 있었다.

〔표 8〕 k-means와 EM군집화 검증 데이터

	Partition Coefficient(PC)		Partition Entropy Coefficient(PE)	
클러스터수	K-means	EM	K-means	EM
2	0.954032	0.978437	0.061277	0.0590105
3	0.511616	0.521539	0.837685	0.8230589
4	0.461168	0.429531	1.007766	1.0077667
5	0.420628	0.461347	1.165709	1.1527635
6	0.356598	0.470771	1.362796	1.1159567

V. 결론

클라우드 컴퓨팅 서비스에서 Guest OS를 제대로 보호하지 못해 악의적으로 사용될 경우 좀비PC로 활용되어 DDos공격을 수행하여 정상적인 서비스를 방해할 수 있다. 이를 보호하기 위해 클라우드 컴퓨팅 시스템에서 내부의 침입탐지 기술은 기존의 네트워크의 환경과는 다른 접근 방법이 필요하다. 본 논문에서는 클라우드 서비스 중 반가상화 환경에서 중요한 Guest OS를 보호하기 위한 방법으로 특정 변인인 하이퍼바이저에서 호출되는 하이퍼콜의 빈도수를 사용하여 클라우스 시스템 내부에 DDoS공격 탐지가 구현 가능함을 증명하였다. 본 논문에서 제시한 변인들을 활용하면, 클라우드 환경에서의 DDoS탐지를 보다 신속하고 정확하게 탐지할 수 있을 것이다. 향후에는 가상화 기법 중 전가상화기법이 적용된 클라우드 환경에서의 내부 공격에 대한 악성행위 탐지와 악성여부를 판단하는 방안에 대해 연구할 예정이다.

참고문헌

〔1〕 Dave Thomas, "Enabling Application Agility Software as a Service, Cloud Computing and Dynamic Languages," Journal of Object Technology, Vol.7, No. 4,2008.5.

〔2〕 김지연, 김형종, 박춘식, 김명주, "클라우드 컴퓨팅 환경의 가상화 기술 취약점 분석 연구", 정보보호학회지, 제19권 제4호, 2009.8.

〔3〕 "Cloud Computing Benefits, risks and recommendations for information security:, European Network and Information Security Agency, 2009.

〔4〕 M. Rosenblum, T. Garfinkel, "Virtual machine monitors: current technology and future trends," Computer(IEEE Computer Society), Vol. 38, Issue, pp.39-47, 2005. 5.

〔5〕 Xen 하이퍼바이저, http://www.xen.org

〔6〕 Sriram Govindan, Arjun R.Nath Amitayu Das "Xen and Co.:Communication-aware CPU Scheduling for Consolidated Xen-based Hosting Platforms" VEE 2007 No1

〔7〕 B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, I. Pratt, A. Warfield, P. Barham, and R. Neugebauer. Xen and the art of virtualization, In Proc, of the ACM Symposium on Operating Systems Principles(SOSP), Ict 2003.

〔8〕 ISS, "Network vs Host-based intrusion detection," whitepaper : Oct. 1998.
〔9〕 Y. Liao and V. Vemuri, "use of K-Nearest Neighbor Classifier for intrusion detection," Computer & Security, vol. 21, no. 5, pp. 439-448, Oct. 2002.
〔10〕 Q. Qian and M. Xin, "Research on Hidden Markov for System Call Anomaly Detection," Pacific Asia Workshop on Intelligence and Security Informatics 2007, LNCS 4430, pp. 152-159, 2007
〔11〕 L. Richard, W.Joshua, Haines, J.David, K.Jonathan, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation." Computer Networks, vol. 34, no. 4, pp.579-595, Aug. 2000.
〔12〕 C.Hoang. Protecting Xen hyper-calls. Intrusion Detection/Prevention in a Virtualizaed Environment. MS Thesis, University of British Columbia. Jul 2009.
〔13〕 Xen Performance Tools http://fossies.org/dox/xen-4.1.2/xenperf_8c.html
〔14〕 Krishna. K, Narasimha Murty. M, Genetic K-means algorithm, IEEE, Vol29, Jun 1999
〔15〕 C.F.Jeff Wu, On The Convergence Properties Of The Em Algorithm, The Annals of statistics, Vol. 11, No.1, pp. 95-103, 1983
〔16〕 Sergiios Theodoridis, Konstantinos Koutroumbas, Pattern Recognition 3rd Edition, Academic Press, 2006.

〈著者紹介〉

신 승 훈 (Seunghun Shin) 학생회원
2010년 2월: 한국외국어대학교 컴퓨터공학과 학사 졸업
2010년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 클라우드컴퓨팅 보안, 침입탐지, 시스템 보안, 악성코드

정 만 현 (Man-hyun Chung) 학생회원
2006년 2월: 동국대학교 컴퓨터학과 학사
2009년 2월: 고려대학교 정보경영공학전문대학원 정보경영공학과 석사
2010년 9월~현재: 고려대학교 정보경영공학전문대학원 정보보호학과 박사과정
〈관심분야〉 패턴인식, 시스템 보안, 네트워크 보안

문 종 섭 (Jongsub Moon) 종신회원
1981년 2월~1985년: 금성 통신 연구소 연구원
1991년: Illinois Institute of technology 졸업(전산학 박사)
1993년~현재: 고려대학교 전자 및 정보공학부 교수
〈관심분야〉 생체인식, 침입탐지, 운영체제