Journal of The Korea Society of Computer and Information
Vol. 22 No. 11, pp. 9-16, November 2017

www.ksci.re.kr
https://doi.org/10.9708/jksci.2017.22.11.009

A Study on the Isolated Cloud Security Using Next Generation Network

Jae-Kyung Park*, Won Joo Lee**, Kang-Ho Lee***

Abstract

In this paper, we propose to present a model of cloud security that has emerged as the biggest topic of cloud computing, replacing the traditional IT environment. While cloud computing is an extension of existing IT technology, security issues and threats can be applied to traditional security technologies. However, the biggest difference between a typical computing environment and a cloud computing environment is a virtualized environment with a hypervisor. Currently, there are many weaknesses in the virtualized environment, and there are few related security products. In order for a cloud computing environment to function as a reliable IT environment, we expect more research on hypervisor-based security technologies, and we expect to secure safer cloud services through a secure model over the next generation of new-based networks.

▸Keyword: Cloud, NGN, CCN, Cloud Security, Authentication

I. Introduction

클라우드 컴퓨팅 환경에서는 가상화를 통한 해킹, 서비스 거부 공격(DoS), 계정 탈취 등 다양한 형태의 보안 위협이 존재하기 때문에 클라우드 도입을 위해서는 안전한 보안을 함께 설계하는 것이 반드시 필요하다. 클라우드 컴퓨팅 환경에서는 가상머신 자원의 유연한 할당·증감, 가상머신 간의 상호연결·연계 및 다른 호스트 사이 또는 클라우드 사이의 가상머신 마이그레이션 등의 특성으로 인하여 다양한 공격 경로가 존재한다. 구체적으로는 가상머신 간의 도청, 악성코드 전이, 자원 고갈 공격, 의도적인 가상머신 할당 후 이를 활용한 서비스 거부 공격 등이 있다. 또한 하이퍼바이저가 악성코드에 감염될 경우 동일 하이퍼바이저 상의 가상머신들에게 악성코드가 감염되어 확산될 수 있다. 악성코드가 감염되거나 보안패치가 안 된 가상머신의 마이그레이션에 따라 다른 물리적인 플랫폼으로 위험이 전이 될 수도 있다. 가상머신 이동 시 가상머신 이미지 조작, 가상머신 상의 파일 또는 프로세스의 임의 상태 변경 등의 위협 등을 예로 들 수 있다[1].

최근 CAS(Cloud Security Alliance)에서 발표한 클라우드 컴퓨팅의 7대 위협은 다음과 같다.

①클라우드 컴퓨팅 남용 및 불손한 사용
②안전하지 않은 애플리케이션 프로그래밍 인터페이스
③악의를 가지고 있는 내부 관계자
④공유 기술에 취약점
⑤데이터 유실 및 유출
⑥계정과 서비스 및 트래픽 하이재킹
⑦공개되지 않은 위협 프로파일

클라우드 컴퓨팅 남용 및 불손한 사용의 경우 악의적인 의도를 가진 사람들이 클라우드를 도입하게 되면, 모든 정보가 가상에 있게 되기 때문에 기존의 봇넷보다 더 찾아내기 힘들고 위험한 존재가 될 수 있다[2]. 또한, 안전하지 않은 애플리케이션 프로그래밍 인터페이스의 경우 애플리케이션 구축을 서두르기 위해 기존의 코드를 재사용하거나 합성해서 사용하면 보안에 취약 할 수 있다. 클라우드 컴퓨팅이 갑자기 떠오르면서 관련 경험을 가진 사람을 급하게 채용하면 도덕적으로 적합하지 않는 사람을 고용할 가능성이 높아지게 되며, 가상머신을 적절히 관리하지 못하면 하나의 작은 실수로 전체가 다 위협받을 수 있다는 우려도 제기된다. 이 외에도 데이터를 보호하기 위한 기

• First Author: Jae-Kyung Park, Corresponding Author: Won Joo Lee
*Jae-Kyung Park (jakypark@kopo.ac.kr), Dept. of Information Security, Seoul Gangseo Campus, Korea Polytechnics.
**Won Joo Lee (wonjoo2@inhatc.ac.kr), Dept. of Computer Science, Inha Technical College.
***Kang-Ho Lee (lkh@knuw.ac.kr), Dept. of Computer Information Security, Korea National University of Welfare
• Received: 2017. 10. 16, Revised: 2017. 11. 02, Accepted: 2017. 11. 20.

존의 제어방식은 새로운 클라우드 환경에서 적합하지 않을 수 있을 뿐더러 감시하기가 더 힘들다는 지적도 나온다. 이미 각종 악성사이트로 유도하는데 사용된 많은 하이재킹(hijacking) 기법에 취약하고, 서비스 제공업체의 투명성이 떨어져 고객사가 시스템 구성을 새롭게 하거나 소프트웨어 패치를 실행해야 하는지 알지 못할 때가 많다는 점도 클라우드의 주요 보안 위협 가운데 하나이다[3][4].

클라우드 서비스의 보안 이슈로는 크게 3가지의 유형으로 살펴볼 수 있다. 첫째, 통제 손실(Loss of Control)로 모든 서비스에 대한 통제가 클라우드 제공자에 의해 통제되므로 서비스 사용자는 권한, 자원, 정책에 대한 통제를 할 수 없으므로 전적으로 클라이드 제공자에 의존하므로 발생할 수 있는 보안 문제이다. 둘째, 신뢰 부족(Lack of Trust)으로 클라우드 서비스에 신뢰와 위협이 공존하며 모든 정보가 클라우드 제공자에게 위탁되므로 절대적으로 신뢰해야하는 보안 문제이다. 셋째, 다중 사용자(Multi-Tenancy)로 다중의 사용자가 서비스를 이용함에 있어 독립성을 보장받고 다른 사용자와 분리되어야 하나 물리적으로는 같은 공간에 있으며 공격자도 동일한 조건의 머신을 공유하므로 발생할 수 있는 보안 문제이다[5].

이와 같이 클라우드 서비스의 보안문제는 보다 다양한 관점에서 처리되어야하며 서비스 제공자는 이러한 문제를 해결하기 위해 다양한 보안 장치를 설계하여 적용하고 있다. 따라서 기존의 개별 서비스보다 클라우드 서비스 제공자에 의한 서비스는 전문적인 보안관리 및 통제 하에 있고 다양한 보안장비를 통한 보안 서비스도 병행하고 있기 때문에 보다 안전하다는 견해도 있다. 하지만, 자원이 공유되고 사용자가 동일한 플랫폼에 위치하여 외부 공격에 노출될 경우 파급효과가 더 크며 현재의 보안 수준으로는 완벽한 보안이 불가능하다는 지적도 만만치 않은 실정이다[6][7].

따라서 본 논문에서는 보다 더 완벽한 형태의 보안 서비스를 제공하기 위한 클라우드 서비스는 어떻게 진행되어야할 것인가에 대한 구체적이고 현실적인 연구 방법을 제안한다. 본 논문의 2장에서는 관련 연구를 설명하고, 클라우드의 보안 취약점을 살펴본다. 3장에서는 제안하는 차세대 네트워크 기반의 클라우드 보안 모델에 대하여 설명한다. 그리고 4장에서는 제안한 클라우드 보안 모델의 평가 및 결과를 분석하고, 5장에서 결론을 맺는다.

II. Preliminaries

1. Cloud Security Criteria Setting

클라우드 보안 서비스를 수행하기 위해서 보안 기준을 3가지 관점에서 살펴볼 필요가 있다. 보안의 3대 요소인 기밀성, 무결성, 가용성 측면에서 고려되어야할 내용을 살펴보도록 한다. 기밀성은 서비스되는 데이터의 통제를 상실할 경우 발생할 수 있는 보안 문제를 어떻게 해결할 것인가? 보안을 유지해야 하는 민감한 데이터는 클라우드에 안전하게 저장되어야 하며, 클라우드는 사용자의 데이터 유출을 막고 안전하게 보호해야 한다는 것이다[8]. 무결성은 클라우드가 정확하게 데이터를 처리하는지 사용자가 인지할 수 있어야 하며, 클라우드가 저장된 사용자의 데이터를 수정 없이 저장되는 것을 확신할 수 있어야 한다는 것이다. 가용성은 서비스 거부 공격 등으로 클라우드가 공격을 받으면 클라이언트의 중요 서비스가 중단되어서는 안되며, 클라우드 사업자가 폐업을 할 경우에도 사용자 데이터의 보존방안을 마련해야 한다는 것이다. 클라우드 규모가 서비스를 제공할 만큼 충분히 확보되어야 한다[9].

이러한 내용을 기반으로 구체적인 클라우드 보안 서비스에 대하여 설명한다.

2. Cloud Security Vulnerability

2.1 Virtualization infrastructure

가상화 시스템의 내부 영역에는 가상화 기술의 적용으로 가상 네트워크가 구축된다. 이러한 가상 네트워크 내에서는 가상 스위치(vSwitch)를 통해 네트워크 패킷의 스위칭이 일어난다. 가상화 서버 내부 영역에서의 네트워크 통신 즉, 가상머신 간의 네트워크 통신 시에는 가상 네트워크 패킷이 가상화 서버의 외부로 나가지 않고 가상화 시스템 내부 영역에서 스위칭 되어 전달된다. 따라서 가상화 서버 내에 구축된 가상화 네트워크는 기존의 IPS/IDS 및 방화벽에게는 보안 사각지대 (security blind zone)가 된다. 이러한 보안 사각지대로 인해 기존의 보안 장비로는 가상 네트워크 상의 네트워크 트래픽에 대해서는 침입 탐지가 제한적이다[10].

2.2 Vulnerability of Virtualization infrastructure

가상화 기술을 통해 사용자의 가상머신들이 상호 연결되어 다양한 공격 경로가 존재하며 Fig. 1과 같이 다른 가상머신 및 하이퍼바이저로 해킹을 당할 수 있으며 악성코드 등이 전파될 가능성이 매우 높다.

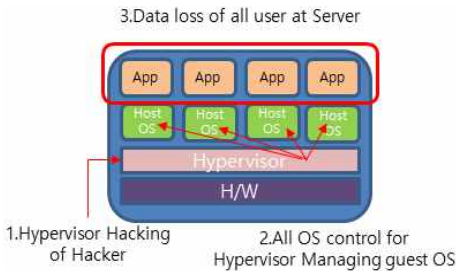

Fig. 1. Hacking of Hypervisor

또한 이를 방어하기 위한 안티바이러스도 Fig. 2와 같이 한계가 있다. 기존의 안티 바이러스는 각 호스트에 에이전트 형태로 설치되어 각 에이전트가 독립적으로 악성코드의 시그니처를 관리한다. 이러한 기존의 안티 바이러스를 가상화 환경에 적용할 경우 크게 세 가지 주요 문제점이 발생할 수 있다.

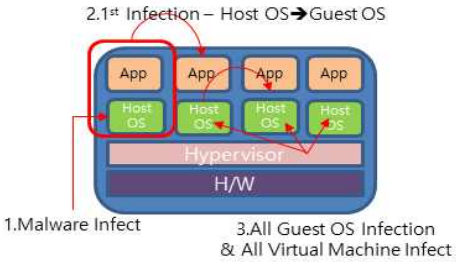

Fig. 2. All guest infection for Malware

첫째는 안티바이러스 스톰 (A/V storm)의 발생 문제이다. 가상화 환경에서 기존의 안티바이러스 기술을 적용하면 각 가상머신에 안티바이러스 에이전트가 설치된다. 일반적으로 기업 환경에서는 안티바이러스들이 동일한 시각에 악성코드 검사를 실시하도록 설정되어 있기 때문에 각 가상머신에 설치된 안티바이러스 들이 동시에 동작할 가능성이 높다. 이러한 경우 각 안티바이러스 에이전트들이 동시에 공유 디스크에 있는 파일들을 스캔하며 악성코드 검사를 하므로 시스템 전체 부하가 매우 높아진다. 이를 안티바이러스 스톰이라 한다. 안티바이러스 스톰은 해당 가상화 시스템에서 동작하는 전체 가상머신들의 성능을 현저히 저하시키는 문제를 유발한다.

둘째, 각 가상머신에 설치된 에이전트의 악성코드 시그니처를 모두 최신 시그니처로 유지·관리해야 한다. 가상화 시스템 내에 운용되는 모든 가상머신들의 보안수준을 동일하게 유지하게 위해서는 각 가상머신에 설치된 안티바이러스의 악성코드 시그니처도 동일하게 최신 상태를 유지해야 한다. 이러한 최신의 악성코드 시그니처 유지 요구사항은 동적인 생성, 중단, 재시작, 이동 등이 용이한 가상머신의 특성상 보안 관리를 복잡하게 만드는 문제가 있다.

셋째, 가상머신 상에 동작하는 안티바이러스 에이전트는 가상화 계층, 즉 하이퍼바이저 등에 존재 가능한 악성코드를 탐지할 수 없다. 하이퍼바이저의 특권 레벨(privileged level) 보다 낮은 레벨에서 동작하는 가상머신은 하이퍼바이저를 포함한 가상화 계층 내에 직접 접근할 수 있는 권한이 없다. 따라서, 가상머신 내에서 동작하는 안티바이러스 에이전트에게는 가상화 계층 영역이 보안 사각지대가 된다. 이로인해, 안티바이러스는 하이퍼바이저에 접근하여 하이퍼바이저 루트킷 등 하이퍼바이저에 침입한 악성코드를 탐지할 수 없는 제한점이 있다. 하이퍼바이저가 악성코드에 감염되어 통제권을 상실하게 되면, 해당 하이퍼바이저 상에 동작하는 모든 가상머신의 제어권 또한 상실하게 되므로 하이퍼바이저의 보안은 중요한 이슈이다[2].

2.3 IT resource sharing and insider threat

IT 자원 공유 및 멀티테넌시 환경에서 해킹 및 관리자 실수 등에 의해 사용자의 정보가 유출되며 내부자의 실수나 고의적 정보 접근으로 인한 사용자의 정보 손실 및 유출이 가능하게 된다. 즉, 인가되지 않은 다른 이용자의 정보 접근 위험이 존재하며 대량 고객의 정보 집중과 가른 고객의 정보가 혼재되어 설정 오류 및 취약한 패스워드 사용 등으로 인해 타인의 접근이 가능해지는 취약점을 가지고 있다[5]. 내부 직원에 의한 권한 이외의 정보에 대한 접근 및 유출을 막기 힘든 이유는 내부 관리자가 고객 편의성으로 고객사의 관리 정보인 아이디 및 패스워드를 관리하고 있으므로 손쉽게 정보 유출이 가능하다는 취약점을 가지고 있다. 또한 고객이 삭제한 정보가 고객의 동의 없이 백업 또는 보관되어 있으니 사용자는 이를 인지할 수 없다는 것 또한 보안상 큰 문제를 야기할 수 있다. 따라서 기존의 하이퍼바이저 환경에서의 클라우드 보안은 다양한 보안의 위협을 그대로 가지고 있으므로 클라우드 서비스를 확산하는 데는 한계가 있으며 보안의 비용도 매우 클 수 밖에 없는 구조이다.

2.4 Unclear security responsibilities and complexity of policy

서비스 모델에 따라 IT 자원의 관리 책임이 서로 달라 보안 책임의 분할이 어려울 수 있다. 또한 접속 환경 및 이용 단말 등이 다양하여 보안책임 소재를 명확히 규명하는 것이 불가능하여 복잡한 보안 정책 적용이 필요하다. 클라우드 서비스 별 IT 자원의 제공 및 관리 범위에 따라 IaaS(Infrastructure as a Service), PaaS(Platform as a Service), SaaS(Software as a Servce)로 범위를 구분하여 서비스 및 책임 소재를 관리해야 한다. 이렇게 규모 및 범위를 구분한다고 하더라도 각 가상화 서비스에 대한 기술 적용이 동적 환경으로 구성됨에 따라 가상머신의 생성, 할당, 회수 등에 대한 정책 수립 및 가상머신 이용률 관리, 이동 등의 복잡성으로 인해 보안 정책을 수립하는데 큰 어려움이 있다.

3. Next Generation Network CCN

CCN(Contents Centric Network)은 기존 위치 중심의 IP 체계를 콘텐츠 중심의 네트워크 체계로 구현하여 콘텐츠 전송 능력을 향상시키고 강화된 보안체계를 제공하는 새로운 개념의 차세대 네트워크이다. CCN은 특정한 인증 이름 규칙을 콘텐츠에 부여하여 IP가 없이도 콘텐츠의 내용으로 데이터를 처리할 수 있는 메커니즘으로 Fig. 3과 같이 구성이 된다[11].

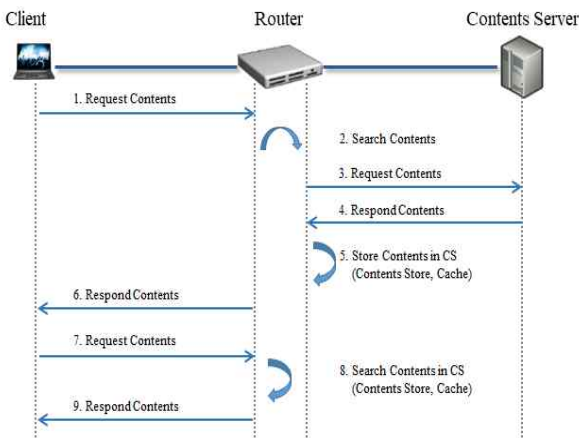

Fig. 3. CCN Protocol Process

CCN을 이용할 경우 요청자 또는 공격자는 콘텐츠 서버에 접근 자체가 불가하여 서버의 운영체제, 웹, 응용, 서비스 등을 전혀 알 수 없다.[12] 공격자가 요청을 통해 콘텐츠를 받을 수는 있으나 이는 콘텐츠 저장소에 저장된 캐시 내용을 받는 것이며 이 또한 정상적인 인증을 해결해야만 받을 수 있는 개념이다[13]. CCN의 가장 큰 특징은 IP가 없으며 콘텐츠를 요청하기 위해 INTEREST를 요청하면 해당 데이터에 대해 DATA로 응답하므로 프로토콜이 매우 단순하여 대량의 데이터를 처리하는데 용이하고 커넥션의 배제로 공격 포인트 제거했다. 또한 데이터에 대한 인증정보를 이름 규칙과 함께 적용함에 따라 기본적으로 보안을 탑재한 구조라라 할 수 있다[14][15].

III. The Proposed Scheme

클라우드 시스템의 보안상 취약점을 보완할 수 있는 방법을 살펴보고 이를 차세대 네트워크를 통해 새로운 형태의 모델로 제안한다.

1. Analyzing cloud security measures

1.1 Security measures through integrated security log collection and analysis

플랫폼 구축시 보안 강화를 위해 클라우드 플랫폼을 활용하여 로그를 통합적으로 수집하여 관리하고 분석하는 시스템 구축이 필요하다. 또한 중복된 많은 정보 중 선택적으로 정보를 수집하여 정보 연계가 용이하도록 하는 것이 중요하다. 기존 시스템에서 분산적으로 발생하는 로그를 클라우드 플랫폼에서 취합하고 선별하여 관리하여 분석과 연계가 용이하도록 구축한다.

1.2 Security measures through isolation and intensive analysis

동적으로 관리 가능한 논리적 가상자원들의 그룹을 형성하여 가상머신, 스토리지 등의 논리적 그룹 또는 Zone을 동적으로 변경 가능하게 구축한다. 이때 추가적으로 격리 존을 구성하여 비정산 가상머신이 발생하면 격리시켜 피해 확산을 방지하고 비정상 가상머신을 집중 분석하도록 한다.

격리된 가상머신은 기본적으로 서비스는 진행하되 제한적인 서비스를 진행하며 지속적인 모니터링을 한다. 이때 자원에 대한 접근제어, 자원할당 정책 제어, 동작 제어 등의 보안정책을 별도로 구분하여 수행하고 만약 비정상적인 악성행위가 발견되면 즉시 서비스 또한 중지시키고 완전 격리시킨다.

1.3 User security measures

클라우드 서비스에 대한 보안을 위하여 클라우드 서비스망 보호를 위한 네트워크 보안(전통적인 보안 조치)을 통해 보안을 강화할 수 있다. 또한 가용성을 위한 회선 이중화로 서비스의 단절을 최소화 한다. 보안장비인 DDoS 대응 솔루션, IPS, 방화벽, VPN 등을 도입하여 기본적인 외부 공격에 대해 안정적인 서비스를 유지할 수 있도록 한다. 또한 서비스 망의 보안 관제를 적용하여 보안장비 및 하이퍼바이저 단에 이상 징후까지 모니터링 할 수 있는 관제를 병행한다. 추가적으로 클라우드 사업자가 제공하는 Managed 보안 서비스를 이용하여 사용자별 네트워크 보안 정책 설정, 관제 서비스를 활용하여 방화벽 필터링 정책, IPS 보안 정책, 이용자 영역에 대한 보안 관제 서비스를 강화할 필요가 있다. 그리고 가용성 확보를 위한 백업 및 DB 서비스 이용도 추가할 것을 권장한다. 외국 기업의 경우, Managed 서비스 보다는 이용자가 제3자 솔루션을 자율적으로 선택 이용하는 정책을 활용하고 있으므로 국내에서도 보다 다양한 형태의 보안 서비스를 제공하는 방법을 찾아야 한다.

1.4 Security measures with SecaaS

최근 클라우드 보안 기술들은 효과적인 적용을 위해 SecaaS (Security As A Service) 형태로의 제공 요구가 제기되고 있다. 즉, 클라우드에서 제공되는 다양한 IT 자원들이 서비스 형태로 제공되는 것과 마찬가지로 보안 기능 또한 클라우드 서비스로서 제공되는 형태가 요구되는 것이다.

CSA(Cloud Security Alliance)에서는 10가지의 보안 영역을 정의하고 이들의 클라우드 보안 서비스 구현을 위한 가이드를 제공하고 있다. 또한, 가트너에 의하면 SecaaS에 대한 요구는 이미 정점을 지났으며, SecaaS는 2~5년 이내에 클라우드 보안 영역에서 일반적으로 통용될 기술로 예상되고 있다. SecaaS의 기본 개념은 보안 기능을 클라우드 서비스 형태로 제공하는 것으로 클라우드 인프라 및 서비스를 위한 보안 서비스 뿐만 아니라, 클라우드가 아닌 기존의 IT 환경을 위한 보안 기능 또한 클라우드 서비스화 하는 것을 포함한다. 단, 하이퍼바이저 기반 가상화 침입 대응 기술의 관점에서 한정적으로 본다면, 해당 기술을 기존 클라우드 환경에 어떻게 잘 통합시키어 기존의 다른 클라우드 서비스들처럼 클라우드 사용자들이 필요할 때 필요한 만큼의 보안 서비스만을 비용 지불을 통해 사용하게 해 주는가 하는 것이다.

하이퍼바이저 기반 가상화 침입 대응 기술을 SecaaS 형태로 제공하는 것은 앞 절에서 설명한 기술들이 적용되는 것에 더해, 사용자(Tenant) 인지 기반의 사용자 별 독립된 보안 기능 적용 기술, 자원 대여의 근간이 되는 보안 기능 사용량에 대한 정의 및 측정 기준, 비용 청구 기술, 컴플라이언스 준수 관련 기술, 감사 기술 등이 추가적으로 적용되어야 한다. 그러나 이러한 기능들은 SecaaS로 제공되어야 할 보안 서비스들에게 공통으로 제공되어야 할 기술들이며 공통적인 SecaaS 프레임워크의 표준화와 그에 기반한 연구·개발 등을 통해 효율적으로 제공되어 질 수 있다. 다양한 국내외 기업들이 아마존 웹서비스(AWS)와 같은 클라우드 서비스 기업들과 협력해 SecaaS를 출시하고 있으며 어떤 기업들이 어떠한 솔루션을 클라우드 기반 서비스로 제공되는지를 잘 검토하여 활용하는 것도 좋은 방안이라고 할 수 있다.

2. Proposed Isolated Cloud Security Model Based on Next Generation Network

기존 클라우드의 보안 모델은 나름대로의 방안을 제시하고 있으나 여전히 보안의 문제를 해결하지 못하고 있는 실정이다. 따라서 본 논문에서는 기존에 제시된 클라우드 보안모델과는 전혀 차별화된 형태의 보안 모델을 제안한다. 본 논문에서는 기존의 네트워크 체계를 벗어나 내부에서는 CCN을 활용하여 네트워크 해킹에 대한 원천적인 보안이 가능하도록 구성한다.

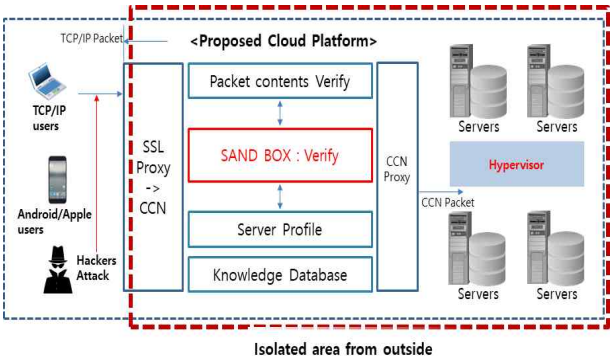

Fig. 4. Isolated cloud systems from outside

Fig. 4와 같이 내부 클라우드 시스템은 기존의 TCP/IP 프로토콜을 사용하지 않는다. 왜냐하면 외부로부터의 공격은 100% TCP/IP를 통한 공격이므로 맨 앞단에서 단순히 내부 망으로 전달하기 위한 프락시로만 TCP/IP를 연동하는 기능을 가지고 있다. CCN의 프로토콜 특성상 기존 TCP/IP의 공격과 같이 플러딩 형태의 공격은 CCN에서는 적용되지 않는다. 즉, 콘텐츠가 없을 경우 CCN은 데이터를 전달할 수 없는 형태로 구성이 되어 있으므로 이러한 특성을 활용하여 클라우드를 구성하면 외부에서 클라우드 시스템에 대한 DDoS 공격을 전혀 할 수 없게 된다[15].

Table 1. process content of Ingress packet

List	Value
Identification	Store with content hash value for requested ID
Method	Method Value of request content
Content	Real Content
Location	Physical Location Info. of Requested content
Operation	Service Env. & Process Info. of Content

2.1 Packet Contents Verification

외부에서 유입되는 패킷은 TCP/IP 패킷 형태로 세션이 생성된 이후 내부망의 서버로의 데이터의 요청에 대한 서비스가 대부분이다. 이때 요청되는 패킷은 특정한 데이터를 요청하는 것으로 콘텐츠에 대한 정보를 포함하고 있다. 이 콘텐츠 정보는 TCP/IP 패킷에서 추출하여 CCN의 콘텐츠로 변환한다.

Table. 1에서 정의한 내용을 토대로 외부에서 데이터가 유입되면 이를 내부적으로 처리하기 위해 패킷을 변환한다. 변환된 데이터 타입만이 내부적으로는 유효하며 TCP/IP의 패킷은 전혀 서비스되지 않는다. 이처럼 내부에 직접적인 TCP/IP 서비스를 수행하고자 하여도 이를 처리할 수 있는 방안이 없으므로 외부 공격 및 내부 자료 유출에 강력히 대응할 수 있다. 이러한 패킷에 대한 검증을 위해 Process Packet(PP)이라는 자료구조를 정의하였으며 내부적으로 TCP/IP 패킷을 PP로 변환하여 처리한다. 이때 PP내부에는 내부망의 서버로의 요청되는 콘텐츠가 포함되어 있어야만 한다. 즉 내부 망에 자료에 대한 정보 없이 임의로 요청되는 콘텐츠는 내부에 전달하지 않는다. PP의 과정을 거치기 위해서는 내부망의 정확한 디렉터리 정보를 가지고 있어야 한다. 외부에서 요청되는 콘텐츠를 엑세스하기 위해 내부적으로 레지스트리를 가지고 있어야 한다. 추가적으로 CCN 프로토콜은 IP를 전혀 사용하지 않고 콘텐츠를 주변에 브로드캐스팅하여 데이터를 요구하는 방식이다. 따라서 만약에 본 논문에서 제안하는 모델 내부에 악성코드가 침입하였을 경우 다른 서버로 전파하거나 또는 외부로 데이터를 유출하기 위한 행위를 수행하려고 할 때도 본 모델은 클라우드 내부에 어떠한 IP도 사용하지 않으므로 이러한 공격은 무용지물이 된다[15].

```
Data: Interest Packet
Result: Content Packet
while ccninterestqueue is not empty do
    dequeue ccninterestqueue;
    decompose content name of Interest Packet;
    /* Check in the file /WEB-INF/web.xml */ ;
    if Servlet request then
        parse parameters;
        reconstruct content name to ccnuri;
        if ccnuri is not in ccnurihashmap then
            if first segment then
                insert ccnuri into ccurihashmap;
                assign processing thread;
                write Content Packets to ccncache ;
            else
                do nothing;
        else
            fetch Content Packet in ccncache;
    else
        perform ccnpush algorithm or discard the
        packet;
```

Fig. 5. CCN get packet processing algorithm

Fig. 5는 패킷이 수신되어 내부 망으로 전송하기 위해 처리하는 과정을 알고리즘으로 구현한 것이다.

2.2 Registry Verification

패킷을 내부 망으로 처리하기 위해서는 내부 웹서버의 디렉터리에 대한 정보를 가지고 비교한다. 이를 본 논문에서는 레지스트리로 정하며 내부에 가지고 있는 호스트 정보에 대한 값을 자료구조 형태로 저장하고 있다. 이는 서버에 주기적으로 자동 요청하여 갱신하며 서버로 요청되는 모든 정보는 이 데이터를 기반으로 처리한다. 만약 해당 디렉터리를 검색하지 못할 경우 비인가 요청으로 해석하여 패킷을 버린다. Fig. 6은 이러한 레

지스트리의 구조체이다.

```
Define of Internal Host Registy
{
        DWORD regid;                    // registy ID
        DWORD flags;                    // Flags (read, write, execution etc)
        STRING fullpathinfo;            // Location
        STRING otherconnectinfo;        // connection information (ingress URI)
        STRING svcobj;                  // service name
        STRING names;                   // Host name
        DWORD multiaccess;              // concurrent access number
        DWORD ip_address;               // server IP address
        DWORD srcnetcnt;                // source network object number
        DWORD seclabel;                 // object level
        DWORD updatetime;               // object update time
}
```

Fig. 6. Host Registry Structure

2.3 Profile Verification

추가적으로 레지스트리를 검색하여 요청하는 데이터가 있을 경우 추가적으로 프로파일 검사를 수행한다. 프로파일은 해당 콘텐츠가 요청되는 종류에 따라 해당 서버에 요청이 맞는지 아니면 해당 서버에는 전달해서는 안 되는 명령이 포함되어 있는지를 판단하는 행위기반의 정책이다. 이 프로파일은 서버별로 수행 가능한 메소드를 정의하고 요청되는 콘텐츠를 분석하여 해당 메소드가 적절한지를 판단한다. Fig. 7은 이러한 내용을 저장하는 프로파일에 대한 정의이다.

```
Define of Server Protection Profile
{
        DWORD policyid;         // Policy ID
        DWORD action;           // pass/deny/encrypt
        DWORD reserved;         // reserve field
        DWORD flags;            // SPD flag
        STRING ifn;             // interface name
        STRING objtype;         // object type
        STRING srcnet;          // source network - egress NW Info
        STRING dstnet;          // destination network - ingress NW Info
        STRING svcobj;          // service object – egress request Info
        STRING ugroup;          // user group – egress user Info
        STRING timeobj;         // time object – service request time Info
        STRING vpnpolicy;       // VPN ploicy
        STRING comment;         // Log comments
        DWORD limit_Kbps;       // bandwidth limit
        DWORD priority;         // Traffic delay priority
        DWORD dpilevel;         // DPI level;
                                // { DLEVEL_CRITICAL, ..., dpilevel }
        DWORD timeout;          //timeout for delay attack
}
```

Fig. 7. Profile Structure

2.4 Operation Protocol

앞 절에서 설명한 절차에 따라 패킷이 수신되면 CCN 패킷으로 변환하며 이때 내부 망에서 사용하기 위한 형태로 변환된다. 변환된 패킷은 내부 콘텐츠를 검증하기 위해 레지스트리와 프로파일을 검증하는 과정을 거치며 검증된 콘텐츠만 내부 망으로 전송된다. 이러한 과정을 통해 기존의 방어 방법보다 매우 효과적으로 해킹을 막을 수 있다. 이러한 일련의 과정을 프로토콜 형태로 표현하면 Fig. 8과 같다.

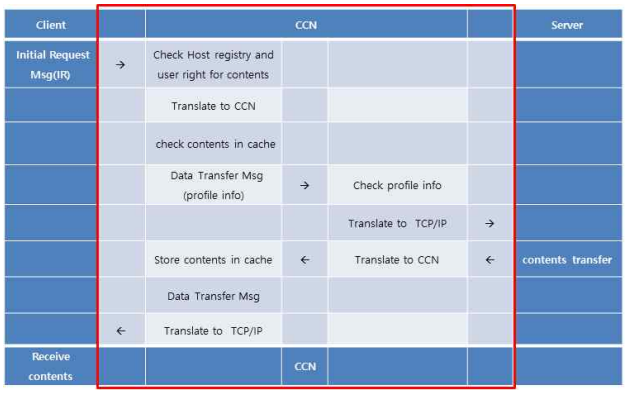

Fig. 8. Operation protocol flow

IV. Evaluation and Result Analysis

1. Evaluation

본 논문에서 제안한 모델을 평가하기 위해 Fig. 9와과 같은 환경에서 테스트를 진행한다.

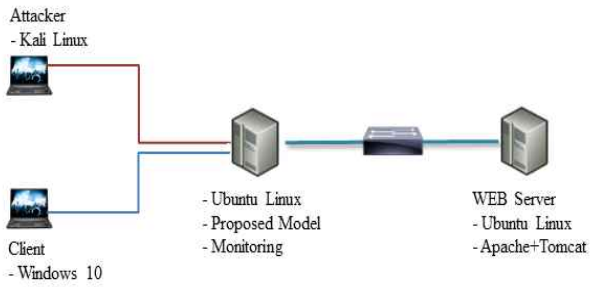

Fig. 9. Test Environment.

내부적으로 운영체제는 Linux를 사용하고 CCN에서 배포한 CCNx를 수정 개발하여 테스트를 진행한다. 내부망의 서버에는 웹 서비스를 테스트하기 위해 Apache 웹서버와 Tomcat도 설치하였다. 클라이언트는 정상적인 사용자와 공격자 PC를 따로 구분하여 정성적인 사용자는 웹 브라우저를 이용하여 웹서버에 접근하였고 공격자는 임의의 공격 패킷 즉, Syn 플러딩 공격 등을 수행한다.

2. Result Analysis

2.1 Result

본 논문에서 실험에서는 Fig. 10과 같이 기존/신규 종류의 DDoS 공격용 Flood 패킷들이 유입되어 네트워크 대역폭 잠식이 시작되는 즉시, 모든 종류의 TCP/IP 기반 Flood 패킷들을 전부 Garbage 및 Junk 패킷으로 분류/인식하여 즉각 차단하는 결과를 보였다.

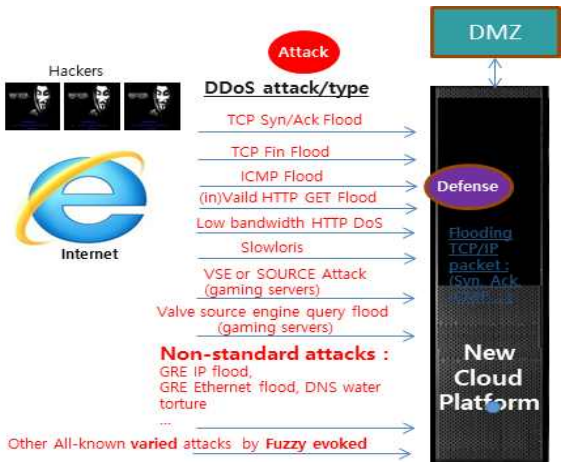

Fig. 10. Defense of DDoS

DDoS 공격 방어의 패킷 차단 기능과 policing 기능을 혼합 사용하여, 네트웍이 DDoS 공격에도 항상 서비스에 영향을 받지 않는 상태로 유지가 되는 것을 확인할 수 있었다.

Fig. 11. Unable to communicate directly with TCP/IP in the internal network

Fig. 11과 같이 내부 망에 악성코드가 감염되어 외부의 해커와 긴밀하게 통신을 위해 TCP/IP로 통신을 시도해도 제안 모델에서는 어떠한 IP도 없으며 TCP/IP를 이해하지 못하므로 외부와의 접속은 완전히 불가능하다. 본 제안 모델에서는 기본적으로 CCN 프로토콜을 사용하기 때문에 해커의 TCP/IP 연결 요청에 응답하지 않는다. 또한 해커의 해킹 코드에는 기본적으로 취약성이 있는 대상을 발견 및 발견 후 공격을 위해 동적인 함수 코드들(ls(), lsc(), Net(), IP(), ICMP(), Ether(), display(), help(), show(), str(), send(), sr(), report_port(), srloop(), sniff() 등)을 포함하고 있다. 따라서 내부의 프로파일에서 웹 서버에 대한 프로파일을 미리 정해놓았기 때문에 해커의 동적 코드를 즉각적으로 프로파일에서 분석/인식하고, 이러한 종류의 패킷은 해커의 비정상적 네트워크 연결요청으로 인식하여 응답을 하지 않는다.

2.2 Sand Box Verification

현재 지식 기반의 엔진은 빅데이터 및 패턴 기반의 엔진을 연동해야하는 문제가 있으므로 본 논문에서는 이를 해결하기 위해 샌드박스를 도입하였다. 샌드박스는 서비스되는 데이터 중 파일 형태를 서버나 PC에 전달하기 전에 임의적으로 실행함으로써 해당 파일이 유해한지 무해한지를 파악할 수 있다. 따라서 샌드박스를 이용하여 실시간으로 파일의 유해성 여부를 파악한다.

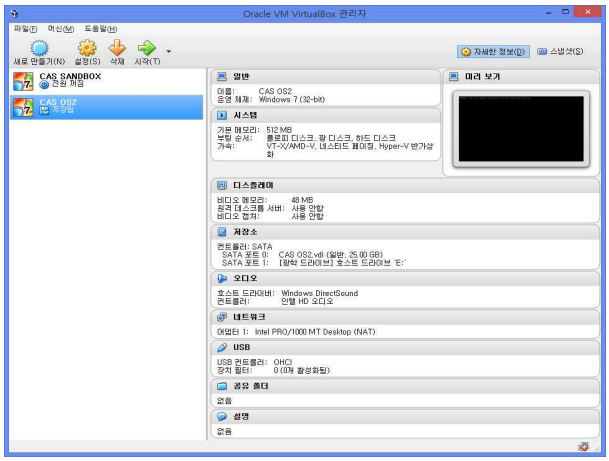

Fig. 12. Oracle Sand Box - VirtualBox

Fig. 12는 Oracle 샌드박스를 네트워크로 CAS와 연결하기 위해 네트워크에 대한 설정 화면을 나타내고 있다. 본 논문에서는 CAS 내부에서 샌드박스 API를 이용하여 외부의 Oracle 샌드박스와 연동한다. 이때 Cuckoo 샌드박스 분석 기능을 통해 실시간으로 악성코드를 검출한다. 이러한 새로운 프로토콜인 CCN을 활용하여 클라우드 시스템에 적용할 경우 완벽하게 클라우드 보안을 수행할 수 있다. 또한 외부에서 유입되는 패킷을 CCN으로 변환화는 과정에서 콘텐츠에 대한 검증, 샌드박스를 활용한 파일 검증, 접속하고자 하는 서버의 프로파일 정보를 확인하여 보다 정교한 보안이 가능하다. 이러한 데이터를 지식기반의 데이터로 축적하여 신규 공격에도 적극 활용이 가능하다.

V. Conclusions

본 논문에서는 기존의 클라우드 시스템에서 발생할 수 있는 보안의 문제점과 이를 해결하는 기존의 보안 시스템에 대해서 설명하였다. 기존의 방식은 여전히 보안에 대한 문제를 가지고 있으며 무엇보다도 클라우드 환경에서는 하나의 서버에 문제가 발생하면 모든 서버로 확대될 수 있다는 치명적인 문제점이 존재한다. 이를 위해 격리 시키는 모델도 제안이 되었지만 동일한 환경에서 감염된 서버는 격리된 이후에도 다른 서버를 감염시킬 수 있는 문제점을 가지고 있다. 이러한 근본적인 문제를 해결하기 위해서는 새로운 형태의 모델이 필요하다.

따라서 본 논문에서는 차세대 네트워크로 주목받고 있는 CCN을 활용하여 새로운 클라우드 보안 모델을 제안하였다. 제안한 모델을 활용할 경우 기존의 문제 즉, DDoS와 같은 외부

공격과 악성코드 감염을 통한 내부 자료 유출 등의 문제를 CCN의 장점을 활용하여 해결할 수 있다. 향후 이러한 CCN을 보다 폭넓게 활용하여 서버뿐만 아니라 사용자 PC까지 확대할 경우 기존의 보안에서 해결하지 못한 많은 문제점들을 해결할 수 있을 것이라고 판단한다.

REFERENCES

[1] Ronald L. Krutz 외, “Cloud Security – A Comprehensive Guide to Secure Cloud Computing”, WILEY, 2010.03

[2] http://news.inews24.com/php/news_view.php?g_serial = 952021&g_menu=020200&rrf=nv, 2016.4.20.

[3] http://www.dt.co.kr/contents.html?article_no=201510 05 02150251753001, 2015.10.05.

[4] http://www3.opengroup.org/getinvolved/workgroups / cloud computing, Oct. 2012.

[5] Hyang-Jin Lee, “Security Consieration for use of Secure Cloud Services,” CloudSec 2012, (2012) Mar 13; Seoul, Korea.

[6] https://cloudsecurityalliance.org/guidance/csaguide.v 3 .0.pdf, Nov. 2011.

[7] https://downloads.cloudsecurityalliance.org/initiatives / top_threats/Top_Threats_Cloud_Computing_Survey_20 12.pdf, Nov. 2012.

[8] Ju-Young Kim, Jong-Pyo Kim, Kyung-Ho Lee, Hyuk-Jun, Kim, Yong-Hoi Kim, Chun-Sik Park, “A Guide of Security Management for Cloud Computing Services,” KISA, 2010

[9] S.K. Un, N.S. Jho, Y.H. Kim and D.S. Choi, “Cloud Computing Security Technology,” Electronics and Telecommunicati ons Trends. ETRI. Vol. 24, No. 4, pp. 79-88. Aug. 2009.

[10] Young-Sang Shin, “Hypervisor-based Security Techn ology of Virtualization Environment for Cloud Computing,” The Clouds 2012, (2012) September 24-25; Seoul, Korea.

[11] Parc Homepage, http://www.parc.com/

[12] 4WARD Homepage, http://www.4ward-project.eu/

[13] PURSUIT Homepage, http://www.fp7-pursuit.eu/

[14] CCN & CCNx Homepage, http://www.ccnx.org/

[15] Hyung-Su Lee, Jae-Pyo Park, Jae-Kyung Park, "A Network Transport System Using Next Generation CCN Technology," Journal of The Korea Society of Computer and Information, Vol. 22, No. 10, pp. 93-100, Oct. 2017.

Authors

Jae-Kyung Park

1994: BS, Department of Computer Engineering, Dongguk University

1996: MS, Department of Computer Science, Hongik University

2002: PhD, Department of Computer Science, Hongik University

Current position: Professor, Department of Information Security, Seoul Gangseo Campus, Korea Polytechnics

Areas of interest: Network security, cyber security

Won Joo Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University, Korea, in 1989, 1991 and 2004, respectively. Dr. Lee joined the faculty of the Department of Computer Science at Inha Technical College, Incheon, Korea, in 2008, where he has served as the Director of the Department of Computer Science. He is currently a Professor in the Department of Computer Science, Inha Technical College. He has also served as the Vice-president of The Korean Society of Computer Information and the Editor-in-Chief for the Journal of The Korean Society of Computer Information. He is interested in parallel computing, internet and mobile computing, and cloud computing.

Kang-Ho Lee received the M.S. and Ph.D. degrees in Electronic Engineering from Chungang University, Korea, in 1986 and 1991, respectively. Dr. Lee joined the faculty of the Department of Dept. of Computer Information Security, Korea National University of Welfare, Pyeongtaek, Korea, in 2003, He has also served as the President of The Korean Society of Computer Information. He is interested in information security and digital image processing.