情報保護學會論文誌
第 22 卷 第 6 號, 2012. 12

ACAS를 통한 클라우드 가상화 내부 환경 보안성 강화 연구

박 태 성,† 최 도 현, 도 경 화, 전 문 석‡
숭실대학교

A Study on ACAS for Enhanced Security in Cloud Virtualization Internal Environment

Tae-sung Park,† Do-Hyeon Choi, Kyoung-Hwa Do, Moon-seog Jun‡
Soongsil University

요 약

최근 다양한 형태의 서비스를 제공하기 위하여 클라우드 컴퓨팅 서비스의 활용도가 증가됨에 따라 가상화 기술이 급부상 하면서 가상화 영역의 접근에 대한 안전성과 신뢰성 등 보안 문제가 이슈화되고 있다. 하이퍼바이저는 복수의 운영체제를 탑재할 수 있는 환경을 제공해주는 시스템으로 가상화 계층에 접근이 가능하면 모든 Agent에 손상을 가져올 수 있기 때문에 통제권 상실이나 권한탈취 등 여러 공격들의 대상이 될 수 있다. 본 논문은 클라우드 가상화 내부 환경의 취약점으로 인하여 Agent에 발생 가능한 보안 위협을 정의하고, 분석 결과를 기반으로 접근통제 Agent를 통하여 클라우드 가상화 내부 환경 보안성 강화 방안을 연구한다.

ABSTRACT

As the utilization of cloud computing service rapidly increases to meet demands for various forms of service recently, the virtualization technology has made a rapid rise, further leading to some issues related to security, such as safety and reliability. As a system to provide environments what multiple virtual operating systems can be loaded, hypervisors may be a target of various attacks, such as control loss and authority seizure, since all the agents fcan be damaged by a malicious access to the virtualization layer. Therefore, this paper was conducted to investigate the access control for agents and suggest a plan to control malicious accesses to the cloud virtualization internal environment. The suggested technique was verified not to have effect on the performance of the system and environment through an analysis of its performance.

Keywords: Access Control, Cloud Virtualization, Agent Management, Hypervisor

I. 서 론

클라우드 서비스는 자동화의 편리성을 제공하고 동적 확장성에 용이 함으로써 자원의 공동이용, 인터넷을 이용한 정보서비스, 정보시스템의 아웃소싱, 다양한 디바이스 환경 등에 사용되고 있다.

최근 가상화 내부 공간의 하드웨어 가상화(INTEL VT, AMD-V 등)와 같은 새로운 기술들이 개발됨에 따라 추가적인 보안위협이 제기 되고 있으며, 이러한 이슈로 인하여 클라우드 컴퓨팅 활성화를 위한 보안 요소가 필요하다. 특히 가상화 영역의 데이터 교환 및 가상화 영역의 접근에 대한 보안 기술의 요구가 계속 증가 할 것으로 예상된다[1].

가상화 영역의 보안 요구사항은 클라우드 서비스 구축시 제공하는 서비스의 범위, 제공방법, 구현방법 등에 따라 다양하게 변경 될 수 있기 때문에 추가적인 보안위협에 대한 대책 마련이 필요하다[2]. 가상화 영역의 보안 요구사항은 기술적 측면에 따라 클라우드 서비스 사용자가 위치한 외부 영역, 가상화 영역인 내부 영역으로 분류될 수 있다. 외부 영역은 사용자 식

접수일(2012년 9월 28일), 수정일(1차: 2012년 10월 22일, 2차: 2012년 12월 6일), 게재확정일(2012년 12월 6일)
† 주저자, parkstar161@gmail.com
‡ 교신저자, mjun@ssu.ac.kr

별 및 인증, 데이터 암호화, 인증 정책, 접근제어 등에 대한 기술들로 기존 보안 기술들이 적용되어 많은 보안 솔루션들이 제공되고 있지만 내부 영역은 가상화된 클라우드 서비스 인프라 구조로 인해 기존 보안 기술들이 적용하기 어렵다.

본 논문에서는 이런 문제를 해결하기 위해 클라우드 가상화 내부 환경을 분석하고 내부영역에 악의적인 접근을 차단하기 위하여 접근통제기법을 제안하였다.

2장은 클라우드 서비스 가상화 내부 환경의 보안 취약성에 대해 분석하고, 3장은 접근통제 에이전트 적용방안을 제안한다. 그에 따라 4장은 제안 내용에 대한 보안성에 대해 검증하고 5장은 결론으로 마친다.

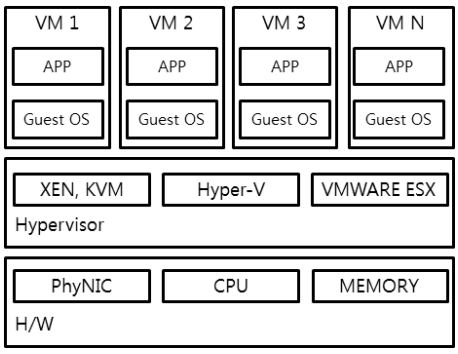
II. 관련연구

2.1 클라우드 가상화 내부 환경

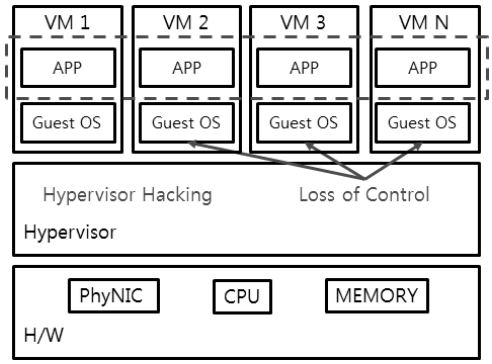
클라우드 가상화 내부 환경은 그림 그림 1과 같이 최하위 물리영역의 H/W, 다양한 O/S를 탑재할 수 있는 Hypervisor, 그리고 다수의 Virtual Machine으로 구성되어 있다. 아래 그림 1은 클라우드 가상화 내부 환경을 나타낸다.

2.2 클라우드 서비스 가상화 영역 취약점 분석

NIST(National Institute of Standards and Technology), CSA(Cloud Security Alliance)는 클라우드 컴퓨팅 환경에서 공격자들이 클라우드 자원을 비도덕적으로 사용하고 자원을 남용할 수 있는 위협요소에 대해 분석하였다[3][4]. 내부 영역인 가상화 환경에는 서비스 거부와 같은 해킹공격과 바이러스에 대한 위협 등 기존 환경의 취약성이 클라우드 환경에 적용될 수 있다[5][6][7]. 초기 썬과 리눅스 운영체제에서만 설치가 가능하였던 루트킷 같은 악성코드 또한 현재 다양한 운영체제에서 해킹요소로 작용되고 있으며, 그에 따른 해결방안은 미흡한 상태이다.



〔그림 1〕 클라우드 가상화 내부 환경 구성도

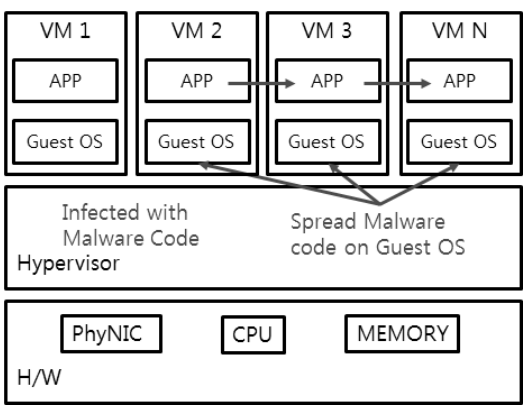


〔그림 2〕 서버에 저장된 모든 이용자의 자료 유실 및 손실

일반적으로 가상화 내부의 취약점은 하이퍼바이저 해킹으로 인한 통제권 상실, 가상화 취약점 상속에 대한 호스트 OS의 감염으로 인한 게스트 OS 간 악성코드 감염, 하이퍼바이저를 통한 게스트 OS로 악성코드 확산 등이 존재한다[8][9]. 〔그림 2〕과 같이 클라우드 서비스가 자원을 통합, 재분배 하여 공유하는 가상화 영역의 특징으로 인해 서버에 저장되어 있는 모든 데이터의 유출 및 손실과 시스템의 취약점을 상속할 수 있는 문제가 존재한다.

그림 3와 같이 호스트 기반 하이퍼바이저의 경우 하이퍼바이저를 경유하여 악성코드나 바이러스가 모든 호스트 OS에 감염이 전파될 수 있는 취약점이 존재한다.

가상 OS위에 백신을 설치할 경우 하이퍼바이저 레



〔그림 3〕 호스트 OS를 통한 게스트 OS 바이러스 감염

벨에서 움직이는 악성코드나 바이러스 등에 무방비한 상태가 된다. 이는 기존 시스템 백신 어플리케이션의 시스템 콜 방식과 가상화 시스템의 시스템 콜 방식의 차이로 인해 가상 머신 내부 데이터에 대해서 접근하는 방법이 다르기 때문이다.

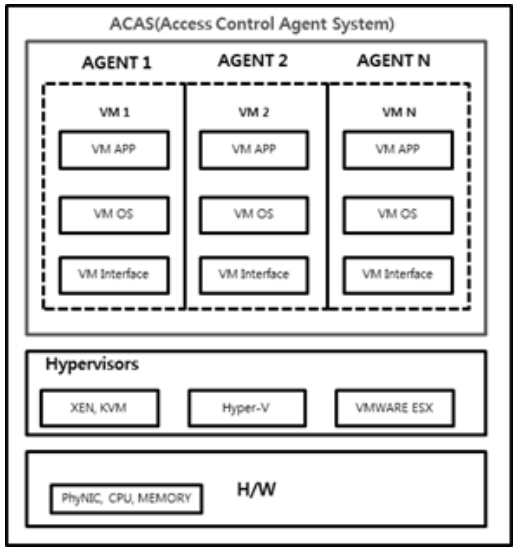
또한 가상 OS위에 설치된 백신은 하이퍼바이저에 접근할 권한이 낮기 때문에 가상 OS영역의 악성코드나 바이러스 탐지에 있어서 제약이 있다. 결론적으로 가상 OS를 컨트롤 하고 실제 데이터 통신 경로를 제어하는 특정권한을 가진 하이퍼바이저 레벨에서 보안 기술이 적용되어야 한다.

III. 제안하는 Agent 관리 기법

3.1 제안하는 Agent 시스템 구성도

본 논문에서는 하이퍼바이저와 동일 레벨에서 악의적인 접근을 차단하기 위해서 접근제어 관리 및 인증을 하는 시스템을 ACAS(Access Control Agent System)라고 정의한다.

제안하는 ACAS구조는 클라우드 서비스 제공자가 운영중인 다수의 Agent에 대한 인증이 제공됨으로써 Agent에 대한 통합관리와 외부로의 접근에 대한 관리 및 통제가 가능하다. 제안하는 ACAS구조는 현재 가상화 내부에서 발생하는 비인가된 접근과 호환성에 대한 문제를 해결한다. 〔그림 4〕는 가상화 영역의 구조와 ACAS의 위치를 나타낸다.



〔그림 4〕 가상화 내부 환경 및 ACAS의 시스템 구성도

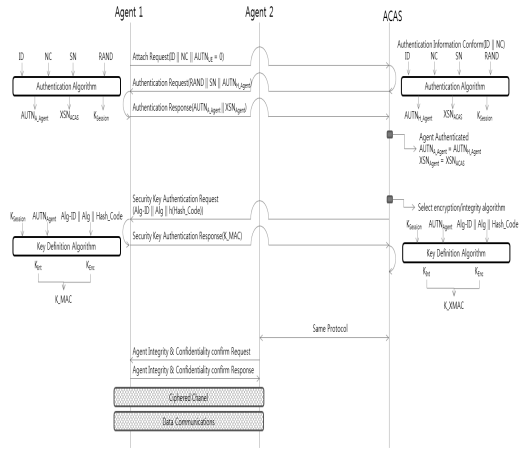
하이퍼바이저는 호스트기반 하이퍼바이저와 베어메탈 기반 하이퍼바이저로 나뉜다. 제안하는 ACAS구조는 호스트기반 하이퍼바이저 방식을 사용하였으며, 호스트기반 하이퍼바이저의 특성을 활용함으로써 OS를 통해 통신하는 방식의 성능 저하단점을 최소화 하였다.

ACAS는 하이퍼바이저와 동일 권한을 가진 에이전트 접근제어 시스템이며, 에이전트 1, 2, … N까지 하이퍼바이저에 탑재된 모든 에이전트를 접근 제어하는 시스템이다.

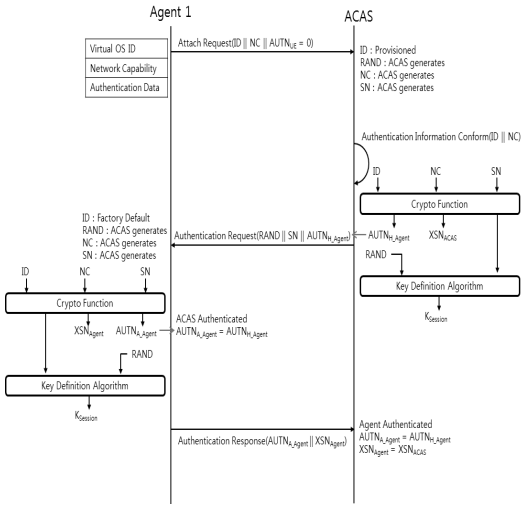
3.2 Agent 접근제어 프로토콜

〔그림 5〕는 제안하는 클라우드 가상화 내부 환경의 악의적인 접근을 차단하기 위한 인증 프로토콜의 전반적인 개요이다.

해당 프로토콜의 구성으로는 클라우드 컴퓨팅 서비스를 받아 가상환경을 구축한 Agent, 그리고 ACAS로 구성되어 있다. Agent는 가상 OS 생성시 하이퍼바이저로부터 부여받은 Virtual OS ID, Network Capability 에 대한 정보를 보유하고 있다. ACAS는 하이퍼바이저와 동일한 권한을 보유하고 있으며, 하이퍼바이저와 하이퍼바이저에 탑재된 가상 OS 영역에 대한 접근 관리 및 탐지할 권한이 있는 시스템이다. Agent와 ACAS는 동일한 Authentication Algorithm, Key Definition Algorithm을 보유하고 있으며, 양방향 통신으로 교환한 데이터를 통하여 인증, 기밀성, 무결성 등에 대한 보안 요구사항을 확인할 수 있다.



〔그림 5〕 하이퍼바이저의 악의적인 접근제어



〔그림 6〕 Agent 인증 과정

3.3 Agent 인증 과정

〔그림 6〕은 제안하는 Agent 인증 과정이다.

Agent는 다른 Agent에 접근하기에 앞서 인증을 받기 위해 보유하고 있는 값으로 접근 요청을 하며 전송하는 정보는 (1)과 같다.

$$ID, NC, AUTH_{UE} = 0 \tag{1}$$

AUTNUE = 0 : Agent에 대한 인증 시도가 이루어지지 않았음을 나타낸다.

ACAS는 수신한 ID값과 NC값이 초기 Agent에 Virtual OS가 탑재될 시 부여된 값인지를 확인한다.

ID, NC값이 올바른 값인지 확인되면, ACAS는 주어진 정보를 통해 Agent가 ACAS를 인증할 수 있는 값을 생성한다. 인증값을 생성하는 연산은 (2)와 같다.

$$Crypto\_Function(ID \| NC \| SN) = AUTN_{H\_Agent} \| XSN_{ACAS} \| Padding \tag{2}$$

ACAS는 (2)의 연산 후 (3)과 같이 연산한 후 RAND값과의 조합을 통하여 KSession값을 추출하게 되는데 해당 값은 Agent 키 교환 과정에서 사용된다.

$$Key\_Definition\_Algorithm(RAND \| Padding) = K_{Session} \tag{3}$$

ACAS는 (2)와 같은 값을 연산한 후 Agent로 ACAS에 대한 인증 정보를 전송한다. 전송하는 데이터는 (4)와 같다.

$$RAND \| SN \| AUTN_{H\_Agent} \tag{4}$$

ACAS는 Crypto Function에서 추출된 AUTNH_Agent, SN, RAND 값을 Agent에 전송해준다.

Agent는 보유하고 있는 정보와 ACAS로부터 수신한 값의 조합으로부터 ACAS를 인증하게 되며, 연산하는 과정은 (5)와 같다.

$$Crypto\_Function(ID \| NC \| SN) = AUTN_{A\_Agent} \| XSN_{Agent} \| Padding \tag{5}$$

Agent는 (5)의 연산 후 (6)과 같이 RAND값과의 조합을 통하여 KSession값을 추출하게 되는데 해당 값은 Agent Key Exchange Process 과정에서 사용된다.

$$Key\_Definition\_Algorithm(RAND \| Padding) = K_{Session} \tag{6}$$

Agent는 (5)의 연산으로부터 산출된 AUTNA_Agent값과 ACAS로부터 수신한 AUTNH_Agent값을 비교함으로써 ACAS를 인증하게 된다.

Agent는 (5)와 같은 값은 연산한 후 ACAS로 Agent에 대한 인증 정보를 전송한다. 전송하는 데이터는 (7)과 같다.

$$AUTN_{A\_Agent} \| XSN_{Agent} \tag{7}$$

3.4 Agent 키 교환 과정

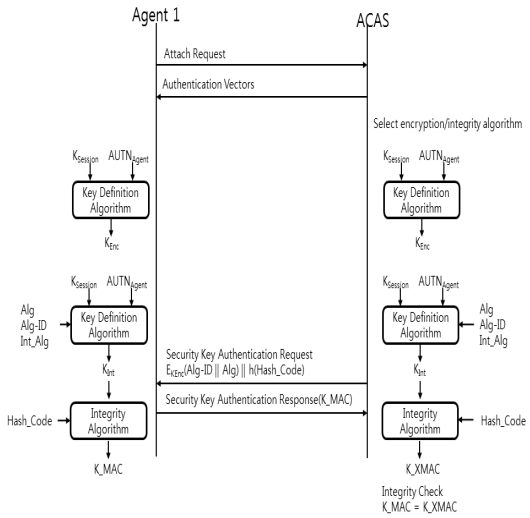
〔그림 7〕는 인증받은 Agent가 ACAS와 키를 교환하기 위한 과정이다.

ACAS에 인증이 완료된 Agent는 각각 주어진 정보와 Agent Authentication Process에서 생성한 KSession값을 활용하여 키를 생성하게 된다. ACAS는 기밀성과 무결성에 사용될 알고리즘을 선택하여 기밀성키와 무결성키를 생성하게 되며 과정은 (8),(9)와 같다.

$$Key\_Definition\_Algorithm(K_{Session} \| AUTN_{Agent}) = K_{Enc} \tag{8}$$

$$Key\_Definition\_Algorithm(K_{Session} \| AUTN_{Agent} \| Alg \| Alg-ID \| Int\_Alg) = K_{Int} \tag{9}$$

ACAS는 (9)연산을 통해 산출된 값을 Hash_Co-



〔그림 7〕 Agent 키 교환 과정

de정보를 통한 무결성 알고리즘에 삽입하여 K_X-MAC이라는 해쉬값을 추출해 내며 과정은 (10)과 같다.

$$Intergrity\_Algorithm(K_{Int} \| Hash\_Code) = K\_XMAC \quad (10)$$

ACAS는 Alg-ID와 Alg값을 KEnc로 암호화하고 Hash_Code값을 해쉬하여 Agent로 전송하며 연산 및 전송하는 데이터는 (11)과 같다.

$$E_{KEnc}(Alg-ID \| Alg) \| h(Hash\_Code) \quad (11)$$

ACAS로부터 데이터를 수신한 Agent는 기밀성키를 구하기 위한 연산을 시작하여 연산과정은 (12)와 같다.

$$Key\_Definition\_Algorithm(K_{Session} \| AUTN_{Agent}) = K_{Enc} \quad (12)$$

(12)과정에서 얻은 KEnc값을 사용하여 ACAS로부터 받은 데이터를 복호화하여 무결성키와 무결성에 사용될 해쉬값을 구한다. 연산 내용은 (13), (14), (15)와 같다.

$$D_{KEnc}(E_{KEnc}(Alg-ID \| Alg)) = Alg-ID \| Alg \quad (13)$$

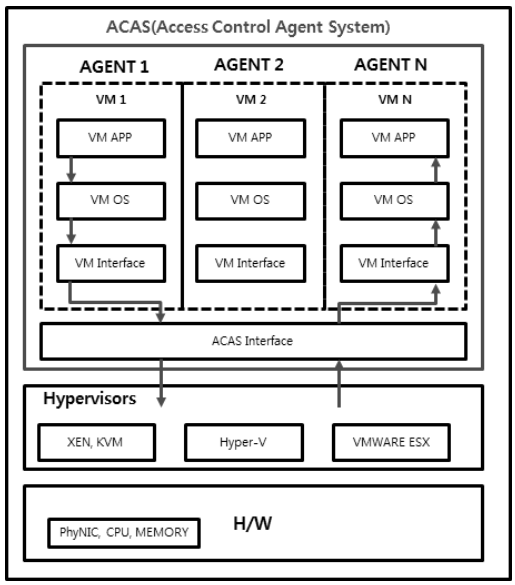
$$Key\_Definition\_Algorithm(K_{Session} \| AUTN_{Agent} \| Alg \| Alg-ID \| Int\_Alg) = K_{Int} \quad (14)$$

$$Intergrity\_Algorithm(K_{Int} \| Hash\_Code) = K\_MAC \quad (15)$$

연산을 완료한 후 Agent는 ACAS로 K_MAC값을 전송한다. ACAS는 Agent로부터 수신한 K_MAC값이 K_XMAC값과 동일하면 무결성 체크가 완료된다.

3.5 인증된 Agent간 통신

〔그림 8〕은 가상화 내부영역의 인증된 Agent간 통신 흐름을 표현한 것이다.



〔그림 8〕 인증된 Agent간 통신 과정

ACAS를 통하여 인증된 Agent는 다른 Agent간 안전한 통신이 가능하며, 하이퍼바이저를 Agent간 내부통신, Agent간 외부통신 모두 ACAS의 관리를 통하여 악의적인 접근에 대한 관리 및 통제가 가능하다. 이러한 구조는 가상 OS를 운영하고 있는 각 클라우드 서비스 제공자가 독립적으로 공유된 데이터에 대한 보호가 가능하게 한다. 각각 분산되어 있는 Agent는 ACAS를 통한 데이터 교환이 이루어 짐으로써 데이터 흐름에 대한 탐지가 가능하며, 기존 하이퍼바이저를 통한 악성코드 및 바이러스 등에 감염되는 문제를 ACAS가 관리함으로써 하이퍼바이저를 통한 악의적인 접근에 대한 취약성을 해결한다.

IV. 분석 및 성능평가

본 논문에서 제안하는 ACAS는 안전한 데이터 흐름을 제어 및 관리하는 구조로 설계되었다. 악성코드 및 비인가된 접근에 대한 탐지율은 탐지차단의 양에 따른 정확도에 따라 평가되기 때문에 본 논문에 대한 평가는 가상화 영역에서 발생하는 System Call에 대한 기준으로 성능 분석을 진행하였다.

〔표 1〕 기존 시스템과 제안 시스템의 비교분석

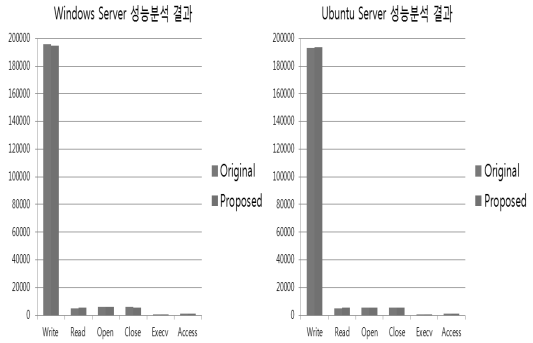
Parameter	Original	Proposed
Agent management	X	O
Secure Communication between Agent	X	O
Potential of infringement rootkit	High	Low
Strength of access control	Low	High
Error-prone	High	Low

〔표 1〕은 기존 시스템과 제안 시스템과의 성능을 비교하여 나타낸 것이다. 〔표 1〕을 통해 제안 시스템은 기존 시스템에 비해 5가지 항목에 대해 개선된 것을 알 수 있다.

Agent management와 Secure Communication between Agent는 프로토콜 구조상 기존 시스템보다 제안시스템이 보안적인 특성을 제공한다. Potential of infringement rootkit와 Strength of access control은 하이퍼바이저와 동일한 권한을

〔표 2〕 구현 환경

No	Parameter	Description
1	Cloud OS : Ubuntu 11.10 64bit Server	Openstack Multi Server (2 Servers)
2	VM OS(Total 2VM Clients)	Window7, Ubuntu10.04 (2cpu, 1024 memory, 10GB Hdd)
3	Hypervisors	Windows (Hyper-V), Ubuntu(KVM)
4	Evaluation List	System Call frequency
5	Time	24 Hours



〔그림 9〕 Windows Server(좌), Ubuntu Server (우) 환경에서의 성능분석 비교 분석 결과

가지는 ACAS로 트래픽을 관리함으로써 제안하는 시스템이 기존 시스템보다 접근제어 강도가 강하고, 루트킷 침해가능성이 낮다. 에러율은 올바른 사용자의 경우 접근제어강도가 높음에 따라 그에 따른 에러율은 낮게 측정됨을 알 수 있다.

〔표 2〕는 해당 시스템을 구현한 환경을 나타낸다. 구현환경은 클라우드 환경의 서버를 구축하고 VM OS로 각각 윈도우와 리눅스를 구축하였다.

〔그림 9〕는 〔표 2〕에서 명시된 환경에서 각각 OS별로 약 24시간동안 발생하는 네트워크 트래픽을 수집하고 그 중 중요한 항목 (write : 파일 지정자로 쓰기, read : 파일 지정자로부터 읽기, open : 파일이나 장치 열기, close : 파일 지정자 닫기, execv : 프로그램의 실행, access : 파일 권한을 검사)을 기준으로 분석한 결과이다. 〔그림 9〕의 왼쪽 그래프는 Server의 OS가 Windows인 경우이며, 수집한 트래픽이 기존시스템을 기준으로 1.1%에 해당하는 2,739개가 제안시스템에 적게 나타났다. (그림 9)의 오른쪽 그래프는 Server의 OS가 Ubuntu인 경우이며, 수집한 트래픽이 기존시스템을 기준으로 0.3%에 해당하는 784개가 더 높게 제안시스템에 나타났다.

V. 결 론

본 논문에서는 클라우드 서비스 가상화 영역에서의 해킹 및 악의적인 접근에 대한 취약점을 개선하기 위해 클라우드 가상화 영역에 하이퍼바이저와 동일 권한을 가진 ACAS를 적용함으로써 클라우드 가상화 내부 환경 접근제어 기법을 제안하였다. 제안하는 구조는 Agent를 인증함으로써 비인가된 접근에 대한 제어가 가능한 구조를 설계하였고, 성능의 효율성과 다

양한 하이퍼바이저 기술과 호환이 용이한 장점을 나타낸다. 제안 시스템은 기존 시스템에 비해 높은 안전성을 제공하며, 기존 시스템에 비해 시스템 성능에 큰 영향을 주지 않음을 성능평가를 통하여 증명하였다.

참고문헌

〔1〕 Gartner, "Gartner Identifies the Top 10 Strategic Technologies for 2012," Orlando Fla, pp. 1-5, October 2011

〔2〕 Soonki Jeong, Manhyun Chung, Jaeik Cho, Taehik Shon and Jongsub Moon, "A Research on Cloud Architecture and Function for Virtualization Security of Cloud Computing," Journal of Security Engineering, vol. 8, no. 5, pp. 627-644, 2011.10

〔3〕 Grance,T and Jansen,W, "Guidelines on Security and Privacy in Public Cloud Computing," NIST, 2011.09

〔4〕 CSA, "Security Gudiance for Critical Areas of Focus in Cloud Computing," CSA, 2011.11

〔5〕 Sung-Jae Jung, Yu-mi Bae and Woo-Young Soh, "A Study on the Secure Enhanced Efficient Web System based on Linux Virtualization," Journal of Security Engineering, vol. 7, no. 4, pp. 335-350, 2010.08

〔6〕 S.Subashini and V.Kavitha, "A survey on Security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, pp. 1-10, 2010.07

〔7〕 Gruschka.N and Jason.M, "Attack surfaces : A Taxonomy for Attacks on Cloud Services," IEEE, pp. 276-279, 2010.07

〔8〕 Korea Internet Security Agency, "Cloud Service Information Security Guide," Research and Developer Team, CSA, 2011. 10

〔9〕 Kim Jiyeon, "Vulnerability analysis of cloud computing environments, virtualization technology research," Korea Institute of Information Security & Cryptography, vol. 8, no. 5, pp. 627-644, 2009.08

〈著者紹介〉

박 태 성 (Tae-Sung Park) 학생회원
1992년 2월: 한국대학교 전자공학과 졸업
1994년 2월: 한국대학교 전자공학과 석사
1996년 3월~현재: 한국대학교 전자공학과 박사과정
〈관심분야〉 정보보호, Virtualization, 암호 알고리즘, PKI, 통신보안

최 도 현 (Do-Hyeon Choi) 학생회원
2008년 2월: 동서울대학 컴퓨터소프트웨어과 졸업
2010년 8월: 숭실대학교 컴퓨터학과 석사
2010년 3월~현재: 숭실대학교 컴퓨터학과 박사과정
〈관심분야〉 Mobile Security, Virtualization, 802.16x, PKI, Secure Coding

도 경 화 (Kyoung-Hwa Do) 정회원
2004년 2월: 숭실대학교 박사 취득
2004년~2007년: 산자부 정보보안기술(JTC1/SC27) 전문위원회 운영위원
2004년 4월~현재: 행정안정부 전문위원
2011년 9월~현재: 숭실대학교 겸임교수
〈관심분야〉 정보보호 및 개인정보보호 정책.시행, 빅데이터, 전자정부, 정보공동이용 및 활용 기술 등

전 문 석 (Moon-Seog Jun) 종신회원
1981년 2월: 숭실대학교 전산과 졸업
1986년 2월: University of Maryland Computer Science 석사
1989년 2월: University of Maryland Computer Science 박사
1986년 9월~1989년 12월: University of Mary 강사
1989년 3월~7월: Morgan State University 조교수
1989년 9월~1991년 2월: NMSU, PSL 연구소 책임연구원
1991년 3월~현재: 숭실대학교 정교수
〈관심분야〉 정보보호, 네트워크 보안, 전자여권, 암호학