공 학 석 사 학 위 논 문

오픈스택 플랫폼에서 클라우드 포렌식에 관한 연구

2015년 8월

부 경 대 학 교 대 학 원

정 보 보 호 학 협 동 과 정

한 수 빈

공 학 석 사 학 위 논 문

오픈스택 플랫폼에서 클라우드 포렌식에 관한 연구

지도교수 신 상 욱

이 논문을 공학석사 학위논문으로 제출함.

2015년 8월

부 경 대 학 교 대 학 원

정보보호학협동과정

한 수 빈

한수빈의 공학석사 학위논문을 인준함.

2015년 8월 21일

주 심 이학박사 이 경 현 

위 원 이학박사 신 원 

위 원 이학박사 신 상 욱 

목 차

(표 차례)

(그림 차례)

Study on Cloud Forensic in OpenStack-based Cloud Environment

Han Su Bin

Department of Interdisciplinary Program of Information Security,
Graduate School, Pukyong National University

Abstract

Cloud forensics is difficult to collect evidence because the definition of evidence data in cloud environment is not clear and it needs more cooperation from CSPs. Also, it may cause a loss of evidence because the evidence data is physically distributed and the cloud computing resources may be present in virtual space. In this thesis, therefore, we propose a cloud platform-based forensic procedures for cloud forensics and classify the log data by cloud layer according to the proposed procedure. By building a cloud environment using the OpenStack, we describe limitations of the actually collected evidence and propose methods to address the challenges of cloud forensics. It ensures improved integrity and reliability of evidence. To deal with these limitations, we make a snapshot of the instance and analyze details of a snapshot. In addition, the collection and storage of evidence data in the cloud are performed by the processes as intactly as possible that are defined in the OpenStack.

Ⅰ. 서론

1.1 연구 배경

최근 국내외 기업들의 클라우드 컴퓨팅 서비스 사용이 급격하게 증가하고 있다. 미국의 정보 기술 연구 및 IT분야 전문 조사 기관인 가트너(Gartner)는 “개인용 클라우드(Personal Cloud)가 개인 PC를 대신해 디지털 라이프의 새로운 허브역할을 할 것”이라고 전망하며 “현재의 단말 의존적인 IT환경이 클라우드 기반 서비스 환경으로 전환될 것”이라고 발표했다[27].

클라우드 컴퓨팅의 성장은 미국의 네트워킹 하드웨어 및 보안 서비스 회사인 시스코(Cisco)의 보고서[28]에서 나타나고 있다. 보고서에 따르면, 2013년을 기준으로 향후 5년간 전 세계 데이터센터 트래픽이 약 3배가량 증가하고, 이 중 76%를 클라우드 트래픽이 차지할 것으로 내다봤다. 시스코는 전 세계 데이터센터 트래픽이 2013년 3.1제타바이트에서 2018년 8.6제타바이트로 약 3배가량 증가해 23%의 연평균 성장률을 기록할 것으로 전망했다. 여기서 ‘데이터센터 트래픽’의 범위는 데이터센터와 사용자 간(data center-to-user), 데이터 센터 간(data center-to-data center) 그리고 데이터 센터 내에서 발생하는 트래픽 모두를 포함한다. 또한, 2018년까지 전 세계 가정 내 인터넷 사용자의 53%가 소비자용 클라우드 스토리지를 사용할 전망이며, 사용자 한 명당 발생하는 월 평균 클라우드 스토리지 트래픽은 2013년 186메가바이트에서 811메가바이트로 증가할 것으로 예측된다.

이러한 클라우드 컴퓨팅 서비스의 급격한 성장과 함께 클라우드 서비스를 대상으로 하는 해킹 사고 또한 최근 들어 다양한 형태로 증가하고 있다[1].

사이버 범죄의 사례로 악의적인 공격자들이 해외 업체를 대상으로 한 DDoS공격에 클라우드 컴퓨팅 서비스에서 제공하는 가상 머신(Virtual Machine)을 좀비(Zombie)로 활용하는 특이사항이 모니터링 되고 있다. 이 외에도 클라우드 서비스의 간편한 결제로 여러 가상 머신을 생성하고 즉시 사용할 수 있는 장점과 삭제가 용이하다는 점을 이용하여 불법적인 행위를 클라우드 컴퓨팅 서비스에 사용하고 있다. 이에 따라 성장하는 클라우드 컴퓨팅 시장과 증가하는 사이버 범죄에 대응하여 클라우드 컴퓨팅에 대한 이해를 바탕으로 디지털 포렌식 시스템을 체계적으로 준비할 필요가 있다[2].

특히 포렌식 절차에서 증거 수집의 경우 클라우드 컴퓨팅은 자원이 가상 공간에 존재하거나, 증거 데이터가 물리적으로 분산되어 있기 때문에 일반적인 디지털 포렌식 조사 수행이 어렵다. 또한, 사용되는 클라우드 플랫폼 및 가상화 기술 구성이 매우 다양하고 복잡하게 나타나지만 포렌식 조사관의 접근 가능 범위는 관리 시스템 정도로 제한적이다[3]. 이는 클라우드 환경에서 효과적인 증거 수집을 위해 기존의 포렌식 수사 방식과 다르게 접근해야 함을 의미하지만, 이를 지원하는 뚜렷한 포렌식 도구, 실질적인 정책 또는 절차들이 전무한 상황이다[2].

1.2 연구 목적 및 내용

클라우드 환경에서 증거로써 의미있는 데이터들이 대부분 가상화된 영역에 저장되어 있음을 고려해볼 때, 이를 증거로 활용하기 위해서는 해당 데이터를 획득하는 절차의 신뢰성 문제를 우선적으로 해결해야 한다.

따라서 본 논문에서는 추상화된 클라우드 계층에 따른 기존 포렌식 절차상의 데이터 수집 방법에 관한 한계를 분석하고, 수집한 증거 데이터의 신뢰성 보장 및 다양한 클라우드 환경에 보다 유연하게 적용할 수 있는 디지털 증거 수집 절차를 제안한다. 해당 절차는 클라우드 구성 요소들 중 물리적인 자원들을 가상화하여 논리적으로 구성할 수 있도록 하며, 가상화된 자원들을 서비스 목적에 따라 폭넓게 활용할 수 있도록 관리 체계를 제공해주는 클라우드 플랫폼을 기반으로 한다.

또한, 제안한 절차에 따라 증거를 획득하기 위해 클라우드 컴퓨팅 플랫폼 기반의 클라우드 환경을 구축하여 구축된 환경에서 사용자 가상머신 및 수집 가능한 증거데이터를 정리 및 분석하고, 획득한 증거데이터에 대한 신뢰성과 무결성 향상을 위한 추가적인 절차 방안을 제안한다. 제안된 방안을 통해 획득한 증거 데이터를 포렌식 도구를 이용하여 분석한다. 결과적으로 오픈스택 기반 클라우드 환경에서 클라우드 포렌식이 가지고 있는 문제를 고려한 클라우드 포렌식을 수행한다.

클라우드 환경 구축을 위한 클라우드 컴퓨팅 플랫폼은 오픈스택 플랫폼을 이용하며 테스트 환경은 오픈스택 기반으로 수행된다. 또한, 클라우드 환경에서 저장되고 수집되는 데이터는 최대한 오픈스택에서 정의되어있는 과정으로 수행한다. 획득한 증거 데이터에 대한 포렌식 도구는 Encase v.7.08을 사용하여 분석하였다.

Ⅱ. 관련 연구

2.1 클라우드 컴퓨팅

클라우드 컴퓨팅은 광범위하게 사용하는 용어로 많은 기업이나 기관에서 다양하게 정의되는데, Cloud Computing Usecase Group에서는 “다양한 클라이언트 디바이스에서 필요한 시점에 인터넷을 이용하여 공유 풀에 있는 서버, 스토리지, 애플리케이션, 서비스 같은 IT 리소스에 쉽게 접근할 수 있게 하는 모델” 이라고 클라우드 컴퓨팅을 정의하고 있다[29]. 클라우드 컴퓨팅을 구분하는 기준은 여러 가지가 있지만 일반적으로는 제공 서비스와 운용 방식을 기준으로 한다[4].

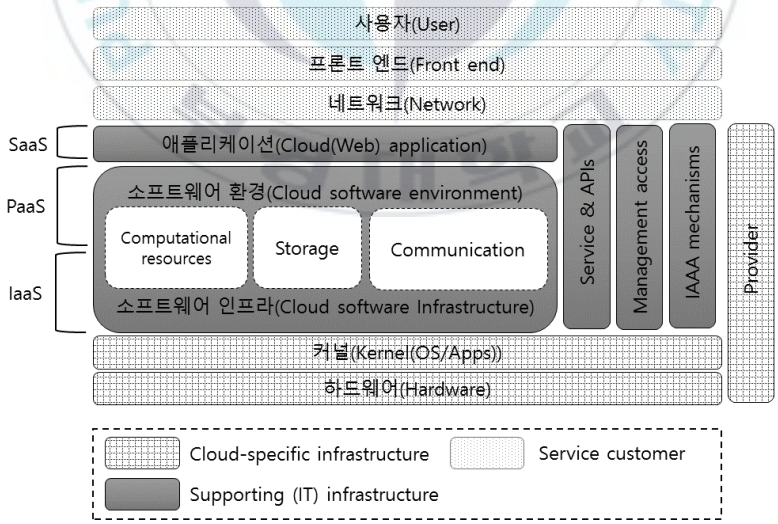


그림 1 클라우드 컴퓨팅 서비스 모델[1].

그림 1은 클라우드 컴퓨팅의 제공 서비스에 따라 클라우드 컴퓨팅 서비스 모델을 나타내고 있으며 IaaS(Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service)로 나뉜다.

▪ IaaS(Infrastructure as a Service)

IaaS는 서버, 스토리지 등의 서비스를 제공하는 것으로 인프라 서비스를 구축할 때 필요한 환경을 제공해주며, 대표적인 서비스는 아마존의 AWS, IBM, KT등 국내외 여러 기업에서 제공하고 있다[5].

▪ PaaS(Platform as a Service)

PaaS는 응용프로그램이 실행되는 환경을 서비스 형태로 제공하는 플랫폼 서비스로, 대표적인 서비스 사례로 구글 앱 엔진, Microsoft의 Azure 등이 있다.

▪ SaaS(Software as a Service)

SaaS는 소프트웨어를 설치하지 않고 서비스 제공자가 제공하는 소프트웨어를 사용하며, 대표적인 서비스 사례로 Salesforce.com과 구글 앱스를 들 수 있다.

NIST에서는 4가지의 운용 모델(Deployment Model)에 따라 공용, 사설, 하이브리드, 커뮤니티 클라우드로 정의한다.

▪ 공용 클라우드(Public Cloud)

공용 클라우드는 기업 내부가 아닌 외부의 사용자에게 자원을 제공하는 방식으로, 대중 사용자를 위한 서비스다. 공용 클라우드 벤더는 일반적으로 사용자에게 접근제어 메커니즘을 제공한다.

▪ 사설 클라우드(Private Cloud)

사설 클라우드는 기업의 내부 구성원을 대상으로 하는 서비스로, 별도로 서비스를 구축하기 때문에 서비스 구축비용과 전문 인력이 필요하며 일정 규모 이상이 되어야 비용 절감 효과가 크다.

▪ **하이브리드 클라우드(Hybrid Cloud)**

하이브리드 클라우드는 공용과 사설을 혼합한 서비스 개념으로 이 모델을 사용하는 사용자들은 중요하지 않은 처리는 공용 클라우드에서 수행하고, 중요한 서비스와 데이터는 사설 클라우드에서 운영한다.

▪ **커뮤니티 클라우드(Community Cloud)**

커뮤니티 클라우드는 특정 요구사항이나 공통의 업무와 같이 동일한 목적 또는 관심을 가진 사용자들이 구성된 그룹에 의해 운영되고 사용된다. 그룹의 구성원은 클라우드의 데이터를 접근하고 공유할 수 있다.

2015년 국내 클라우드 시장은 정부의 진흥정책과 공공기관의 민간 클라우드 도입에 따른 시장의 성장, 해외 기업들의 국내 시장 공략 강화, 이에 따른 국내 기업들의 서비스 경쟁력 강화, 신규 서비스 출시 등이 맞물리면서 클라우드 시장의 가파른 성장이 전망되고 있다. 또한 2015년 3월 '클라우드 컴퓨팅 발전 및 이용자 보호에 관한 법률(이하 클라우드 발전법)'이 국회 본회의를 통과해 9월부터 본격 시행된다.

하지만 여전히 클라우드 산업은 여러 가지 문제점을 가지고 있다. 특히, 클라우드 발전법에서도 신뢰성 향상 및 이용자 보호에 관한 내용이 주요 내용으로 다루어졌다. 이는 아직 클라우드 산업이 정보보안에 대한 문제점을 해결하지 못했다는 반증이다. 이에 따라, 안정적인 클라우드 서비스 제공과 사이버 범죄로부터 대처하고 예방할 수 있는 기술적 연구가 필요하다.

2.2 클라우드 포렌식

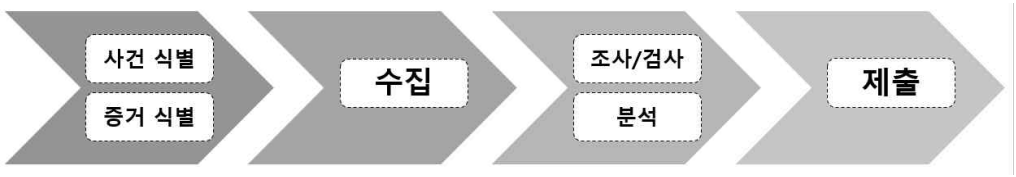


그림 2 디지털 포렌식 절차

디지털 포렌식이란 전자증거물을 사법기관에 제출하기 위해 PC나 노트북, 휴대폰 등 각종 저장매체 또는 인터넷 상에 남아 있는 각종 디지털 정보를 식별, 보관, 분석, 제출하는 일련의 과정이다[2]. 클라우드 포렌식은 클라우드 컴퓨팅 환경에 디지털 포렌식 기술을 적용하는 것을 의미한다. 그림 2는 이러한 절차를 도식화하여 나타낸 것이다[8].

최근 클라우드를 통해 일어나고 있는 각종 범죄들은 클라우드 포렌식의 중요성을 알려주고 있다. 해커가 클라우드를 활용하면 블록 IP 주소를 우회할 수 있고, 웹 사이트 생성과 도메인 등록이 훨씬 편리해지기 때문에 이러한 방법을 이용한 사이버 공격이 계속 증가하는 추세이다. 또한, Attack as a Service라는 말까지 등장하면서 해커들이 해킹 공격을 마치 클라우드 기반의 서비스처럼 제공한다.

가. 클라우드 포렌식 연구

J. Dykstra and A. T. Sherman(2012)[6]은 클라우드 포렌식에서 클라우드 환경의 신뢰를 이해하는 것에 대한 중요성을 언급하며 IaaS 클라우드 컴퓨팅에서의 6개의 신뢰모델을 제시했다. 또한, 클라우드 컴퓨팅 환경에서 포

렌식 조사를 수행할 때 선택할 수 있는 방법과 수집을 위한 도구 평가에 대해 초점을 맞추고 있다. 클라우드 환경에서의 신뢰는 법원에서 판사에게 제출된 증거를 믿을 수 있느냐에 대한 결정을 의미한다. 예를 들어 기존의 포렌식에서 용의자의 머신이 물리적으로 존재할 때, 이미징 작업을 위해 하드디스크를 빼내는 경우 디스크를 제대로 읽기 위해 하드 드라이브 하드웨어를 신뢰해야 한다. 만약, 라이브 포렌식의 경우 포렌식 도구를 실행 할 때 하드웨어뿐만 아니라 호스트 운영 시스템의 무결성도 신뢰 할 수 있어야 한다. 표 1은 [6]에서 제시한 신뢰모델이며, 각 계층은 하위에서 상위로 올라갈수록 신뢰가 누적된다. 각 계층에서 다른 포렌식 수집 활동이 이루어지며, 계층에 대한 신뢰성과 안정성의 정도는 각각 다르게 요구된다.

표 1 IaaS 클라우드 환경에서의 신뢰 계층[6]

계층	클라우드 계층	획득 방법	신뢰 요구
6	Guest application/data	Depends on data	Guest OS, Hypervisor, Host OS, Hardware, Network
5	Guest OS	Remote forensic software	Guest OS, Hypervisor, Host OS, Hardware, Network
4	Virtualization	Introspection	Hypervisor, Host OS, Hardware, Network
3	Host OS	Access virtual disk	Host OS, Hardware, Network
2	Physical hardware	Access physical disk	Hardware, Network
1	Network	Packet capture	Network

J. Dykstra and A. T. Sherman(2013)[7]은 오픈스택 클라우드 컴퓨팅 플랫폼에 대한 디지털 포렌식 도구를 구현했다. 이 논문에 따르면 저자는 도구는 게스트 가상 머신 (VM) 또는 하이퍼바이저를 신뢰할 필요 없이 클라

우드 공급자의 지원을 필요로 하지 않고 포렌식 데이터에 대한 액세스를 제공하기 때문에 관리 면에서 사용자 중심의 포렌식 기능을 위한 솔루션이라고 말한다. 그러나 클라우드에 있는 데이터의 보존과 오픈스택의 업데이트에 따른 문제와 같은 문제점을 가지고 있다.

S. Zawoad and R. Hasan(2013)[8]은 디지털 포렌식의 수사 과정에 대해 설명하면서 클라우드 포렌식에 대한 전체적인 개요를 명시하면서 클라우드 환경에서 CSP(Cloud Service Provider: 클라우드 서비스 제공자)에 대한 의존도가 높을수록 포렌식 증거를 획득하는데 어렵다는 점과 일반적인 디지털 포렌식과는 다르게 접근해야 한다고 설명하고 있었다. 해결방안으로는 CSP에 대한 의존도를 줄이고, 보안과 신뢰성을 고려한 포렌식 모델을 구축해야 한다고 언급했다. CSP의 의존도를 줄이기 위해 오픈스택 계층에서 증거를 획득하려는 점은 같지만, 독립적인 모델을 구축하는 방안이 IaaS를 제공하는 클라우드 플랫폼에서 증거를 수집하고 획득함으로써 신뢰성과 CSP의존도를 줄이려는 본 논문의 방향과 다르다. 이 외에도 클라우드 환경에서의 신뢰성 확보[9]와 클라우드 컴퓨팅 환경의 신뢰성 증가를 위한 보안 로깅[10]에 관하여 많은 연구자들이 언급하고 있었다.

나. 클라우드 포렌식의 과제 및 해결방안

클라우드 컴퓨팅 시장의 성장에 따라 클라우드 컴퓨팅을 이용한 다양한 사이버 공격은 계속해서 증가할 것으로 예상되지만 이에 따른 포렌식 연구 및 관련 도구는 미비한 실정이다. 미국표준기술연구소(NIST, National Institute of Standards and Technology)는 클라우드 컴퓨팅 환경에서의 디지털 포렌식 조사의 문제점을 바탕으로 9가지 카테고리로 분류한 65가지의 과제를 보고서로 발표했다. NIST의 보고서에 따르면 클라우드가 많은 기

술적, 조직적, 법률적 사안들을 어렵게 만들고 있으므로 이러한 문제점을 파악하는 것이 첫 번째 단계이며, 다음 단계는 기존의 표준과 모범사례를 통해 새로운 표준과 기술이 충족될 수 있도록 간격을 메우는 것이라 주장한다[11]. 그림 3은 NIST에서 발표한 클라우드 컴퓨팅 포렌식 과제 보고서에서 클라우드 포렌식이 가지고 있는 문제에 관해 9가지를 분류하고 세부사항을 그림으로 나타낸 것이다. 이 문서에서는 클라우드 포렌식의 과제를 구조적, 법적, 표준, 데이터 수집, 분석, 안티 포렌식, 역할 관리, 사건 초동조치, 교육으로 9가지를 분류하였다.

그림 3 NIST에서 발표한 클라우드 포렌식 과제[11].

클라우드 포렌식이 어려운 이유 중 하나는 클라우드 환경은 기존의 데스크탑 환경과는 달리 증거에 대한 물리적인 접근이 불가능하고 증거 데이터가 가상공간에 존재할 수 있기 때문에 데이터의 휘발성이 강하고 증거자료 보관 및 획득에 신뢰성 및 무결성 문제가 생길 수 있다. 이러한 문제점들은 클라우드 포렌식에 반드시 해결하는 과제이며, 클라우드 환경에서 보다 더 유연한 접근 방법이 필요하다.

I. Y. Jung and Insoon. J(2011)[21]에서는 클라우드에 저장되는 정보의 보호 및 신뢰성 확보에 집중하며, 프로비넌스(provenance)를 통해 클라우

드 상의 데이터 신뢰확보에 대한 효과적이고 유용한 해법을 제안한다. 클라우드 컴퓨팅 환경에서 데이터의 신뢰 확인을 위한 프로비넌스를 도입할 수 있는 모델을 제안하고 데이터의 무결성은 프로비넌스의 무결성과 함께 검증되어 데이터 신뢰성 확인에 신뢰성을 더해준다. 또한 클라우드 컴퓨팅 환경에서 프로비넌스 관리 시스템 이외에 다른 장치를 특별히 요구하지 않기 때문에 적용이 쉽고 시간적, 공간적 관점에서 오버헤드가 크지 않아 프로비넌스를 통한 신뢰확인을 할 수 있다는 장점이 있다.

S. Simos, C. Kalloniatis and E. Kavakli(2014)[12]에서는 클라우드 포렌식에서 해결해야 할 과제에 집중하고 있다. 표를 통해 현재 클라우드 포렌식이 가진 과제를 나타내고 구체적인 문제점을 언급한다. 또한, 클라우드 포렌식이 가진 문제점에 대한 해결방안을 다른 논문을 참고하여 표로 나타내었다. 구체적인 해결방안을 제안한 것이 아니라 다른 논문을 연구한 결과를 바탕으로 개괄적인 방향을 제시하고 있다. 아래의 표 2와 표 3은 클라우드 포렌식의 과제와 이 논문에서 주로 다룰 문제에 해당하는 해결방안을 표로 나타낸 것이다.

표 2. 클라우드 포렌식 과제[12].

클라우드 포렌식 과제/단계	해당되는 서비스 모델		
	IaaS	PaaS	SaaS
식별(Identification)			
Access to evidence in logs	부분	√	√
Physical inaccessibility	√	√	√
Volatile data	√	X	X
Client side identification	√	X	√
Dependence on CSP - Trust	√	√	√
Service Level Agreement (SLA)	√	√	√
보존-수집(Preservation-Collection)			
Integrity and stability	√	√	√
Privacy	X	√	√
Time synchronization	√	√	√
Internal Staffing	√	√	√
Chain of custody	√	√	√
Imaging	X	√	√
Bandwidth limitation	√	X	X
Multi-jurisdiction - collaboration	√	√	√
Multi-tenancy	√	√	√
검토-분석(Examination-Analysis)			
Lack of forensic tools	√	√	√
Volume of data	X	√	√
Encryption	√	√	√
Reconstruction	√	√	√
Unification of log formats	√	√	√
Identity	√	√	√
증거 제시(Presentation)			
Complexity of testimony	√	√	√
Documentation	√	√	√
카테고리 분리 안됨(Uncategorised)			
Compliance issues	√	√	√

표 3 클라우드 포렌식 해결방안[12].

클라우드 포렌식 과제	해결방안
Volatile data	Live investigation
	Cost globalization between CSPs
	Data synchronization
	Continous synchronization API
Dependence on CSP -Trust	Accountable cloud
	Trust Cloud framework
	Eucalyptus framework
	Layers of trust model
Integrity & stability -Privacy&multi-tenancy	Digital signature
	Distributed signature detection framework
	Multi-tenancy model
	Cyber-crime forensic framework
	Proofs Of Retrievability(PORs)
	Trusted Platform Module
	SLA contracts
Chain of custody	Trained and qualified personnel
	Organizational policies and SLAs
Lack of forensic tools	Proofs Of Retrievability (PORs)
	Management plane
	Forensic Open-Stack Tools (FROST)

2.3 오픈소스 기반 클라우드 컴퓨팅 플랫폼

클라우드 플랫폼은 클라우드를 구축하기 위한 소프트웨어로 서버, 스토리지, 네트워크와 같은 자원들을 수집, 제어, 운영하기 위한 클라우드 운영체제이다. 좀 더 편리하게 클라우드 컴퓨팅 환경을 구현하기 위해서 클라우드 플랫폼이 생겨났고 오픈소스 소프트웨어로는 오픈스택(OpenStack), 클라우드스택(CloudStack), 유칼립투스(Eucalyptus), 오픈네뷸라(Open Nebula) 등이 있다[13].

본 논문에서는 오픈소스 기반 클라우드 컴퓨팅 플랫폼인 오픈스택을 사용한다. 오픈스택은 Rackspace사와 NASA의 합작으로 시작된 IaaS(Infrastructure as a Service) 클라우드 플랫폼 프로젝트이다. 이 프로젝트의 목표는 하드웨어에 구애받지 않고 실행되는 클라우드 서비스를 제공하는 것이다[14].

오픈스택은 컴퓨트, 오브젝트 스토리지, 이미지, 인증 서비스 등이 유기적으로 연결되어 하나의 커다란 클라우드 컴퓨팅 시스템을 구축한다. 또한, 오픈스택의 구성요소는 개별적인 프로젝트로 명명되어 개발이 진행되며 그림은 오픈스택 공식 홈페이지에 설치 가이드에서 제공하는 IceHouse버전의 전체적인 구조를 도식화 한 것이다[15]. 그림 4에서는 각 프로젝트의 이름과 함께 간단하게 전체적인 흐름을 같이 보여준다.

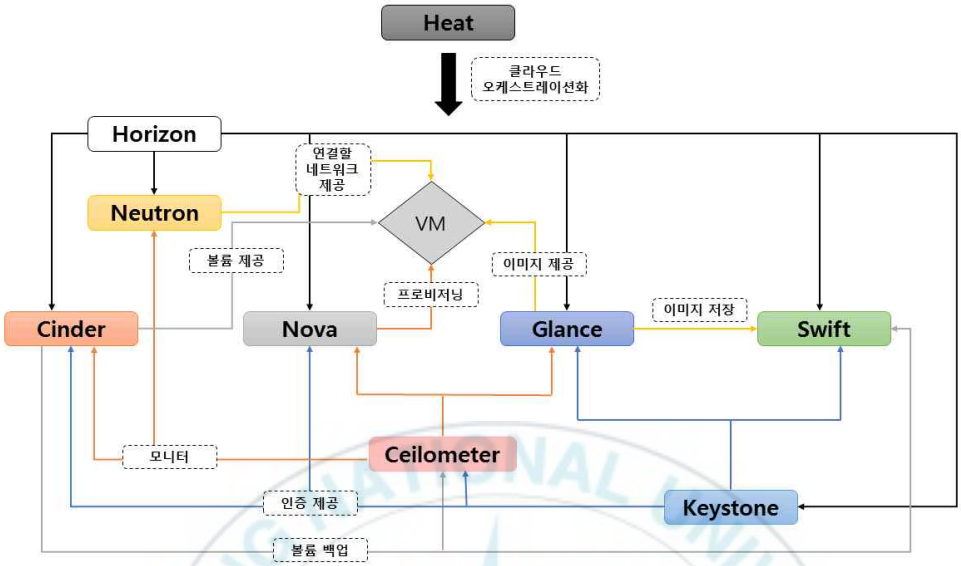


그림 4 오픈스택 구성요소 구조

오픈스택의 대표적인 프로젝트로는 가상머신의 생명주기를 관리하는 OpenStack Compute(Nova), 정형화되지 않은 데이터 객체를 저장하고 조회하는 Object Storage(Swift), 가상머신 디스크를 저장하고 조회하는 Image Service(Glance), 보안인증을 담당하는 Identity(Keystone), 사용자 인터페이스 서비스를 제공하는 Dashboard(Horizen), 네트워크 연결을 가능하게 하는 Netwoking(Neutron), 블록 스토리지를 관리하는 Storage Service(Cinder), 배포된 자원의 사용량 및 성능을 모니터링 하는 Telemeter Service(Ceilometer) 등 다양한 프로젝트들이 오픈스택의 구성요소로써 개발 중이다[15].

IaaS 환경을 제공하는 오픈스택을 선택한 이유는 제한적인 클라우드 제어와 CSP에 대한 의존도를 줄이기 위함이다. 그림 5는 클라우드 서비스 모델 별 사용자의 제어 영역을 나타낸 것이다. 또한, IaaS에서는 디지털 포렌

식 관점에서 다른 모델에 비해 Virtual Introspection, Snapshot 등의 특징으로 더 많은 증거 데이터를 획득 할 수 있으며, 다른 서비스 모델에 비해 사용자의 제어 영역이 가장 넓기 때문에 CSP에 대한 의존도를 줄일 수 있다[1].

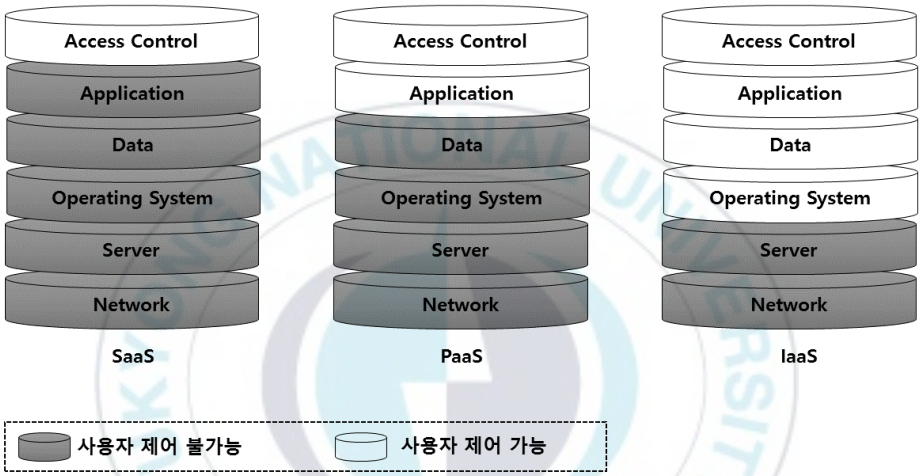


그림 5 클라우드 서비스 모델 별 사용자 제어 영역

Ⅲ. 클라우드 컴퓨팅 플랫폼을 고려한 포렌식 절차

3.1 클라우드 환경에서 획득 가능한 증거 데이터 분석

가. 클라우드 계층에 따른 증거 데이터 분류

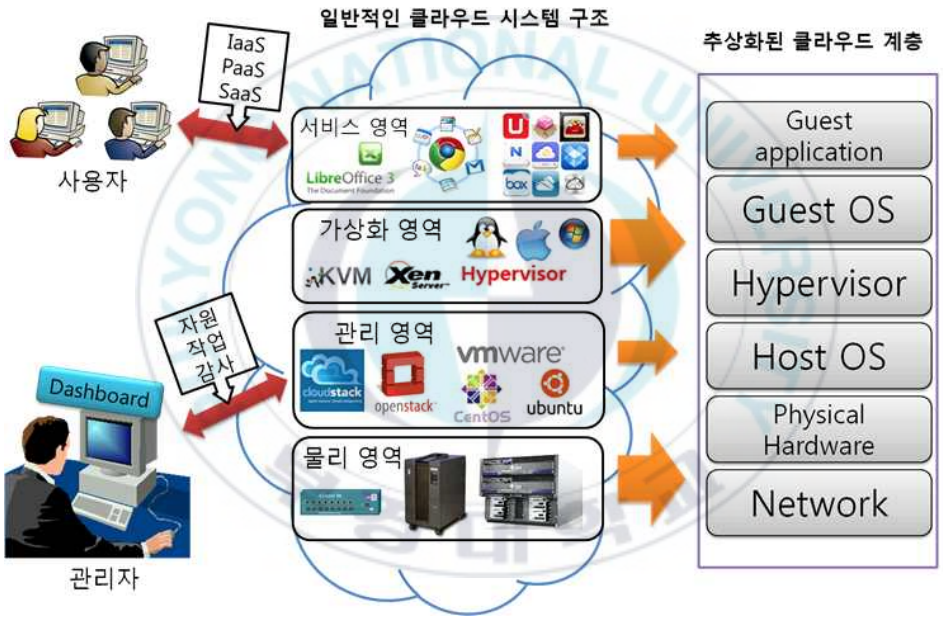


그림 6 일반적인 클라우드 시스템 구조와 추상화된 클라우드 계층

그림 6은 일반적인 클라우드 컴퓨팅 구조와 그에 따른 추상화된 클라우드 계층을 보여준다. 또한, 추상화된 각 계층은 앞서 언급했던 J. Dykstra and A. T. Sherman(2012)[6]에서 제시한 신뢰모델을 참고하였다. 따라서 본 논문에서 언급되는 신뢰는 수집된 증거가 법원에 제출되어 증거로써 믿을 수 있고 가치를 가질 수 있느냐에 대한 의미로 사용된다. 각 계층에서는 각

각 다른 포렌식 수집 활동이 이루어지며, 계층에 대한 신뢰성과 안정성의 정도가 다르게 요구되기 때문에 계층을 나누어서 분석해야 하고 상위로 올라 갈수록 누적된 신뢰가 필요하다. 공용 클라우드에서 모든 계층은 제공자에게 일부 신뢰를 요구하는데 특히, 악의적인 내부자에 대비한 신뢰를 요구한다[6]. 표 4는 그림 6에서 추상화된 클라우드 계층 별 획득 가능한 데이터를 정리하여 나타낸 것이다. 예를 들어 게스트 OS에서 데이터를 획득할 시 아래 계층의 데이터에 대한 신뢰가 요구된다.

표 4 클라우드 계층 별 수집 가능한 데이터

클라우드 환경에서 수집 가능한 데이터	
사용자 어플리케이션 (Guest application)	사용자 이벤트 등
사용자 운영체체 (Guest OS)	OS 할당 메모리 영역 덤프, 설치된 S/W 목록 등
하이퍼바이저 (Hypervisor)	시스템 지원 할당, VM 사용 시간, 스냅샷
호스트 운영체제 (Host OS)	클라우드 인프라의 중앙 제어에 관한 구성요소, 서비스 사용, 권한 등의 클라우드 플랫폼에서 제공하는 정보
물리적 하드웨어 (Physical Hardware)	파일시스템, 각 종 로그 파일 등
네트워크 (Network)	네트워크 연결, 메모리 덤프, 실행 중 프로세스, 로그온 사용자 등

▪ **네트워크(Network)**

네트워크 자원은 물리적 또는 가상이 될 수 있는데 이러한 자원들은 사용자 간에 공유가 될 수 있다. IaaS에서는 호스트 시스템의 네트워크 카드는 여러 가상머신에 의해 이용되고, 다양한 사용자를 포함하고 있다. 특히, 클라우드 컴퓨팅과 같은 복잡한 모델에서는 다양한 목적을 가진 사용자들이

참여하기 때문에 특정 사용자에 의해 이용되는 네트워크 자원을 모니터링 할 수 있어야 한다. 각 사용자의 트래픽을 구별하는 것은 책임추적성에 대한 핵심적인 문제이며, 의심되는 사용자 또는 악의적인 사용자의 흔적이 증거로 사용되는 중요한 요소이다.

▪ **물리적인 하드웨어(Physical hardware)**

원본 하드디스크 확보는 디스크 복제를 통해 저장된 데이터의 수집 및 분석하고 원본을 보존할 필요가 있을 때 수행된다. 이 때 획득할 수 있는 데이터는 파일 시스템의 메타데이터(마지막 수정 시간, 마지막 접근 시간, 생성시간, 변경 상태 등), 시스템과 응용 프로그램의 로그를 확인 할 수 있는데 여기에는 에러 로그, 설치 로그, 네트워크 연결 로그, 보안 로그 등이 있다[2].

▪ **호스트 운영체제(Host OS)**

Host OS에는 클라우드 인프라의 중앙제어에 관한 구성요소, 클라우드 서비스 사용, 접근권한, 환경구성, 자원 프로비저닝, 정책, 사용자 로그인 등의 정보를 제공하는 클라우드 플랫폼이 포함되어 있으며 클라우드 관리 시스템이라는 용어로도 사용된다. 가상 디스크에 대한 접근으로 데이터를 획득할 수 있고, 클라우드 플랫폼에서 제공하는 로그파일과 가상머신의 스냅샷은 관심 있게 볼 수 있는 정보이다. 또한 관리자는 클라우드 플랫폼을 통해서 자원관리, 작업관리, 감사 등을 수행할 수 있다[4].

▪ **하이퍼바이저(Hypervisor)**

하이퍼바이저에서 데이터의 사용은 Intrusion Detection Systems(IDS)의 동작을 통해 증거를 획득할 수 있다[16]. 이러한 조사는 하이퍼바이저에 대한 액세스 권한이 필요하기 때문에 IaaS 클라우드 조사에 적합하다.

▪ **사용자 운영체제/어플리케이션(Guest OS/Application)**

인스턴스 내부에서 정보를 얻기 위해서는 원격 포렌식 소프트웨어를 통해

증거를 획득하는 방법이 있으며, 인스턴스 내부에 추가로 소프트웨어를 설치해서 내부정보를 획득할 수 있다.

어플리케이션에서는 어플리케이션 로그, 인증 로그를 얻을 수 있으며 멀티테넌트 로그 데이터에 대해서는 다중 리소스로부터 분리와 병합이 함께 되어야 한다.

나. 클라우드 환경에서 데이터 수집의 한계점

클라우드 컴퓨팅 환경에서 보안사고 또는 범죄가 발생하여 포렌식 수사를 수행할 때, 증거 데이터를 확보하고 수집하는 것에 어려움이 따른다. 데이터가 국제적으로 분산되어 있다면 이것은 기술적인 문제가 아니라 법적인 문제가 발생한다. 또한, CSP에서 클라우드 포렌식을 위한 로그 파일이나 데이터를 저장 하지 않거나 사용자와 다른 타임스탬프를 가진다면 증거로 채택하기 어렵다. 특히, 데이터 수집에서의 취약점은 포렌식 수사과정과 증거 획득 여부에 결정적인 영향을 끼치기 때문에 취약점을 분석하고 이에 따른 해결방안에 대한 연구가 필요하다.

표 5는 그림 6과 표 4에서 추상화된 클라우드 계층을 분석한 것을 바탕으로 각 계층에 따른 데이터 수집의 취약점을 표를 통해 정리한 것이다. 클라우드 컴퓨팅 플랫폼은 서버, 스토리지, 네트워크와 같은 물리 자원을 가상화 하여 논리적으로 구성할 수 있도록 하며, 스토리지는 논리적이며 할당된 공간에 초점이 맞추고 있기 때문에 물리적 접근과는 다르게 데이터를 확보해야 한다. 이러한 점은 물리적 디스크를 압수하는 일반적인 포렌식 수사와는 달리 복잡한 과정을 거쳐야 하며, 신속하게 획득해야 하는 휘발성 데이터에 대한 수집을 어렵게 하고 포렌식 수사의 지연을 발생시킨다[4].

표 5 클라우드 계층에 따른 데이터 수집 취약점

Cloud Layer	취약점
사용자 어플리케이션 (Guest application)	(1) 사용자 이벤트가 공급자 측에 위치 (2) 휘발성 데이터
사용자 운영체제 (Guest OS)	(1) 스냅샷의 보존 문제
하이퍼바이저 (Hypervisor)	(1) 종류에 따라 데이터에 대한 보존과 형식이 다양
호스트 운영체제 (Host OS)	(1) 공용 클라우드는 라이브 포렌식 및 휘발성 데이터에 대한 액세스를 허용하지 않음 (2) 클라우드 플랫폼에 따라 로그파일 및 리소스가 분산
물리적 하드웨어 (Hardware)	(1) 서버 시스템의 특성상 하드웨어를 압수할 수 없는 경우가 대부분이기 때문에 원본 획득이 어려움.
네트워크 (Network)	(1) 기존의 네트워크 장치 또는 모니터링 솔루션은 multi-tenant환경에서 증거를 획득하는데 어려움.

이러한 취약점들은 수사과정에서 매우 중요한 영향을 끼치고 있으며, 증거 획득 여부에 결정적인 역할을 하고 있다. 그러므로 이러한 취약점에 대한 지속적인 연구와 이에 대응하는 해결방안에 관한 연구도 함께 이루어져야 한다.

3.2 클라우드 컴퓨팅 플랫폼을 고려한 포렌식 절차

클라우드 환경에서의 기존 디지털 포렌식 수사는 앞서 살펴본 바와 같이 데이터 수집에 따른 취약점이 존재한다. 이러한 취약점을 해결하기 위해 먼저 법정에서 증거 데이터에 대한 증거 효력이 유지 될 수 있는 클라우드 포렌식 절차를 제안한다. 이를 위해 가상 자원들을 서비스 화하여 관리 체계를 제공하는 클라우드 플랫폼을 이용해서 접근한다. 클라우드 플랫폼을 이용한 접근은 J. Dykstra and A. T. Sherman(2012)[6]에서도 언급한 바와 같이 하위계층부터 누적된 신뢰를 충족시키기 위해 클라우드 플랫폼을 고려한 포렌식 절차를 통해 증거 데이터의 신뢰성을 향상시키고자 한다.

클라우드 플랫폼에서 모니터링을 통해 각 사용자에 대한 네트워크 트래픽을 구별하여 이를 통해 특정 사용자에 관한 증거를 획득하고 데이터를 통합 관리하는 데이터베이스에 수집하고 보존한다면 분산되어 있는 데이터에 대한 수집의 한계를 해결할 수 있다. 또한, 영구저장소의 부재에 따른 보존 문제와 사용자의 조작, 부팅 또는 강제 종료에 의한 데이터 손실에 대해서는 클라우드 플랫폼에서 영구 저장소에 스냅샷을 보존하여 데이터 손실이 발생하는 것을 방지한다. 사용자 이벤트, S/W 목록과 같은 데이터에 대한 수집은 공인된 포렌식 도구가 설치된 OS를 사용자에게 제공해서 데이터를 수집할 수 있다[4].

위에서 언급한 방안들을 반영한 포렌식 절차는 그림 7과 같이 도식화 할 수 있다. 먼저 클라우드 구성요소 식별을 통해 식별대상을 얻고, 식별된 정보를 통해 물리 영역 또는 가상화 영역을 선정한다. 물리 영역은 Host OS 수준으로 일반적인 OS와 마찬가지로, 기존 포렌식 도구를 이용하여 활성, 비활성에 따라 수집 대상을 수집한다. 가상화 영역의 경우 클라우드 플랫

폼 수준에서 데이터의 종류는 자원과 사용자로 나뉘며, 자원과 관련된 데이터는 시스템 자원 할당, VM 사용 시간, 스냅샷 이미지를 수집할 수 있으며 이러한 정보와 클라우드 플랫폼의 사용자 관리 정보를 통해 사용자 연관 정보를 수집 할 수 있다. 사용자에 대한 데이터는 사용자에게 공인된 포렌식 도구를 설치한 OS를 제공하여 OS할당 메모리 영역 덤프, 설치된 S/W 목록, 사용자 이벤트를 수집할 수 있다[4].

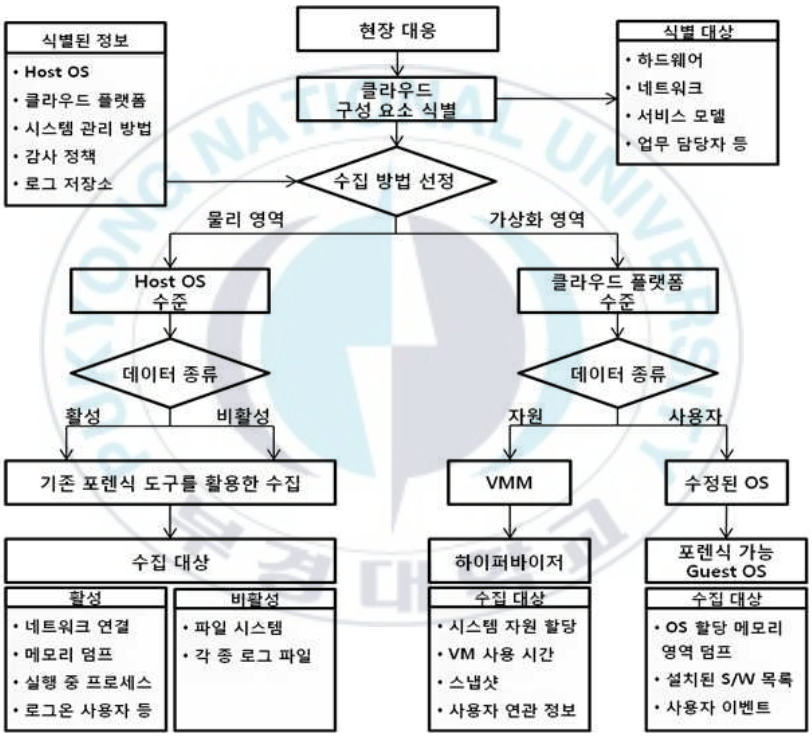


그림 7 클라우드 플랫폼을 고려한 포렌식 절차

다양한 클라우드 환경에 유연하게 적용할 수 있는 디지털 증거 수집 절차를 제안하기 위해 추상화된 클라우드 계층에 따라 수집 가능한 데이터를 분류하고, 확보한 증거 데이터의 신뢰성 보장을 위해 클라우드 플랫폼 기반의 데이터 수집 절차를 도식화하여 제안했다.

Ⅳ. 오픈스택 기반 클라우드 환경에서의 디지털 포렌식 분석

4.1 오픈스택을 이용한 클라우드 컴퓨팅 분석

오픈스택은 오픈소스 클라우드 플랫폼 중에 하나로 이제 막 4년을 넘긴 프로젝트이다. 그럼에도 불구하고 6개월에 한 번씩 새로운 버전을 릴리즈 하고 있으며 성능개선과 유지보수를 함께 진행하고 있다[15].

세계 최대 온라인 결제서비스 업체 '페이팔'은 최근 자사 핵심 인프라 중 트래픽 처리 영역과 중간 계층 서비스 운영기술을 오픈스택 기반으로 전환했다고 밝혔다. 또한, 미국에서 급부상한 클라우드 기반 인사관리(HR) 솔루션 업체 '워크데이'는 아마존웹서비스(AWS)에서 HP의 오픈스택 기반 퍼블릭 클라우드 서비스로 이전했다. 국내에서는 KBS에서 레드햇 엔터프라이즈 리눅스 오픈스택 플랫폼 기반 클라우드를 도입해 클라우드 시스템 구축을 추진하고 있다.

오픈스택 플랫폼을 이용하여 구현하는 것은 오픈스택이 오픈소스 클라우드 플랫폼에서 최근 가장 많은 성장과 많은 CSP가 오픈스택 기반 환경을 구축하기 때문이다. 특히, 비용과 효율성의 증대는 오픈스택의 가장 큰 강점이자 기업들이 오픈스택을 선택하는 이유이다.

따라서 오픈스택 기반 클라우드 환경에 대한 클라우드 포렌식의 문제점을 분석하고 오픈스택에 적용 가능한 클라우드 포렌식을 통해 CSP에 대한 의존성을 줄이고 증거에 대한 신뢰성을 높이고자 한다.

가. 오픈스택을 이용한 클라우드 컴퓨팅 구축

오픈스택을 이용하여 클라우드 환경을 구축하는 방법은 목적에 따라 다양하게 구성할 수 있다. 실제로 이 논문에서 구축된 환경은 그림 8과 같이 구성되어 있으며, 오픈스택 공식 홈페이지에서 설치 매뉴얼을 참고하여 4대의 노드를 구축하였다. 그림 8은 로그 수집을 위해 실제 구축한 환경을 각 노드별로 설치된 오픈스택 프로젝트와 함께 이해하기 쉽도록 도식화 하였다.

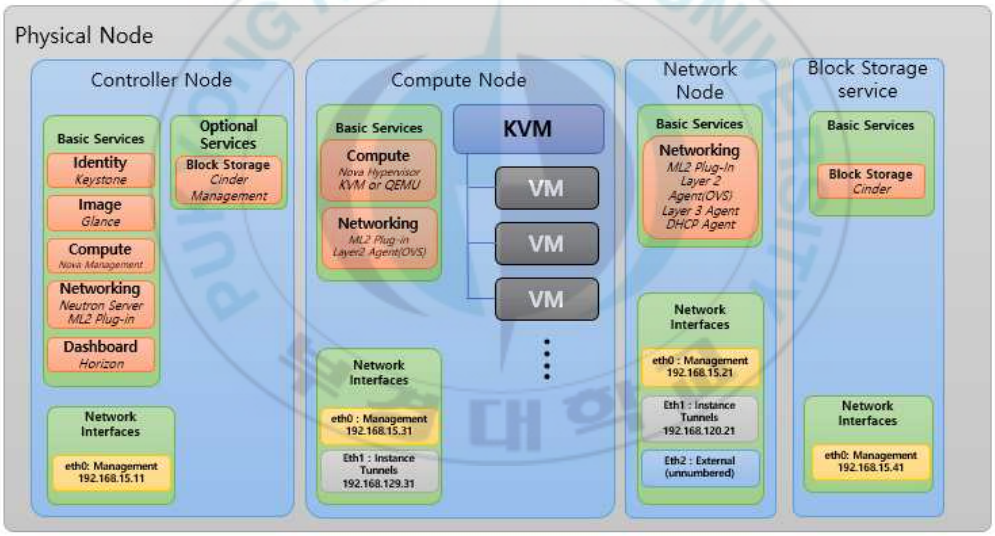


그림 8 구축한 오픈스택 노드 구성

Controller Node, Compute Node, Network Node, Block Storage Service는 물리적으로 분리되어 설치되어 있다. 먼저, Controller Node는 플랫폼 전체를 제어하는 역할과 인스턴스의 생명주기를 관리하는 Nova, 오픈스택의 서비스를 위한 인증과 권한부여를 제공하는 Keystone, 웹 기반 사용자

인터페이스를 제공하는 Horizon, 가상 디스크 이미지를 위한 저장소와 목록을 제공하는 Glance, 인스턴스에 영구적인 블록 저장소를 제공하는 Cinder 기능들이 수행된다. Compute Node는 다수로 구성될 수 있으며 서버 가상화 기능을 제공하는 Xen, KVM 과 같은 하이퍼바이저들이 설치되어 인스턴스들이 생성되어 실제로 수행되는 물리 서버들이다. 다수로 구성될 경우 마이그레이션이 수행될 수 있다. 마지막으로, Network Node는 오픈스택 내부의 가상 네트워크의 구성과 네트워크 서비스를 제공한다. Neutron에는 최근 이슈가 되고 있는 SDN의 근간이 되는 오픈플로우 기술을 지원한다[15].

구축환경은 Controller Node, Compute Node, Network Node, Block Storage Service 모두 IceHouse 버전을 사용하였고, Host OS는 Ubuntu 14.04 LTS 이다.

나. 오픈스택 기반 클라우드 컴퓨팅 환경에서의 로그 데이터 분석

클라우드 환경과 같이 서로 다른 프로세스에서 로그를 분석하는 것은 디지털 포렌식 수사에 중요한 역할을 한다. 로그는 능동적 보안과 지속적인 컴플라이언스 활동에 가치가 있지만 사고를 조사하고 대응하는 가치있는 정보의 출처이기도 하다. 그러나 클라우드 환경에서 이러한 로그 데이터를 수집하는 것은 일반적인 컴퓨터 환경과는 달리 접근이 어렵거나 불가능한 경우가 대부분이다. 클라우드 환경에서 로깅의 문제점은 아래와 같다.

▪ 분산

클라우드 인프라에서 로그데이터는 중앙 집중식 로그서버에 존재하지 않으며, 여러 서버 간에 분산되어 존재한다. 또한 여러 사용자의 로그 정보들

이 뒤섞여 배치될 수 있다.

▪ **휘발성**

가상머신의 경우 로그데이터는 휘발성을 가지고 있으며 가상 인스턴스를 사용자가 종료할 경우 모든 로그는 사라지게 되고, 증거 데이터로 사용할 수 없다. 따라서 이러한 로그는 특정 시간에만 접근 가능하며 사용할 수 있다[12].

▪ **다중 계층**

클라우드 구조는 여러 계층이 존재하며 각 계층마다 상이한 로그 데이터가 생성된다. 예를 들어 애플리케이션, 네트워크, 운영체제, 데이터베이스 등은 포렌식 조사에 유용한 로그를 생성한다. 이러한 여러 계층에서 신뢰할 수 있는 절차를 통해 획득하는 과정을 연구하는 것이 클라우드 포렌식의 과제이다.

▪ **로그의 접근성**

시스템 관리자, 포렌식 수사관, 개발자 등은 각 각 역할에 해당하는 접근 권한을 가져야 한다. 시스템관리자는 시스템 문제를 해결하기 위한 관련 로그가 필요하며, 개발자는 응용 프로그램의 버그를 수정하기 위한 로그가 필요하다. 포렌식 수사관은 수사에 도움이 될 수 있는 로그가 필요하며 이러한 로그 접근과 획득은 안전한 방법으로 이루어져야 한다. 이러한 과정을 위해 접근 제어 메커니즘을 이용한다.

▪ **CSP에 대한 의존성**

로그의 가용성은 클라우드 서비스 모델에 따라 달라진다. CSP가 로그를 제공하지 않는 한 SaaS에서 사용자는 자신의 시스템의 로그를 얻지 못한다. PaaS의 경우 사용자의 응용 프로그램 로그를 얻을 수 있다. 만약 네트워크 로그, 데이터베이스 로그, 운영체제 로그를 얻기 위해서는 CSP에 의존할 수밖에 없다. IaaS의 경우 사용자는 네트워크 또는 프로세스 로그를

얻을 수 없다[12].

▪ 로그 데이터의 표준 부재

로그 데이터의 표준 형식이 없는 것은 포렌식 입장에서 까다로운 문제를 야기한다. CSP마다 이기종 형식과 뚜렷한 정의가 없다면 로그 데이터를 수집하여도 증거 데이터로 쓰는데 한계가 있기 때문이다.

이러한 문제에도 불구하고 로그 데이터는 포렌식 수사에 중요한 역할을 한다. 따라서 오픈스택 환경에서 획득 가능한 로그 데이터를 분석하고 이를 바탕으로 해결방안을 제안한다[18].

표 6 클라우드 컴퓨팅 플랫폼 기반 수집 가능한 로그

Cloud Layer	로그 위치	획득 정보
사용자 어플리케이션	로깅 지원 시 해당 프로그램에 따름	어플리케이션 로그
사용자 운영체제	윈도우: 레지스트리 리눅스: /var/log/*	Guest OS 시스템 이벤트
하이퍼바이저	qemu : /var/log/libvirt/qemu/ /var/log/libvirt/libvirtd.log	Hypervisor 운영로그 및 생성 인스턴스 로그
호스트 운영체제	/var/log/nova /var/log/glance /var/log/cinder /var/log/keystone /var/log/apache2/ /var/log/syslog /var/log/cinder/cinder-volume.log /var/log/neutron	오픈스택의 각 프로젝트 로그
물리적 하드웨어	/var/log/dmesg	인식되는 하드웨어 정보 로그

표 6에서는 오픈스택 환경에서 로그데이터 수집 전에 클라우드 컴퓨팅 시

스템 구성 요소의 추상화된 계층에 따른 수집 가능한 로그들을 분류하고 각 계층에 따른 로그들의 위치를 정리하였다. 수집 이전에 로그 위치와 클라우드 계층에 따른 획득 정보를 정리함으로써, 로그수 집을 더 용이하게 한다. 또한, 오픈스택 기반의 로그 데이터 수집을 수행하므로, Host OS에서의 로그위치에 따른 수집이 수행된다.

표 7 OpenStack 로그 위치

노드타입	서비스	로그위치
클라우드 컨트롤러	nova-*	/var/log/nova
	glance-*	/var/log/glance
	cinder-*	/var/log/cinder
	keystone-*	/var/log/keystone
	neutron-*	/var/log/neutron
	horizon-*	/var/log/apache2/
모든 노드	misc(swift, dnsmasq)	/var/log/syslog
컴퓨트 노드	libvirt	/var/log/libvirt/libvirtd.log
	VM 인스턴스의 콘솔 (부팅 메시지)	/var/log/nova/instances/instance-<instance-id>/console.log
블록 스토리지 노드	cinder-volume	/var/log/cinder/cinder-volume.log

오픈스택 각 프로젝트에서 수집 가능한 여러 로그들 중 다음 일부 프로젝트의 로그를 수집하였으며, 각 로그는 다음 정보를 보유하고 있다. 표 7은 로그 수집을 위해 구축한 오픈스택에서 각 노드별 로그 위치를 보여준

다. 그림 9는 실제 구현된 오픈스택 기반 클라우드 환경에서 Controller, Network, Compute, Block Storage service 노드별 로그위치를 캡처하였다.

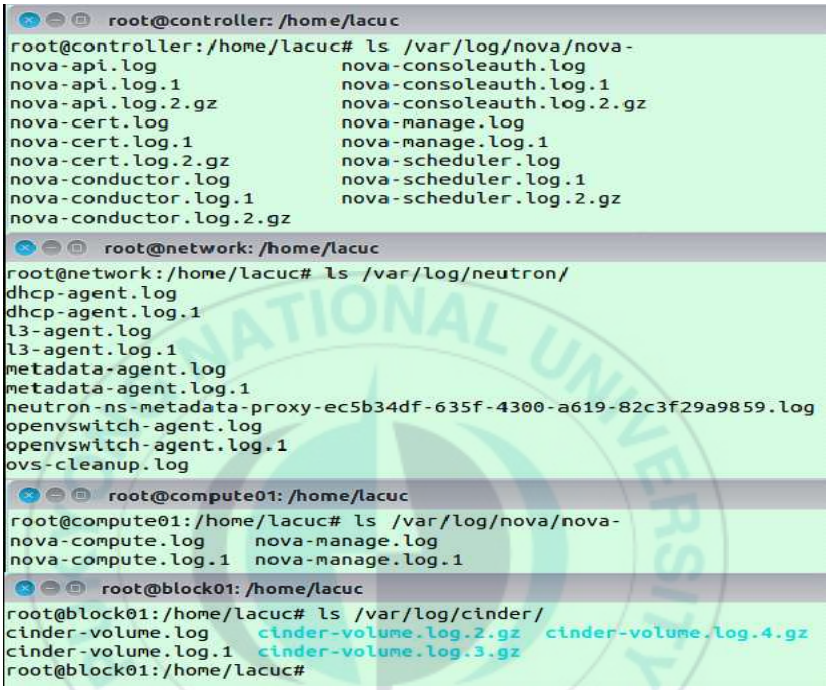


그림 9 각 노드의 오픈스택 로그 위치

▪ **컨트롤러(Controller)**

오픈스택과 사용자와의 상호작용과 오픈스택의 다른 구성요소와의 상호작용 메시지 항목을 포함하는 nova-api.log*, 노바 콘솔 서비스와 관련된 인증 세부 정보를 포함하는 nova-consoleauth.log*, nova-cert 프로세스에 관한 메시지 항목을 포함하는 nova-cert.log*, 노바 관리 명령어가 수행될 때의 메시지 항목을 포함하는 nova-manage.log*, 데이터베이스 정보에 대한 요청을 서비스에 대한 메시지 항목을 포함하는 nova-conductor.log*, 큐 공간에서 노드 작업 할당, 메시지, 일정에 관한 항목을 포함하는 nova-scheduler.log*가 있다[19].

▪ 네트워크(Network)

DHCP 에이전트에 관한 로그 항목을 포함하는 dhcp-agent.log*, l3 에이전트와 그 기능에 관한 메시지 항목을 포함하는 l3-agent.log*, 노바 메타데이터 서비스의 프록시인 뉴트론에 관련된 메시지 항목을 포함하는 metadata-agent.log*, open vswitch 작업에 관련된 메시지 항목을 포함하는 openvswitch-agent.log*, 가상 브릿지 br-int와 br-ex의 클린업 정보를 포함하는 ovs-cleanup.log*가 있다[19].

▪ 컴퓨트(Compute)

compute 노드의 리소스를 추적하고 해당 노드에서 생성되는 인스턴스의 정보를 포함하는 nova-compute.log*, 노바 관리 명령어가 수행될 때의 메시지 항목을 포함하는 nova-manage.log*가 있다[19].

▪ 블록 스토리지 서비스(Block Storage service)

cinder에서는 cinder-api 서비스에 대한 항목을 포함하는 로그와 cinder scheduling 서비스에 대한 cinder-scheduler.log*,
Cinder volume 서비스에 관한 항목을 포함하는 cinder-volume.log*가 있다[19].

수집되는 로그들은 오픈스택 프로젝트의 운용에 관한 정보들이며, 클라우드 플랫폼에서의 오류 또는 각 프로젝트 간의 연결에 대한 정보를 파악할 수 있다. 이러한 로그정보는 실제 클라우드 포렌식에서 IaaS를 제공하는 CSP의 운용 정보를 파악 할 때 사용하거나, 인스턴스의 생성에 관한 로그정보를 필요로 할 때 사용할 수 있다.

이러한 정보만으로는 최종 사용자가 게스트 OS 상에서 실행한 행위를 파악하기는 어렵다. 특히, 물리적으로 분산된 로그 파일을 찾고 획득하는 과정이 복잡하고 어렵기 때문에 실제 사건이 발생했을 때에도 사건과 관련된 로그를 파악하고 획득하는 것이 어려울 수 있다. 따라서 오픈스택 기반

클라우드 환경에서 클라우드 포렌식의 문제점을 분석하고 해결해야 한다.

4.2 오픈스택 기반 클라우드 포렌식의 한계 및 해결방안

클라우드 포렌식에서 가장 큰 취약점은 CSP의 의존성과 대부분의 CSP의 포렌식에 대한 인식 부족이다. 이러한 취약점을 해결하기 위해 클라우드 플랫폼 계층에서 포렌식 증거를 획득하고 저장해야 한다. 이러한 방법은 CSP에 대한 의존도를 줄이고, 증거 데이터의 신뢰성을 높여준다[12].

그러나, 오픈스택 플랫폼에서 제공하는 로그 데이터 혹은 서비스만으로는 포렌식에 필요한 증거 데이터를 획득하기가 어렵다. 또한, 오픈스택 플랫폼 기반 클라우드가 가지는 한계점은 앞서 살펴본 클라우드 포렌식의 문제와는 다른 세부적인 문제와 한계가 있을 수 있다. 따라서 오픈스택 플랫폼 기반 포렌식의 문제점을 살펴보고 이와 더불어 앞서 살펴본 클라우드 포렌식의 문제점을 해결할 수 있는 방안을 제안한다.

가. 오픈스택 기반 클라우드 포렌식 문제점 분석

▪ 휘발성 데이터

가상머신(VM)을 삭제하거나 전원을 끄는 경우 인스턴스 내부 데이터는 저장되지 않고 손실된다. 또한, 악의적인 사용자가 오픈스택의 가상머신을 사용하고 삭제했을 경우 영구 저장장치가 기본적으로 제공하지 않기 때문에 모든 데이터는 잃게 된다. 마찬가지로 사용자가 시스템을 종료할 때 가상환경 내에 저장될 레지스트리 또는 임시 파일은 손실될 것이다[1].

▪ **무결성**

사건이 발생하고 포렌식 수사관이 데이터를 수집할 때 데이터가 생성, 저장 그리고 획득하는 과정동안의 무결성을 보장할 수 없다. 가상머신과 이미지에 대한 체크섬을 따로 보관하지 않기 때문에 저장되고 획득하는 과정동안 조작, 손실 등이 발생한다면 이를 보장할 수 있는 절차가 부족하다.

▪ **중앙 집중 로그관리 부재**

오픈스택에서는 앞서 살펴본 것처럼 물리적으로 분산되어 있는 노드에서 다양한 로그 데이터가 생성된다. 또한 중앙로그서버를 기본적으로 제공하지 않고, 추가적인 설치와 복잡한 설정이 필요하다. 이러한 점은 사용자의 로그 정보를 파악하는 과정을 지연시키며 포렌식 수사를 어렵게 하는 원인이 된다[12].

▪ **연계보관성(Chain of custody)**

디지털 포렌식에서 증거는 획득되고, 이송/분석/보관/법정 제출이라는 일련의 과정이 명확해야 하고 이러한 과정은 포렌식 조사에서 가장 중요한 문제 중 하나이다. 기존의 포렌식 환경과는 다르게 클라우드 환경에서는 데이터가 가상 환경으로 부터 획득해야하기 때문에 원본이라는 개념이 모호하다[1]. 따라서 오픈스택 기반 클라우드 환경에서는 데이터의 생성과 저장단계에서부터 무결성 및 신뢰성을 보장할 수 있어야 한다. 하지만, 현재 오픈스택에서는 이러한 것을 보장해주는 구체적인 기능이 부족하다.

▪ **빈번한 릴리즈(Release)**

오픈스택은 현재(2015년 4월)를 기준으로 10번의 릴리즈가 있었다. 2010년 오스틴(Austin) 릴리즈를 시작으로 벡사(Bexer), 켄사추(Cactus), 디아블로(Diablo), 에섹스(Essex), 풀섬(Folsom), 그리즐리(Grizzly), 하바나(Havana), 아이스하우스(IceHouse), 주노(Juno)로 이어졌다[13]. 현재는 주노 버전에서 계속 업데이트되고 있으며 부가 서비스가 추가될 예정이다.

빈번한 릴리즈 혹은 업데이트는 클라우드 컴퓨팅 운용에 혼란을 주며, 포렌식 관점에서는 매번 버전에 맞는 서비스를 파악하고 제공되는 데이터 처리를 위한 작업을 수행해야하는 부담이 있다.

나. 클라우드 포렌식을 위한 오픈스택 기반 중앙로그관리

클라우드가 대부분 많은 서버로 구성되어 있기 때문에 필요한 로그데이터에 접근하기 위해서는 각 서버의 로그를 점검해야 한다[22]. 이는 관리적, 포렌식 관점에서 보았을 때 로그의 접근성이 떨어진다. 또한, 각 서버의 로컬 환경에만 로그데이터를 저장할 경우 악의적인 사용자가 삭제 혹은 변경한다면 로그데이터에 대한 획득이 불가능하다. 보안적인 측면에서도 데이터의 위·변조 방지를 통한 데이터 무결성 유지, 실시간 로그분석을 통한 보안사고 탐지/예방, 개인정보 등 데이터 보관에 대한 법적/제도적 규제 대응을 위한 방안으로 통합로그 관리가 요구된다. 특히 최근 국내에서 대형 개인정보 유출사건과 내부정보 유출사건에 대한 종합적인 보안대책의 일환으로 중앙집중형 로그관리의 필요성이 대두되었다. 중앙로그서버를 구축하는 방안은 매우 다양하며, 도구 및 적용기술에 따라 중앙로그관리 기능뿐만 아니라 웹UI에서 실시간 검색, 모니터링, 사용자의 모든 커맨드 확인 등의 서비스가 가능하다. 실제 중앙로그서버를 구축하는 것은 어떤 로그를 수집하느냐에 따라 통합로그관리시스템을 달리 적용할 수 있다.

특히 클라우드 환경처럼 분산된 머신으로부터 로그를 수집하는 경우 로그 수집기 도구 (Log Collectors Tool)를 이용할 수 있으며 최근 클라우드 시스템이 확대되면서 필요성이 크게 증가하였다.

표 8 로그 수집기 도구 및 설명

로그 수집기 도구 (Log Collectors Tools)	설명
Flume	Apache의 Top-Level Project, 많은 양의 로그 데이터의 수집, 취합, 전달 등을 위한 로그 수집기 도구이며 모든 데이터는 HDFS에 저장된다.
Logstash	모든 소스로부터 로그를 파싱, 인덱싱 할 수 있으며 Elasticsearch family가 되면서 로그를 웹으로 보여주는 kibana와 함께 쓰이면서 액세스 및 로그를 검색하기 위한 인터페이스를 제공한다.
Fluentd	ruby와 C로 짜여진 로그 수집기 시스템이다. 기본적인 구조는 flume와 비슷하며, 가장 큰 특징은 각 파트별 plugin을 쉽게 만들 수 있다.
Splunk	비정형 데이터를 분석하는 솔루션으로 보안/관제/모니터링 쪽에서 기능성과 유연성으로 떠오르는 솔루션이다. 하지만 상용 프로그램으로써 비용측면에서 높은 가격이 단점이다.

다른 방법으로는 우분투에서 디폴트 로깅 서비스로 rsyslog를 사용하며 기본적으로 원격 위치에 로그를 전송할 수 있기 때문에 따로 다른 것을 설치할 필요가 없다[22]. 또한, 가로채기를 방지하는 암호화된 VPN을 사용하거나 관리 네트워크상에서 로깅을 실행하는 것을 고려한다[22]. 이 논문에서는 위와 같은 방법으로 중앙집중형 로그관리서버를 구축하였다.

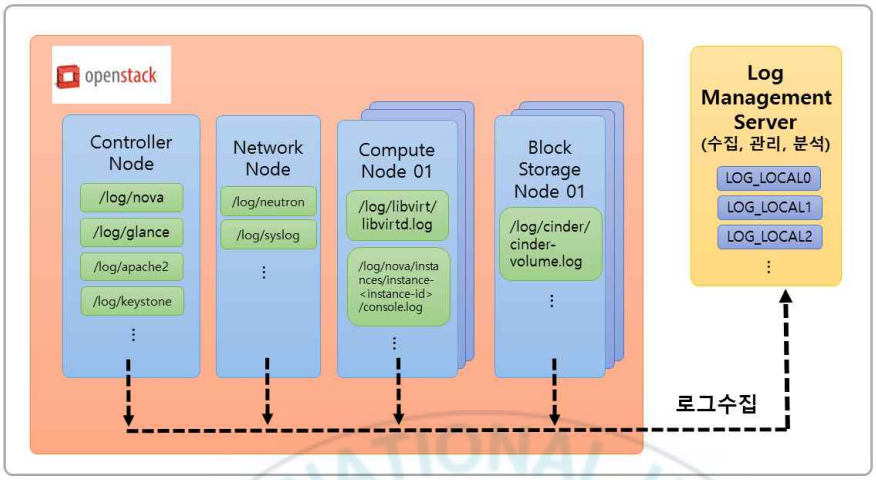


그림 10 rsyslog를 이용한 중앙로그관리 시스템

그림 10은 rsyslog를 이용한 중앙로그관리 시스템을 도식화하여 나타낸 것이다. 표준 로그 파일 장소에 추가하여 syslog에 로그하도록 모든 오픈스택 컴포넌트들을 구성하고 다른 syslog 설비를 사용하여 로그하도록 각 컴포넌트를 구성한다(그림 10의 LOG_LOCAL0,1,2 등). 중앙로그서버에서 컴포넌트 별로 개별적으로 분산하여 로그를 관리하는 것이 더 쉬워진다[22].

원격시스템에서 서버로 로그를 보내는 것은 비교적 간단하게 설정할 수 있지만, 서버에서는 로그를 파일에 저장할 것인지 DB에 저장할 것인지 설정해야하며 저장된 내용을 어떻게 보여주고 관리할지 복잡한 설정이 필요하다. 또한, 원격시스템들의 시간이 일치하지 않는다면, 한 곳에 모아진 로그의 의미를 이상하게 만들 수도 있다[24]. 따라서 정확한 시간 동기화를 통해 중앙로그에 보내지는 로그의 타임라인을 맞추는 것이 중요하다.

본 논문에서는 Log Analyzer를 이용하여 중앙로그관리 시스템을 구성하였다. Log Analyzer를 개발하고 있는 Adiscon사는 RFC 5424(The syslog

protocol)을 제출한 회사이며 syslog 전문회사로 웹에서 syslog를 분석할 수 있는 툴을 제공하며 rsyslog에서 서버로 전송되는 로그는 mysql를 통해 DB에 저장한다.

그림 11은 중앙로그서버에 설치된 Log Analyzer를 이용하여 웹에서 controller syslog만을 따로 확인하는 창을 캡처한 것이다. 노드별 syslog를 따로 관리하고 모니터링 할 수 있다는 편리성이 있다. facility는 메시지를 발생시킨 프로그램의 타입을 나타내는 값이며, Severity는 메시지의 성격 또는 중요도를 나타낸다[24].

그림 12와 그림 13은 Controller의 syslog를 XML문서와 CSV문서의 형태로 저장되는 것을 보여주며 이러한 기능은 Log Analyzer에서 제공하는 기능이며 이 외에도 사용날짜, Severity별 저장 개수, syslogtag별 메시지 개수 등을 통계적으로 그림으로 제공하고 검색과 관련된 다양한 기능을 제공하고 있다.

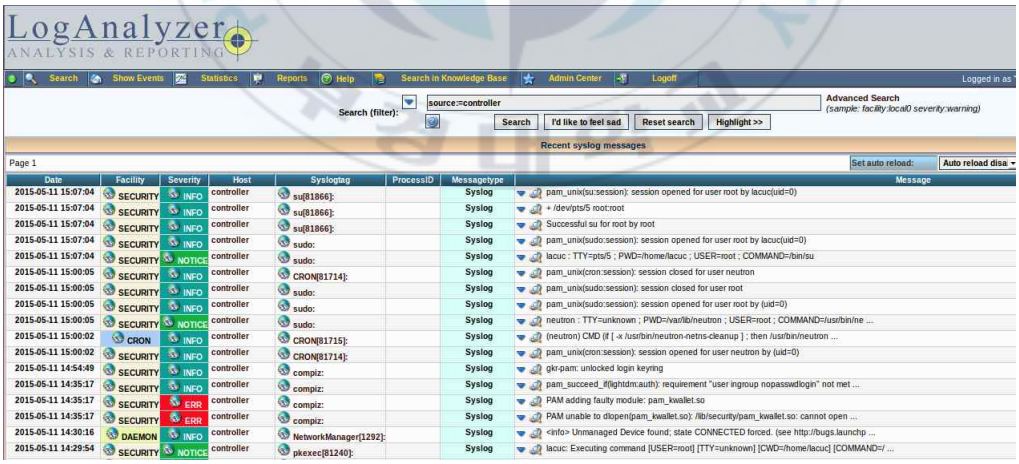
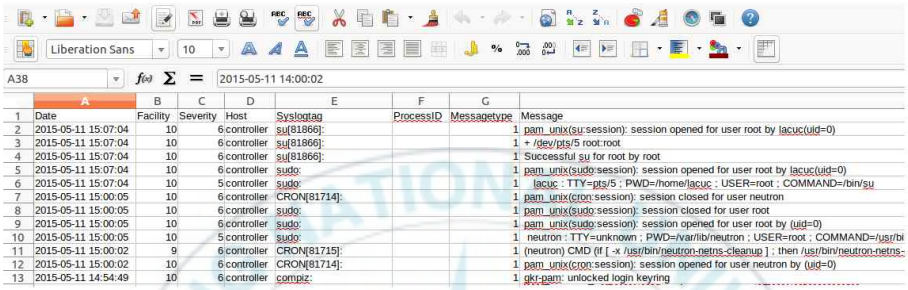


그림 11 Log Analyzer에서 controller syslog 목록

```
−<syslogmessages>
 −<syslogmsg>
    <timereported>2015-05-11 15:07:04</timereported>
    <syslogfacility>10</syslogfacility>
    <syslogseverity>6</syslogseverity>
    <FROMHOST>controller</FROMHOST>
    <syslogtag>su[81866]:</syslogtag>
    <procid/>
    <IUT>1</IUT>
   −<msg>
     pam_unix(su:session): session opened for user root by lacuc(uid=0)
    </msg>
```

	A	B	C	D	E	F	G	
1	Date	Facility	Severity	Host	Syslogtag	ProcessID	Messagetype	Message
2	2015-05-11 15:07:04	10	6	controller	su[81866]:		1	pam_unix(su:session): session opened for user root by lacuc(uid=0)
3	2015-05-11 15:07:04	10	6	controller	su[81866]:		1	+ /dev/pts/5 root:root
4	2015-05-11 15:07:04	10	6	controller	su[81866]:		1	Successful su for root by root
5	2015-05-11 15:07:04	10	6	controller	sudo:		1	pam_unix(sudo:session): session opened for user root by lacuc(uid=0)
6	2015-05-11 15:07:04	10	5	controller	sudo:		1	lacuc : TTY=pts/5 ; PWD=/home/lacuc ; USER=root ; COMMAND=/bin/su
7	2015-05-11 15:00:05	10	6	controller	CRON[81714]:		1	pam_unix(cron:session): session closed for user neutron
8	2015-05-11 15:00:05	10	6	controller	sudo:		1	pam_unix(sudo:session): session closed for user root
9	2015-05-11 15:00:05	10	6	controller	sudo:		1	pam_unix(sudo:session): session opened for user root by (uid=0)
10	2015-05-11 15:00:05	10	5	controller	sudo:		1	neutron : TTY=unknown ; PWD=/var/lib/neutron ; USER=root ; COMMAND=/usr/bi
11	2015-05-11 15:00:02	9	6	controller	CRON[81715]:		1	(neutron) CMD (if [-x /usr/bin/neutron-netns-cleanup] ; then /usr/bin/neutron-netns-
12	2015-05-11 15:00:02	10	6	controller	CRON[81714]:		1	pam_unix(cron:session): session opened for user neutron by (uid=0)
13	2015-05-11 14:54:49	10	6	controller	compiz:		1	gkr-pam: unlocked login keyring

그림 12 controller syslog에 대한 XML파일과 CSV파일

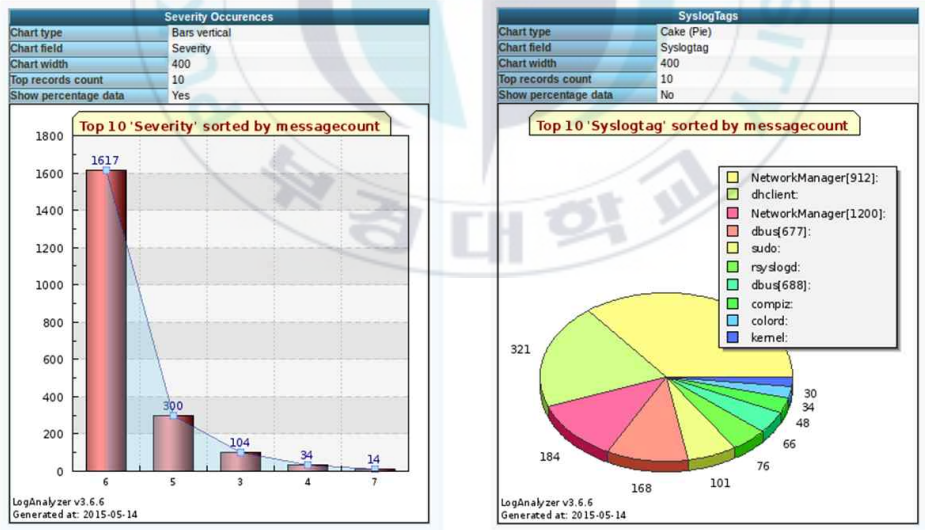


그림 13 Severity별 개수와 Syslogtag에 대한 개수

다. 오픈스택 기반 증거데이터의 신뢰성 향상을 위한 절차

오픈스택은 IaaS 플랫폼으로써 사용자에게 가상 머신을 제공하며, 클라이언트 시스템에서 원격을 통해 클라우드에 존재하는 가상 머신에 접속하여 컴퓨팅 자원, 스토리지 자원, 네트워크 자원 등 최근에는 데이터베이스 서비스를 비롯한 다양한 인프라 서비스를 제공한다.

그러나 오픈스택에서는 가상 머신을 영구 스토리지에 저장하지 않으며 가상 머신에 대한 신뢰성을 보장할 만한 데이터를 따로 저장하지 않기 때문에 가상 머신을 획득하는 것이 어려우며, 획득하여도 법정에서 증거로써 보장 받을 절차가 부족하다. 따라서 가상 머신의 저장과 수집, 조사 과정에서 신뢰성 향상을 위한 추가적인 방안을 제안한다.

구축한 클라우드 환경에서 가상머신 수집은 Compute Node에서 구동되고 있는 가상머신을 스냅샷을 통해 Controller Node에서 이미지를 관리하는 Glace를 통해 저장된다[20].

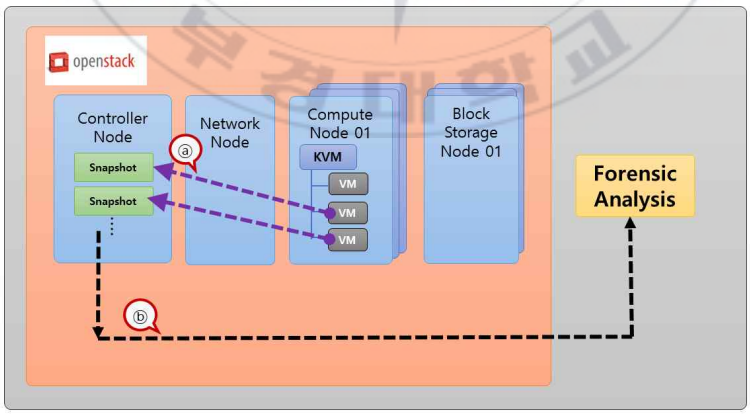


그림 14 오픈스택 기반 클라우드 환경에서 이미지 수집

그림 14는 오픈스택 환경에서 사용자의 가상머신이 스냅샷을 통해 저장되고, 저장된 이미지를 증거로 획득하는 과정을 나타낸 것이다. 그러나 이러한 과정은 앞서 언급하였듯이 획득된 증거에 대한 무결성 및 신뢰성 문제가 제기된다.

a과정에서 저장되는 이미지는 Snapshot을 이용하여 가상 머신을 획득하는 방법을 고려한 것이며, 저장된 이미지에 대한 신뢰성을 향상시킬 필요성이 있다. 또한, 이미지가 저장되는 순간부터 이미지에 대한 무결성이 보장이 되어야 한다[20].

이러한 문제를 해결하기 위해 오픈스택 플랫폼 계층에서 가상 머신을 저장하고 보관하여 신뢰계층 관점에서 생성과 저장, 보관, 획득을 같은 계층에서 수행한다. 또한, 스냅샷을 통해 가상 머신이 저장 될 때 이미지에 대한 해시 값, 타임 라인, 사용자에 대한 id 등 가상 머신의 정보를 저장한다.

그림 15는 해결방안으로 제안한 과정을 도식화하여 나타낸 것이다. 현재 오픈스택에서 스냅샷을 통해 저장할 때, 이미지 이외에 저장되는 정보 또는 해시 값이 없으며, 로그가 저장되더라도 분산되어 저장되기 때문에 삭제나 수정과 같은 문제가 발생할 수 있다[20]. 이러한 점은 앞서 말한 증거에 대한 신뢰성 문제가 발생하는 이유가 된다. 클라우드 포렌식에서 증거 데이터에 대한 정의 및 요구사항이 명확하게 명시하지 않기 때문에, 획득된 데이터에 대한 시간 정보, 이미지에 관한 로그데이터 및 사용자 이름과 해시 값을 추가한 정보를 XML문서로 보관하여, 증거에 대한 신뢰성을 향상시킨다.

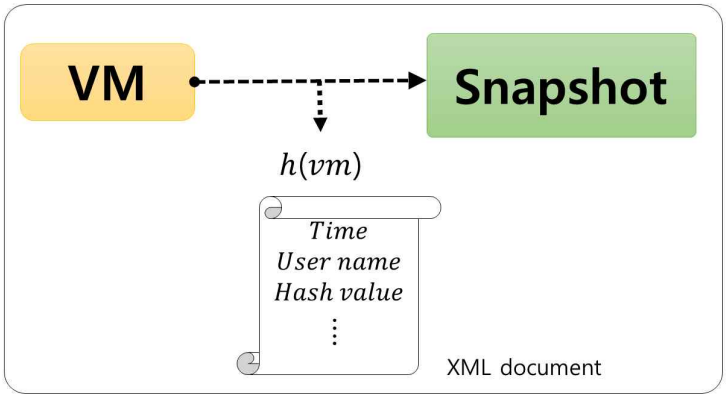


그림 15 가상 머신이 저장될 때
신뢰성 향상을 위해 추가되는 과정

그림 16에서 이미지를 획득할 때, 포렌식 조사관의 전자서명 과정 $sig_{SK_{IV}}(hash(snapshot))$을 통해 서명 값을 보관한다. 이러한 과정은 일반적인 디지털 포렌식의 물리적인 디바이스와는 다르게 클라우드 환경에서는 물리적인 접근이 되지 않기 때문에 포렌식의 연계 보관성과 증거의 무결성과 신뢰성이 과정의 단계마다 지켜져야 한다. 따라서 증거 획득 시 조사관의 개인키를 이용하여 전자서명을 함으로써 증거의 무결성과 연계보관성을 유지할 수 있다.

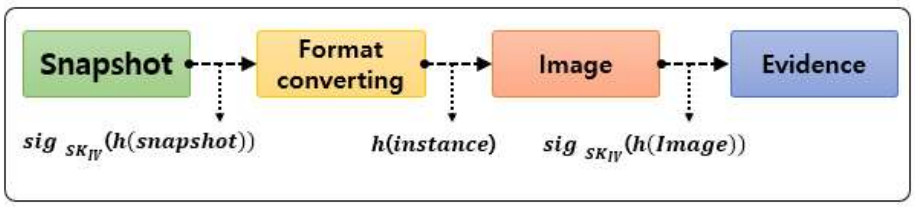


그림 16 무결성을 보장하기 위한 수집한 증거데이터 처리 과정

또한, 오픈스택은 지원하는 하이퍼바이저의 종류가 다양하며, 하이퍼바이저들은 서로 다른 VM이미지를 지원한다. 이미지 포맷에는 어떤 VM을 사용하는지에 대한 정보를 포함하고 있으며, 오픈스택에서 지원하는 VM 이미지 종류는 raw, rhd, vmdk, iso, qcow2, aki, ari, ami 가 있다. 대체적으로 이미지를 내려받아 Glance에 등록할 때는 raw 포맷이나 qcow2 포맷으로 등록하지만, 스냅샷 과정에 걸리는 시간을 고려하면 qcow2 포맷을 사용하는 것이 효율적이다. qcow2 이미지 포맷은 QEMU 에뮬레이터에 의해 지원되는 포맷으로, 동적으로 확장 할 수 있고 Copy on Write를 지원한다[13].

하지만, qcow2 포맷을 사용했을 때는 그림 4에서 나타낸 것처럼 포맷 변환과정을 거쳐야 한다. 이러한 변환과정에서 이미지에 대한 무결성은 반드시 지켜야하기 때문에 해시함수$h(instance)$를 통해 해시 값을 보관하여, 이미지 데이터에 대한 무결성을 유지한다[20].

마지막으로 이미지는 분석하는 포렌식 도구에 맞게 변환과정을 끝내고 분석 도구를 통해 VM에 대한 분석 전에, 포렌식 조사관에 의해 개인키로 서명을 하고 난 후$Sig_{SK_{IV}}(h(Image))$에 분석을 진행한다. 이러한 과정은 조사과정 중에 쉽게 수정 또는 유실될 가능성이 있기 때문에, 이러한 위험에 대비하여 증거에 대한 신뢰성과 무결성을 좀 더 명확하게 지켜져야 한다.

4.3 해결방안을 적용한 클라우드 포렌식

일반적으로 IaaS 유형의 경우 CSP는 데이터 센터의 서버에 가상 머신을 생성하여 고객에게 서비스를 제공하게 된다. 따라서 클라우드 컴퓨팅 환경의 디지털 포렌식 조사 과정은 기본적으로 가상 머신 포렌식 기술을 기반으로 수행되게 된다. 4.1과 4.2에서 오픈스택 기반 클라우드 환경을 구축하고 각 로그데이터 및 문제점을 분석하고 이에 따른 해결방안과 추가적인 절차를 제안하였다. 이 절에서는 제안된 절차를 바탕으로 실제 오픈스택 클라우드 환경에서 가상 머신을 이용하여 클라우드 포렌식 문제점을 고려한 오픈스택 환경에서의 디지털 포렌식 조사를 수행한다.

가. Snapshot을 이용한 증거 수집

Snapshot은 한 번의 클릭으로 실행 중인 시스템의 메모리도 함께 포함하여 복제할 수 있는 기능으로 포렌식 조사 과정에 얻은 Snapshot을 통해 현재 로그인된 사용자, 열린 포트, 구동 중인 프로세스, 시스템 그리고 레지스트리 정보 등 활성 및 비활성 데이터를 획득 할 수 있다[17].

오픈스택에서는 Snapshot 메커니즘을 통해 실행중인 인스턴스에서 새로운 이미지를 만들 수 있다. 대시보드를 통해 간단하게 스냅샷을 생성하는 방법과 커맨드라인을 통해 스냅샷을 생성하는 방법이 있다. 스냅샷에 대한 대시보드 인터페이스가 이미지와 스냅샷 페이지로 분할하였기 때문에 혼동할 수 있다. 하지만 이미지와 스냅샷의 유일한 차이점은 스냅샷에 의해 만들어진 이미지는 데이터베이스에 추가 속성을 가지고 있다는 것이다. 이러한 추가 속성은 image_properties 테이블에서 찾을 수 있다.

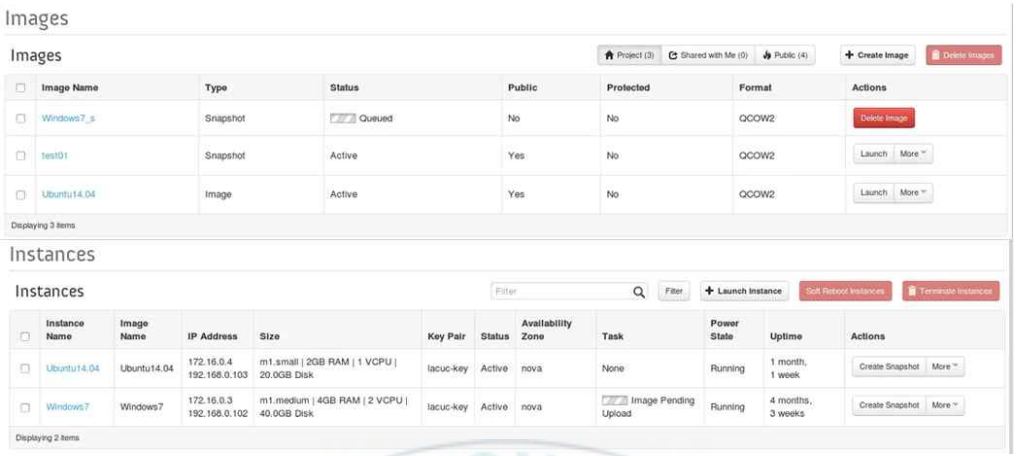


그림 17 Snapshot을 이용하여 인스턴스 스냅샷 이미지 생성

그림 17은 오픈스택 대시보드에서 Ubuntu14.04 가상 머신에 대한 스냅샷 이미지 생성하는 것을 나타낸다. 실행중인 가상 머신에서 스냅샷을 생성할 때 그림17에서 나타나듯이 Pending 작업을 진행하면서 Task 상태가 변하는 것을 알 수 있다. 오픈스택에서는 사용자들이 일시중지 없이 실행중인 가상 머신을 스냅샷 할 수 있는 방법을 오픈스택 공식 매뉴얼인 OPENSTACK OPERATIONS GUIDE에서 언급하고 있다. Live snapshots은 단순히 디스크 전용 스냅샷이며 인스턴스를 스냅샷 하는 것은 QEMU 1.3+ and libvirt 1.0+ 이상의 버전일 때 수행할 수 있다. 또한, Live snapshots은 오픈스택에서 제공하는 기본적인 기능이 아니라 fsfreeze 명령어를 통해 획득하는 방법이다.

4.2절 '다'에서 오픈스택 스냅샷 이미지 생성 시 무결성 검증을 위해 포렌식 관점에서 의미 있는 데이터를 수집하여 XML문서로 저장 및 관리하는 절차를 제안하였다. 이때 저장되는 데이터의 값은 오픈스택 Image service API v2를 참고하였으며 오픈스택에서 제공하는 REST API를 통해 획득한다.

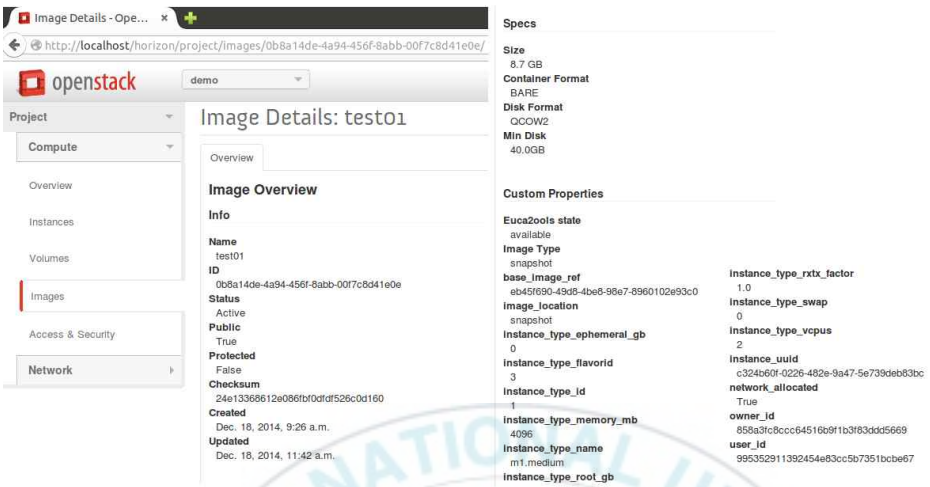


그림 18 대시보드에서 API를 이용한 스냅샷 이미지에 대한 정보

그림 18은 대시보드에서 스냅샷 이미지에 대한 정보를 웹에서 확인하는 것을 나타내고 있다. 그림에 보여지는 데이터는 Image API를 이용하여 나타내는 것이며, 대시보드 상에서 확인할 수 있지만 따로 저장 및 관리를 하는 기능은 없다. 따라서 본 논문에서는 그림 18에 나타난 데이터를 API를 이용하여 XML 문서로 저장하여 스냅샷 이미지에 대한 신뢰성을 향상시킨다.

오픈스택은 이미 모든 서비스가 REST API라는 개발 도구를 제공하고 있으며 Curl에서 REST API 사용법과 Client 툴에서 REST API 사용하는 방법이 있다. Curl은 콘솔(우분투에서는 터미널)에서 수행하며, Client 툴은 구글 웹 스토어에서 DHC-REST HTTP API Client 툴을 설치하여 설치한 툴을 이용한다[13].

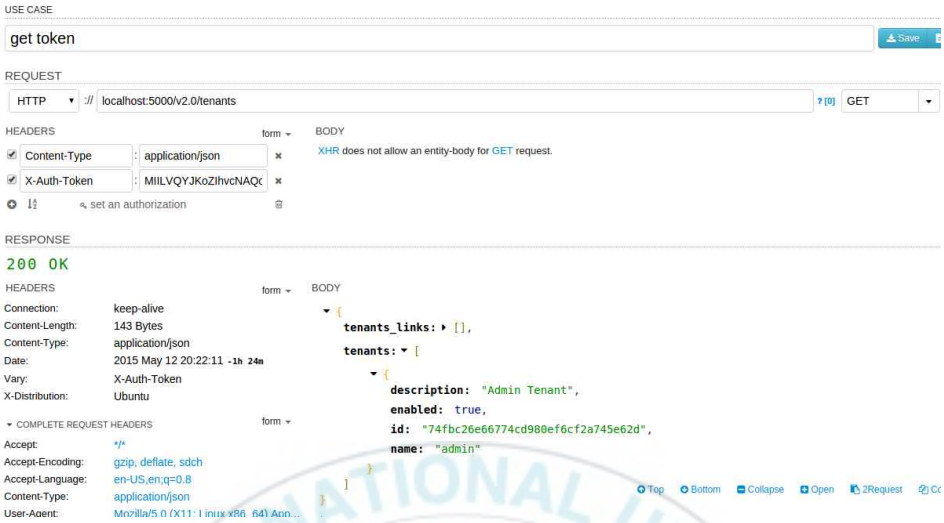


그림 19 서비스 URL과 GET 메소드를 이용한 테넌트 목록을 조회

그림 19는 DHC-REST HTTP API Client 툴을 이용한 REST API를 활용하여 테넌트 목록을 조회한 것이다. 이러한 방법으로 사용자의 인증 토큰을 획득하고 서비스 URL과 메소드 방식을 이용하여 제공하는 데이터를 획득 할 수 있다. 이러한 방법으로 API를 호출하여 특정 스냅샷 이미지에 대한 정보를 획득할 수 있다. 획득 가능한 정보는 현재 Image API v2를 기준으로 표 9와 같은 데이터를 획득할 수 있다. 그리고 실제 API를 호풀할 때 필요한 값과 획득한 데이터의 값을 그림 20에서 확인 할 수 있다. 같은 방법으로 관리 및 저장한다. 따로 문서화하여 저장하는 이유는 사용자가 인스턴스를 삭제한 경우 스냅샷 이미지에 대한 신뢰성 문제가 발생하고, 또한 현재 오픈스택에서는 스냅샷 이미지에 대한 정보를 따로 문서화하여 저장하지 않고 단지 대시보드에서 화면에 출력으로만 제공하기 때문에 변조 및 삭제를 한다면 스냅샷 이미지에 대한 무결성 및 신뢰성을 보장할 수 없다. 따라서 스냅샷이 생성되는 순간 스냅샷 이미지에 대한 정보를 API를 통해 문서화하여 저장하고 보관한다. 표 9는 오픈스택 공식홈페이지

에서 제공하는 Image API v.2를 참고하였다[26].

표 9 스냅샷 이미지에서 획득 가능한 데이터[26].

Parameter	Description
status	이미지 상태
name	이미지 이름
tag	이미지 태그
container_format	아미지의 컨테이너 포맷
created_at	이미지가 생성된 시간
disk_format	이미지의 디스크 포맷
updated_at	이미지가 업데이트 된 시간
min_disk	이미지가 부팅할 때 필요한 기가바이트의 최소 디스크 크기
protected	이미지가 삭제될 수 있는지 여부
id	이미지의 유일한 ID
min_ram	이미지가 부팅할 때 필요한 최소한의 메가바이트의 RAM
checksum	사용된 이미지의 해시 값
owner	이미지의 소유자 또는 테넌트의 id
visibility	이미지 가시성
size	이미지 데이터의 사이즈
url	이미지 파일에 대한 접근 할 수 있는 URL을 외부 저장소에 보관
metadata	워피 메타데이터
properties	속성
direct_url	이미지 파일에 대한 접근 할 수 있는 URL을 외부 저장소에 보관
self	가상 머신 이미지의 URL
file	가상 머신 이미지 파일의 URL
schema	가상 머신 이미지 스키마에 대한 URL

그림 20은 오픈스택 공식 홈페이지에서 제공하는 Image API v.2에서 발췌한 그림이다. 스냅샷 이미지에 대한 데이터를 획득할 때 필요한 값은 Image_id 이고 이 값은 일반적으로 UUID를 사용한다[26]. 또한, 획득 할 때 데이터의 형태는 그림에서 보여지는 형태로 나타난다. 그림 20은 획득된 데이터를 XML 문서로 저장한 것을 나타낸 것이다.

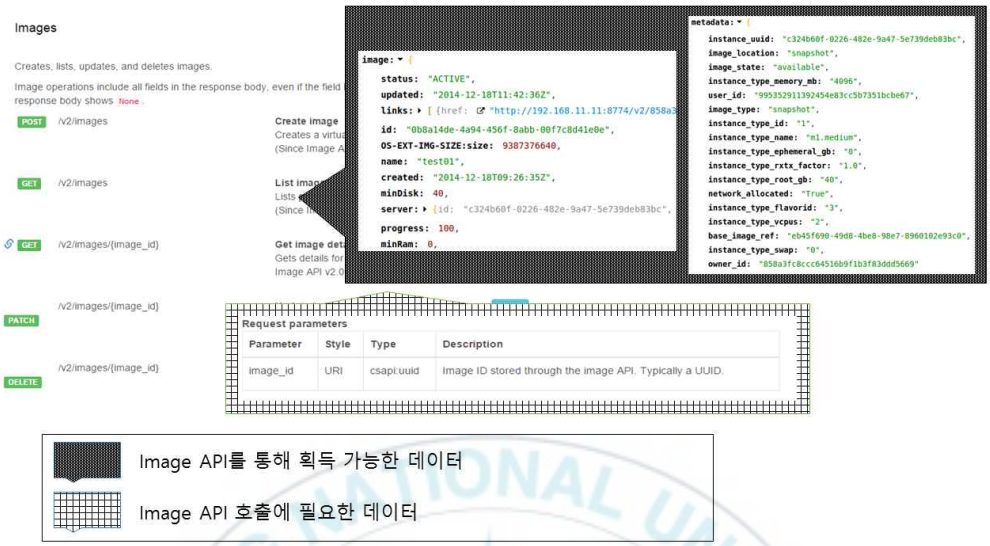


그림 20 API 호출시 획득 가능한 데이터

raw파일

XML파일

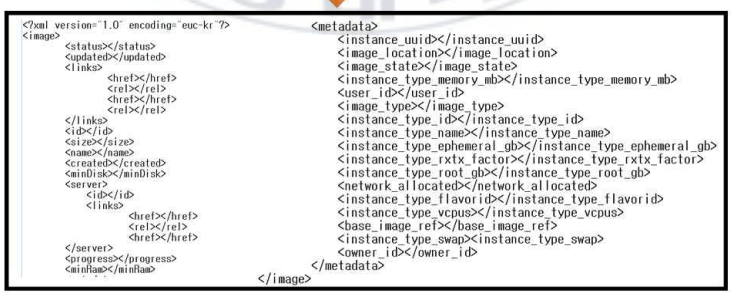


그림 21 raw 파일형식에서 XML문서로 변환

그림 22 XML문서로 저장된 스냅샷 이미지에 대한 데이터

그림 22는 그림 21에서 나타낸 XML문서 파일에 맡게 변환된 스냅샷 이미지에 대한 XML문서를 캡처한 그림이다. 각 태그는 표 9에서 정리한 데이터 값이며, 실제 해당되는 데이터들이 포함되어 있다. 각 각의 데이터 값들은 실제 포렌식 수사에 필요한 데이터이며, 스냅샷 이미지 파일과 동일한 위치에 저장된다.

나. 포렌식 도구를 이용한 증거 이미지 분석

포렌식 도구를 이용하여 획득한 증거 이미지 분석에 앞서 포렌식 수사를 위해 윈도우 7 가상 머신에서 임의의 행위들을 실시하였다. 또한, 앞서 설명한 snapshot 기능을 이용하여 포렌식 절차에 따라 이미지를 획득하고 포렌식 과정을 수행한다. 본 논문에서는 획득한 스냅샷 이미지 분석을 위해 Encase V7.08을 사용하였으며, 획득한 이미지를 변환하기 위해 qemu-img convert와 FTK Imager 3.4.0을 이용하였다.

윈도우 포렌식 분석은 파일 분석, 레지스트리 분석, 멀웨어탐지, 타임라인 분석, 애플리케이션 분석 등 실제 포렌식 분석 과정은 사건의 성격이나 특징에 따라 분석 방법을 적용한다[27]. 또한, 실제 획득한 가상 머신 이미지는 많은 데이터와 사용자 행위를 포함하고 있기 때문에 모든 분석을 적용하여 논문에서 보여주는 것은 현실적으로 불가능 하다. 따라서, 본 논문에서는 윈도우 7 가상 머신의 웹 히스토리 정보를 분석하고, 컴퓨터 내에 다운받은 파일 정보와 설치된 프로그램 정보 등을 파악하여 사용자 행위를 분석한다.

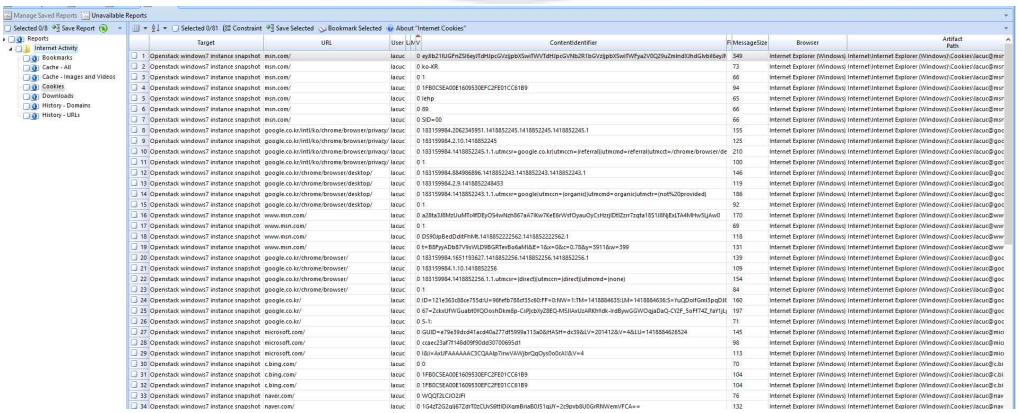


그림 23 가상 머신에서의 사용자 쿠키 정보

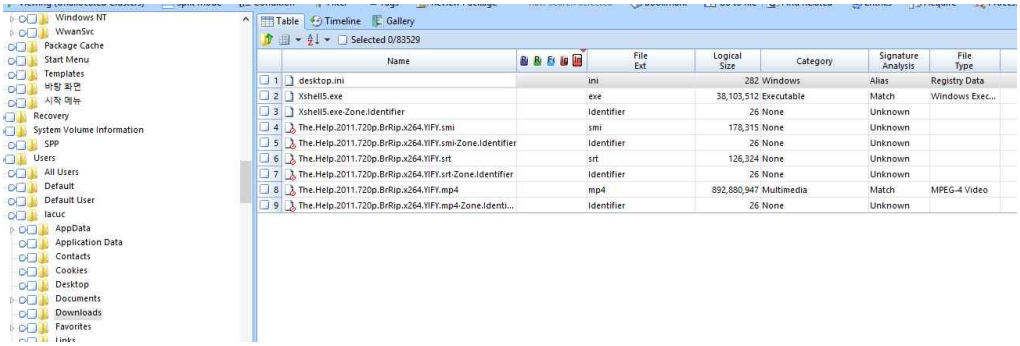


그림 24 다운로드 받은 동영상 파일 및 프로그램 파일

그림 23과 24는 Encase를 통해 획득한 스냅샷 이미지에서 사용자 웹 히스토리 분석을 위한 쿠키정보와 다운받은 파일 및 프로그램 등을 캡처한 화면이다. 그림 24에서는 가상 머신을 사용한 사용자가 다운받은 동영상과 Xshell5 프로그램을 다운받은 흔적을 확인할 수 있었다.

또한, 윈도우 7에서 마이크로소프트 디펜더(Microsoft Defender) 로그가 “ProgramData\Microsoft\Windows Defender\Support” 디렉터리에 남고, 이런 로그는 검사 기록과 삭제된 멀웨어에 관한 기록은 물론 이때 행해진 행위에 관한 기록까지 남게 된다[27]. 또한 사용한 스캐닝 엔진이나 시그니처 데이터베이스의 업데이트 정보까지도 기록된다[27]. 그림 25는 스냅샷 이미지에서 저장된 로그를 캡처한 화면이다.

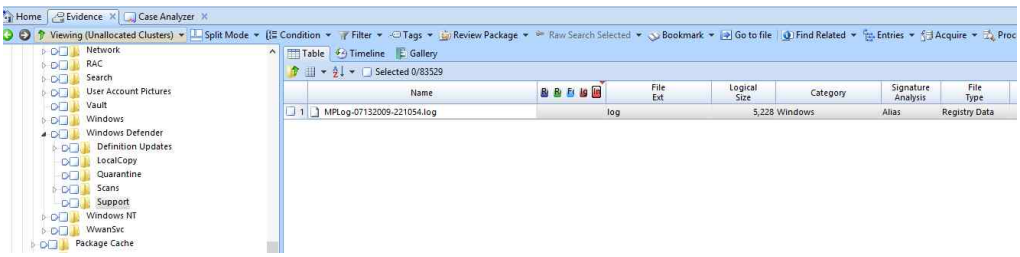


그림 25 마이크로소프트 디펜더 로그 파일

그림 23에서 25까지 캡처된 데이터는 오픈스택 계층에서는 확보할 수 없

는 사용자 데이터로써 실제 포렌식 수사에 중요한 증거로 사용된다. 이러한 데이터 획득을 위해 4.3절 '가'에서 스냅샷 이미지 수집에 앞서, 저장 및 보관하는 과정에서 무결성과 신뢰성을 위한 과정을 추가하여 가상 머신 이미지를 획득하고 분석하는 과정을 통해 증거 데이터로써 활용할 수 있도록 수행하였다. 획득된 가상 머신 이미지를 Encase Tool을 이용하여 사용자 행위 데이터를 추출할 수 있었다.

Ⅴ. 결론 및 향후 계획

클라우드 컴퓨팅 서비스 개발에 참여하는 국내외 기업들이 다양한 서비스를 제공함에 따라 클라우드 컴퓨팅 서비스를 사용하는 사용자 또한 증가하고 있다. 클라우드 컴퓨팅 서비스를 사용하는 경우, 기존 데스크톱 환경과는 달리 자원이 가상공간에 존재 할 수 있다. 따라서 클라우드 컴퓨팅 서비스를 사용하는 사용자에 대한 포렌식 조사가 이루어 질 경우, 기존의 디지털 포렌식을 수행하는 것은 한계가 있으며, 클라우드 환경에 대응하기 위한 새로운 포렌식 방안이 필요하다. 클라우드 컴퓨팅에 대한 디지털 포렌식은 현재 실질적인 역할을 수행하기에 아직 미비한 실정이며, 클라우드 포렌식에 대한 증거 수집 및 조사 과정에 대한 전반적인 체계를 확립할 필요가 있다.

본 논문에서는 다양한 클라우드 환경에 유연하게 적용할 수 있는 디지털 증거 수집 절차를 제안하기 위해 추상화된 클라우드 계층에 따라 수집 가능한 데이터를 분류하고, 확보한 증거 데이터의 신뢰성 보장을 위해 클라우드 플랫폼 기반의 데이터 수집 절차를 도식화하여 제안했다. 또한 오픈스택 기반 클라우드 컴퓨팅 환경에서 클라우드 포렌식의 문제점을 분석하고 이에 따른 해결방안으로 중앙로그서버를 구축하여 로그를 수집하고, 가상 머신 이미지 획득 절차에서의 신뢰성 향상을 위한 절차를 제안하였다.

마지막으로 클라우드 포렌식 문제점을 고려한 오픈스택 환경에서의 디지털 포렌식 분석을 위해 Snapshot을 이용한 가상 머신 이미지를 획득하여 제안한 절차를 적용하여 디지털 포렌식 도구를 이용한 포렌식 분석을 수행하였다.

본 연구를 통해 오픈스택 기반 클라우드 환경에서 클라우드 포렌식 문제

점을 해결할 수 있는 방안을 도출하였고, 신뢰성 향상을 위한 포렌식 조사 방안을 포렌식 절차에 추가하여 증거 데이터에 대한 신뢰성을 제공하였다. 또한, 제안한 절차에 따라 가상 머신 이미지를 획득하여 포렌식 도구인 Encase를 통해 사용하여 가상 머신 내에 있는 사용자 행위 데이터를 획득하였다. 이러한 과정은 가상 머신을 생성하여 불법 동영상을 다운받거나 악의적인 행위를 수행한 뒤 가상 머신을 삭제하여 증거데이터를 남기지 않는 공격자의 행위를 잡을 수 있으며, 가상 머신 이미지를 통해 공격자의 흔적을 찾아 낼 수 있다. 따라서, 클라우드 환경에서 가상 머신의 쉬운 생성 및 삭제를 이용한 악의적인 사용자의 행위를 획득함으로써 이에 따른 공격이나 악의적인 사용을 줄일 수 있을 것으로 기대된다.

본 논문에서 수행했던 클라우드 포렌식 과정은 오픈스택 기반 클라우드 환경에서 수행하였기 때문에 무료로 제공하는 오픈소스가 활용되었다. 무료로 사용할 수 있으며 많은 CSP가 오픈스택 기반 클라우드 서비스를 제공하기 때문에 가지는 장점이 있지만 본 논문에서 언급한 오픈스택 운용상의 로그 데이터와 오픈스택 API를 이용한 데이터는 오픈스택에 의존적일 수밖에 없다. 증거 데이터에 대한 신뢰성 문제를 해결하였지만, 일반적인 클라우드 환경에 적용하기 위해서는 세부적인 클라우드 플랫폼에 따른 데이터 형식 및 구성이 다르기 때문에 그대로 적용하기에는 한계가 있다.

따라서, 일반적인 클라우드 포렌식에서도 적용 가능한 독립적인 포렌식 서비스를 목표로 하는 연구가 필요하다고 생각된다.

참고문헌

[1] Zawoad .S and R. Hasan, "Cloud forensics: a meta-study of challenges, approaches, and open problems." arXiv preprint arXiv:1302.6312, 2013.

[2] 이상진, "디지털 포렌식 개론", 이룬, pp.105-137, 2010.

[3] C. H. Lee, "클라우드 환경을 고려한 디지털 포렌식 프레임워크", 한국항행학회 논문지, Vol 17, No.1, pp.63-38, 2013.

[4] 한수빈, 이태림, 신상욱, "클라우드 컴퓨팅 디지털 증거 수집 절차", 정보처리학회 학술발표대회 논문집, Vol 21, No.1, 2014

[5] 공용준, 오영일, 심탁길, "실전 클라우드 인프라구축 기술", 한빛미디어, pp.22-33 , 2014.

[6] J. Dykstra , A .T. Sherman, "Acquiring Forensic Evidence from Infrastructure-as-a-Service Cloud Computing: Exploring and Evaluating Tools, Trust, and Techniques", Digital Investigation, Vol 9, pp.90 - 98, 2012.

[7] J. Dykstra , A .T. Sherman, "Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform", Digital Investigation, Vol 9, pp.87 - 95, 2013.

[8] S. Zawoad, R. Hasan, "Digital Forensics in the Cloud", The Journal of Defense Software Engineering, Vol 26, No.5, pp.17-20, 2013.

[9] I. M. Abbadi, J. Lyle, "Challenges for Provenance in Cloud Computing", 3rd USENIX Workshop on the Theory and Practice of Provenance, USENIX Association, 2011.

[10] M. M. Potey, D. D. Nikumbh, "Achieving Accountability and

Secure Logging to Increase Trust in Cloud Environment", International Journal of Computer Applications, Vol 73, No.17, 2013.

[11] NIST Cloud Computing Forensic Science Working Group, "NIST Cloud Computing Forensic Science Challenges", Draft NISTIR 8006, 2014

[12] S. Simos, C. Kalloniatis and E. Kavakli, "Cloud Forensics Solutions: A Review." Advanced Information Systems Engineering Workshops. Springer International Publishing, 2014.

[13] 장현정, "오픈스택을 다루는 기술", 길벗, pp.70-72, 2014

[14] 김병식, 이범철, "오픈스택을 이용한 클라우드 서비스 플랫폼 구축 및 활용", 한국통신학회 학술대회논문집, 669-670, 2014

[15] www.openstack.org, OpenStack Open Source Cloud Computing Software

[16] T. Rubsamen1, C. Reich, "Evidence for Accountable Cloud Computing Services", Pre-Proceedings of International Workshop on Trustworthiness,

[17] 정일훈, 오정훈, 박정흠, 이상진, "IaaS 유형의 클라우드 컴퓨팅 서비스에 대한 디지털 포렌식 연구." 정보보호학회논문지, Vol 21, No.6, 2011.

[18] 한수빈, 이병도, 심종보, 신상욱, "클라우드 포렌식을 위한 오픈스택 플랫폼에서 로그데이터 수집", 정보처리학회 학술발표대회 논문집, Vol 21, No.2, 2014

[19] K.Jackson, C.Bunch, "OpenStack Cloud Computing Cookbook Second Edition", Packt Publishing Ltd, 304-306, 2013

[20] 한수빈, 이병도, 신상욱, “클라우드 포렌식을 위한 오픈스택 플랫폼에서 증거 신뢰성 향상”, 한국정보보호학회 영남지부 학술발표회 논문집, 94-69, 2015

[21] 정임영, 조인순, 유영진, "클라우드 컴퓨팅 환경의 데이터 신뢰 확보." *한국통신학회논문지* Vol. 36, No.9, 2011

[22] 한근희, 이경환, 이정근, “클라우드 컴퓨팅을 위한 오픈스택 운영가이드”, 인포더북스, pp.224-238

[23] Kent, Karen, "Guide to Computer Security Log Management", NIST Special Publication 800-92, 2007

[24] http://system-monitoring.readthedocs.org/en/latest/log.html

[25] 임성수, “VMware Workstation 가상 머신 이미지에 대한 디지털 포렌식 조사 절차 및 손상된 이미지 복구 방안”, 정보보호학회논문지, Vol. 21, No.2, 2011

[26] www.openstack.org, Image service API v2 Guide

[27] 할랜 카비, 고원봉, “윈도우 포렌식 분석 툴킷 : Windows Forensic Analysis Toolkit”, SYNGRESS, pp.191-205, 2013

[27] http://www.gartner.com/technology/topics/cloud-computing.jsp

[28] http://www.cisco.com/web/KR/about/news/2014/9-12/news_1117.html

[29] Cloud Computing Use Case Discussion Group. “Cloud Computing UseCases Version 3.0”, 2010.

Acknowledgement

감사의 글

길고도 짧았던 5년 6개월 동안 힘들었지만 즐거웠고 설레었던 시간들이었습니다. IT융합응용공학과를 시작으로 정보보호학협동과정으로 5년 6개월 동안 저를 응원해주시고 격려해주신 많은 분들에게 감사한 마음을 전하고자 합니다.

먼저 이제 막 1학년을 마친 저에게 연구실에 들어 올 수 있었던 기회를 주시고, 5년 동안 많은 격려와 조언으로 저에게 더 큰 세상과 길을 알려주신 신상욱 교수님께 감사드립니다. 그리고 늘 온화한 미소로 인사 받아주시고 아낌없는 조언과 격려를 주신 이경현 교수님께도 감사인사 전해드리고 싶습니다. 저에게 주신 소중한 가르침과 말씀 잊지 않고 늘 기억하겠습니다. 또한, 논문 지도를 위해 바쁘신 와중에도 세심한 지도와 관심을 주신 동명대학교 신원교수님께 감사드린다는 말씀 전하고 싶습니다. 그 외에도 훌륭한 가르침을 주셨던 부경대학교 IT융합응용공학과 교수님들께도 진심으로 감사드립니다.

LACUC 연구실은 저의 대학생활의 대부분을 보냈던 정말 소중한 공간이고, 좋은 추억입니다. 연구실에서 때로는 엄마였다가 때로는 아빠 같았던 태림선배 정말 감사해요. 그리고 다시 연구실에서 함께 수업 들었던 주영선배! 덕분에 더 많은걸 배우고 항상 많은 이야기 들려줘서 좋았어요. 이제는 너무 편해져버려서 자꾸 장난치지만 병도오빠 덕분에 석사 생활 더 재밌게 잘 할 수 있었어요. 항상 고마워하고 있습니다! 그리고 앞으로 연구실 구성원으로 남아있을 기웅오빠, 완석, 효민, 소정, 병조, 은영, 동진이도 앞으로 연구실에서 좋은 추억 쌓고 다들 응원할게! 또한, N.A.N 동아리 상철오빠에게도 정말 고맙다는 말 전하고 유리, 소현이도 동아리 잘 이끌어줘서 고마워! LACUC 연구실 그리고 N.A.N 동아리 선·후배님 모두 감사드립니다.

신입생 때부터 학교에서 유일한 친구들인 김예미, 남보현, 장은아. 지금까지 함께

보낸 시간동안 정말 즐거웠고 함께 격려하고 응원하는 만큼 다들 각자 원하는 대로 이루어지고, 시간이 지나도 지금처럼 편하게 수다 떨 수 있는 사이가 되었으면 좋겠어! 그리고 먼저 졸업한 영미도 서로 힘들 때 의지 할 수 있어서 좋았고 고마웠어. 조만간 꼭 보자! 지금은 거의 얼굴보기도 힘들지만 말도 많고 탈도 많았던 남자동기들 용수, 완석, 영수, 성진, 태호, 항석 다들 원하는 것 이루고 항상 응원할게! 그리고 멀리 있어도 항상 힘이 되어주는 친구들 지영, 은진, 정현, 지혜 모두 모두 고마워! 또 항상 힘들 때마다 힘이 되어주고 늘 제일 먼저 응원해주고 격려해준 이용수 정말 고맙고, 앞으로도 잘 지내자!

마지막으로 제가 이렇게 대학부터 대학원까지 무사히 마무리 할 수 있도록 아낌없는 지원을 해주시고 저를 믿어주신 가장 존경하고 사랑하는 엄마, 아빠에게 정말 감사드립니다. 부모님에게 자랑스러운 딸이 될 수 있도록 앞으로도 더 최선을 다할게요. 그리고 지금 중국에서 열심히 공부하고 있는 내 동생 다빈이도 고맙고 앞으로 열심히 해서 원하는 목표 이루고, 항상 응원할게!

많은 분들의 도움이 있었기 때문에 무사히 졸업할 수 있었습니다. 지금 가지고 있는 감사함 잊지 않고, 사회에 나가서도 도움이 필요한 곳에서 제가 가진 것들을 가치 있게 쓸 수 있는 사람이 되도록 하겠습니다. 그리고 기대에 보답하고 멋진 사람이 될 수 있도록 항상 최선을 다하겠습니다. 다시 한 번, 정말 감사드립니다.

2015.08

한 수 빈 드림