Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.2, Apr. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.2.295

Amazon S3 제로 트러스트 모델 설계 및 포렌식 분석*

조 경 현,[1†] 조 재 한,[2] 이 현 우,[3] 김 지 연[4‡]
[1,3,4]대구대학교 (대학원생, 학생, 교수), [2]부산대학교 (대학원생)

Design and Forensic Analysis of a Zero Trust Model for Amazon S3*

Kyeong-Hyun Cho,[1†] Jae-Han Cho,[2] Hyeon-Woo Lee,[3] Jiyeon Kim[4‡]
[1,3,4]Daegu University (Graduate student, Undergraduate student, Professor),
[2]Pusan National University (Graduate student)

요 약

클라우드 컴퓨팅 시장이 성장하면서 다양한 클라우드 서비스가 안정적으로 제공되고 있으며 국내 행정·공공 기관은 모든 정보시스템을 클라우드 시스템으로 운영하기 위한 전환사업을 수행하고 있다. 그러나 인터넷을 통해 클라우드 자원에 접근할 경우, 내·외부 인력의 잘못된 자원 사용 및 악의적인 접근이 가능하기 때문에 사전에 클라우드 서비스를 안전하게 운영하기 위한 보안 기술을 마련하는 것이 필요하다. 본 논문은 클라우드 서비스 중, 민감한 데이터를 저장하는 클라우드 스토리지 서비스에 대해 제로 트러스트 기반으로 보안 기술을 설계하고, 설계된 보안 기술을 실제 클라우드 스토리지에 적용하여 보안 기술의 실효성을 검증한다. 특히, 보안 기술 적용 여부에 따른 클라우드 사용자의 상세 접근 및 사용 행위를 추적하기 위하여 메모리 포렌식, 웹 포렌식, 네트워크 포렌식을 수행한다. 본 논문에서는 클라우드 스토리지 서비스로서 Amazon S3(Simple Storage Service)를 사용하고, S3의 제로 트러스트 기술로는 접근제어목록 및 키 관리 기술을 사용한다. 또한, S3에 대한 다양한 접근 유형을 고려하기 위하여 AWS(Amazon Web Services) 클라우드 내·외부에서 서비스 요청을 발생시키고, 서비스 요청 위치에 따른 보안 기술 적용 효과를 분석한다.

ABSTRACT

As the cloud computing market grows, a variety of cloud services are now reliably delivered. Administrative agencies and public institutions of South Korea are transferring all their information systems to cloud systems. It is essential to develop security solutions in advance in order to safely operate cloud services, as protecting cloud services from misuse and malicious access by insiders and outsiders over the Internet is challenging. In this paper, we propose a zero trust model for cloud storage services that store sensitive data. We then verify the effectiveness of the proposed model by operating a cloud storage service. Memory, web, and network forensics are also performed to track access and usage of cloud users depending on the adoption of the zero trust model. As a cloud storage service, we use Amazon S3(Simple Storage Service) and deploy zero trust techniques such as access control lists and key management systems. In order to consider the different types of access to S3, furthermore, we generate service requests inside and outside AWS(Amazon Web Services) and then analyze the results of the zero trust techniques depending on the location of the service request.

Keywords: Cloud Computing, Zero Trust, Cloud Forensics, Amazon Web Services, Amazon S3(Simple Storage Service)

Received(01. 19. 2023), Modified(03. 17. 2023), Accepted(03. 17. 2023)
* 본 논문은 2022년 한국정보보호학회 영남지부 학술대회에 발표한 우수논문을 개선 및 확장한 것임.
† 주저자, jgh1118@daegu.ac.kr
‡ 교신저자, jyk@daegu.ac.kr(Corresponding author)

I. 서 론

클라우드 컴퓨팅(Cloud Computing)은 인터넷 접속을 통해 컴퓨팅 자원을 이용하는 현대 컴퓨팅 패러다임으로서 수요에 따라 자원을 탄력적으로 운영할 수 있다는 장점이 있어 개인 및 기업뿐 아니라, 공공기관에서의 수요가 증가하고 있다[1]. 미국의 경우에는 2017년부터 연방정부의 정보시스템을 클라우드로 전환하기 위한 사업을 수행하고 있고, 국내의 경우에는 2021년부터 행정·공공 기관의 모든 정보시스템을 2025년까지 클라우드로 전환하기 위한 정책을 마련하여 현재 전환사업을 추진하고 있다. 그러나 클라우드 전환사업에 비해 클라우드 보안 모델 개발에 대한 노력은 부족한 실정이기 때문에 국가의 민감 데이터가 운영되는 클라우드 자원에 대한 내·외부 인력의 접근 및 사용을 감시하고, 악의적인 접근을 차단하기 위한 보안 솔루션을 마련해야 한다. 실제로 계정 탈취, 데이터 유출, 서비스 거부 등의 보안 사고가 Amazon, Google, Microsoft 등의 주요 클라우드 서비스 제공자에 계속하여 발생하고 있기 때문에 사전에 다양한 클라우드 자원을 위한 보안 모델을 설계하여 내부 보안 정책 및 보안 기술로서 운영하는 것이 필요하다[2]. 본 논문에서는 다양한 클라우드 자원 중, 민감한 데이터를 저장하는 클라우드 스토리지 서비스에 대한 보안 모델을 제로 트러스트(Zero Trust) 기반으로 설계하고, 이를 실제 클라우드 스토리지에 적용하여 보안 기술의 실효성을 검증하고자 한다.

본 논문에서는 상용 클라우드 스토리지 서비스 중, AWS(Amazon Web Services)의 스토리지 서비스인 Amazon S3(Simple Storage Service, 이하 S3)를 대상으로 보안 모델을 설계하고, 제로 트러스트 기술로는 접근 제어 목록(Access Control List, 이하 ACL) 및 키 관리 서비스(Key Management Service, 이하 KMS)를 적용한다. 또한, 클라이언트의 자원 요청 위치에 따른 제로 트러스트 기술의 적용 효과를 분석하기 위하여 AWS 클라우드 내부 및 외부에서 각각 자원 요청을 발생시키고, 메모리, 웹, 네트워크 포렌식을 통해 보안 기술 유형별 적용 효과 및 사용자의 자원 사용 흔적을 추적하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 제로 트러스트와 클라우드 포렌식을 수행한 관련 연구를 살펴보고, 3장에서는 S3를 위한 제로 트러스트 환경 구축 및 실험 시나리오를 개발한다. 4장에서는 개발된 시나리오별로 실험 결과를 분석하고, 5장에서 결론 및 향후 연구 계획을 제시한다.

II. 관련 연구

2.1 제로 트러스트 기반 클라우드 보안 연구

제로 트러스트는 컴퓨팅 자원에 접근하는 내·외부 사용자에 대해 '절대 신뢰하지 않고, 항상 검증한다'를 전제로 한 개념적 모델로서 컴퓨팅 자원에 접근하기 위해서는 권한과 권한의 유효성을 반드시 입증해야 한다. 제로 트러스트 보안 모델의 종류로는 제로 트러스트 네트워크 접속 및 보안 웹 게이트웨이 등이 있으며 원격근무를 통해 기업의 컴퓨터를 사용하거나 사설 네트워크를 사용하는 이용자들을 위해 다양하고 광범위하게 확장되고 있다.

제로 트러스트 기술에는 사용자가 서버를 통하여 관리가 가능한 키(key) 관리 서버를 예로 들 수 있으며 기존의 프로세스에 기반한 암호화 방식과는 달리 사용자가 암호화 키를 공동 키를 통하여 등록할 수 있으므로 사용자의 암호화 키에 대한 기밀성을 보장받을 수 있다는 장점이 있다[3]. 제로 트러스트 모델에 대한 제안으로는 ID 관리 모델인 ZTIMM(Zero-Trust-Based Identity Management Model)[4], 멀티 클라우드에 대한 제로 트러스트[5], 신뢰 관계 기반의 클라우드 컴퓨팅 제로 트러스트 전략 모델[6] 등과 같이 다양한 모델이 제안되었다. 또한, 클라우드 환경에서 제로 트러스트 네트워크 구조를 구현하는 방법[7], 클라우드 환경에 저장되어 있는 자원의 암호화 및 복호화를 위한 보안 플랫폼[8], IAM(Identity and Access Management)을 위한 제로 트러스트 프레임 워크[9]를 제작하는 연구도 진행되었다.

본 논문에서는 여러 제로 트러스트 기술 중, ACL 및 KMS를 활용하여 클라우드 스토리지 서비스의 보안 연구를 수행한다.

2.2 클라우드 포렌식 연구

상용 클라우드 스토리지 서비스가 증가하면서 Dropbox, Evernote, KT Cloud, MYBOX와 같은 상용 클라우드 스토리지 서비스에 관한 포렌식 연구가 주로 진행되었다[10]. 또한, 2011년에는 클라우

드 스토리지 서비스 사용 시, 윈도우 시스템 및 스마트 폰에 남는 흔적을 분석한 포렌식 연구가 진행되었다[11]. 클라우드 포렌식을 위한 도구인 UFED Cloud Analyzer, FROST와 클라우드 포렌식 단계를 제시한 연구[12]와 기존에 존재하는 데이터 수집 도구를 활용하여 클라우드 환경에서 데이터를 원격 수집하는 포렌식 연구[13]도 진행되었다. 그러나 기존 연구 중, 제로 트러스트 기반의 클라우드 서비스를 구축하고 포렌식을 수행한 연구는 존재하지 않는다. 본 연구는 다양한 클라우드 스토리지 서비스 중, S3를 대상으로 ACL 및 KMS와 같은 제로 트러스트 기술을 적용하고, 제로 트러스트 기술의 효과를 분석하기 위한 포렌식을 수행한다는 점에서 기존 연구와의 차별점이 있다.

III. S3 제로 트러스트 모델 설계

본 장에서는 ACL 및 KMS를 활용하여 S3 제로 트러스트 모델을 설계하고, S3에 대한 클라우드 내·외부 접근을 고려하기 위하여 AWS 클라우드를 설계하는 방안을 설명한다. 표준 AWS 운영 환경을 구축하기 위하여 AWS의 모범 사례를 제공하는 AWS Prescriptive Guidance[14, 15]를 참고하여 EC2, KMS 및 S3의 보안 환경을 구축하였다.

3.1 제로 트러스트 기반의 Amazon S3 설계

3.1.1 ACL(Access Control List)

ACL은 파일 저장소인 S3의 버킷(bucket)과 저장된 각 파일을 의미하는 객체에 모두 적용할 수 있으며 S3 접근 가능 여부를 결정하는 정책으로서 동작한다. S3 버킷에 ACL 기능을 활성화하면 S3 버킷의 URL(Uniform Resource Locator)을 통하여 누구나 S3 버킷에 접근하는 것이 가능하며 객체에 ACL을 적용하면 각 객체에도 접근 가능하다.

S3 버킷의 ACL 기능을 활성화하지 않는다면 객체의 ACL은 디폴트(default)로 활성화되지 않으므로 S3 버킷과 객체의 권한을 소유한 사용자만 S3 관리 콘솔(management console)을 통해서 접근할 수 있게 된다. 본 논문에서는 세 개의 S3 버킷을 생성하고, ACL 활성화 정책을 다르게 설정하여 ACL 여부에 따른 버킷 및 객체의 보안성을 비교 및 분석한다.

AWS 리전(region) 중, 오레곤(us-west-2) 지역에는 두 개의 S3를 생성하여 각각 ACL 활성화, 비활성화하여 운영하고, 버지니아 북부(us-east-1)에는 하나의 버킷을 생성하여 KMS 적용 효과를 검증하기 위해 ACL을 비활성화하였다.

3.1.2 KMS(Key Management Service)

AWS KMS는 AWS 서비스에 맞게 조정된 암호화 및 키 관리 서비스로서 광범위한 AWS 서비스에서 손쉽게 키를 생성하고 관리하며 암호화 사용을 제어할 수 있다. KMS는 사용자가 생성, 소유 및 관리할 수 있는 고객 관리형 키와 AWS 서비스에서 자원 보호를 위하여 고객 대신 생성, 관리 및 사용하는 AWS 관리형 키로 구분할 수 있다. 본 연구에서는 이 중, 데이터의 기밀성과 신뢰성을 보장하고 인증된 암호화를 지원하는 대칭 KMS인 고객 관리형 키를 생성하여 사용하였다.

본 논문에서는 S3 버킷에 객체 업로드 시, 객체 암호화를 먼저 수행하기 위하여 KMS를 사용하였으며 사용자별 AWS 자원 접근을 허용하는 IAM 기능을 통해 각 사용자에게 KMS 권한을 부여한다. IAM은 KMS 읽기, 태그 지정, 쓰기, 권한관리 등을 부여하여 KMS를 생성, 제어 및 사용할 수 있게 하며 본 연구에서는 KMS 권한을 가진 사용자에게 최소 권한을 부여하기 위해 암호화된 객체를 열람할 수 있는 읽기 권한만 부여하였다.

3.2 클라우드 내·외부 접근 분석을 위한 AWS 클라우드 구축

본 논문에서는 클라우드 내부 및 외부에서 발생하는 S3 접근 요청을 모두 고려하기 위하여 AWS 클라우드 내부에도 웹 서버를 구축하고, 웹 서버를 통한 S3 접근 요청을 발생한다. AWS 클라우드 내부에 웹 서버를 구축하기 위하여 사설 네트워크를 구축하는 기능인 AWS VPC(Virtual Private Cloud)를 생성하여 서브넷, 라우팅 테이블, 인터넷 게이트웨이를 배치하였고, Amazon EC2(Amazon Elastic Compute Cloud)를 사용하여 RedHat Linux 기반의 웹 서버를 생성하였다. 본 논문에서 설계한 AWS 클라우드 내부 아키텍처는 Fig. 1.과 같다.

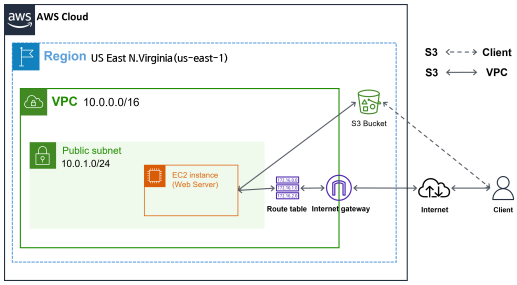

Fig. 1. AWS Cloud Architecture for S3 Access

Fig. 1.에서 클라이언트부터 S3 버킷까지 이어지는 실선을 통해 클라이언트는 VPC 내부의 웹 서버를 통해 S3 버킷에 자원을 요청할 수 있으며 웹 서버를 통하지 않고 클라우드 외부에서 S3 버킷에 접근한다면 Fig. 1의 점선 경로와 같이 직접 자원을 요청할 수 있다.

3.3 실험 시나리오 설계

본 연구에서는 S3 버킷 및 객체의 ACL 활성화 여부, 버킷 객체에 적용된 KMS 여부, 클라우드 내·외부 접근 위치, 포렌식 위치를 고려하여 Table 1.과 같이 총 7개의 실험 시나리오를 설계하였다.

7개의 시나리오는 버킷과 객체의 ACL 활성화 여부와 KMS 적용 여부에 따라 구분되며 접근 유형의 경우 클라우드 외부에서 직접 S3에 접근하는 Client와 VPC의 웹 서버를 통하여 접근하는 Cloud로 분류하였다. ACL-1부터 ACL-4는 ACL 활성화 여부에 따른 차이를 비교하는 목적이기 때문에 모두 KMS는 적용하지 않았다. ACL-1에서 ACL-3은 클라우드 외부에 있는 클라이언트가 직접 접근하므로 포렌식은 클라이언트의 로컬 시스템에서 진행하며 ACL-4는 구축한 클라우드 내부의 웹 서버를 통해 S3에 접근하므로 웹 서버에서 포렌식을 실시한다.

KMS-1부터 KMS-3은 KMS를 적용하였을 때 객체에 대한 접근 가능 여부를 확인하기 위한 시나리오로서 KMS-1은 ACL 활성화, KMS-2와 KMS-3의 ACL은 모두 비활성화하고, KMS을 적용하였다. KMS-1은 클라우드 외부에서 S3 객체에 접근하는 시나리오로서 클라이언트의 로컬 시스템에서 포렌식을 수행하고, KMS-2와 KMS-3은 클라우드 내부에서 S3로 접근하는 시나리오로서 VPC 내부의 웹 서버에서 포렌식을 수행한다. 포렌식은 메모리 포렌식, 웹 포렌식, 네트워크 포렌식을 수행하며 메모리 포렌식을 위해서는 S3 접근 시, 메모리 덤프 파일을 분석한다. 웹 포렌식에서는 클라우드 외부에서 클라이언트가 S3에 직접 접근하는 경우에는 웹 디렉토리 변경, 쿠키 파일 및 히스토리(history) 정보를 수집하고, 클라우드 내부의 웹 서버에서 S3에 접근하는 경우에는 웹 서버의 웹 로그 파일을 분석한다. 마지막으로 네트워크 포렌식에서는 클라우드 내·외부 접근 위치에 상관없이 S3 자원 접근 시 발생되는 패킷(packet)을 실시간 수집하여 결과를 분석한다.

Table 1. Experimental Scenarios

Scen-ario	ACL		KMS	Access		Forensic location	
	Bucket	Object		Client	Cloud	Local	Web
ACL-1	✓	✓	-	✓	-	✓	-
ACL-2	✓	-		✓	-	✓	-
ACL-3	-	-		✓	-	✓	-
ACL-4	-	-		-	✓	-	✓
KMS-1	✓	✓	✓	✓	-	✓	-
KMS-2	-	-		-	✓	✓	-
KMS-3	-	-		-	✓	-	✓

IV. 시나리오별 포렌식 실험 결과 분석

본 장에서는 3장에서 설계한 7개의 시나리오로 진행한 실험 결과를 포렌식을 통해 보안 기술의 적용 결과 및 자원 접근 흔적을 분석한다.

4.1 메모리 포렌식

S3 접근 전·후의 메모리 덤프를 수행하여 S3 자원 접근 기록을 분석한 결과, ACL-1부터 ACL-3까지는 공통적으로 Fig. 2.와 같이 네 개 영역에서 객체 다운로드 흔적이 발견되는 것을 확인할 수 있다.

다운로드 시작부터 완료까지의 진행 과정을 이벤트 명(eventName) 속성의 〈v3-Obj_Download_Start〉, 〈v3-Obj_Download_URL-doc〉, 〈v3-GeneratePresigned URL-sent〉, 〈v3-Obj-Download_Success〉 영역을 통해 확인할 수 있으며 다운로드 받은 파일이름(key), 파일 크기(size), 시간(timestamp) 정보도 확인할 수 있었다. 시간의 경우, S3는 글로벌 표준 시간을 기준으로 제공되기 때문에 한국시간보다 9시간 느리게 기록되는 것

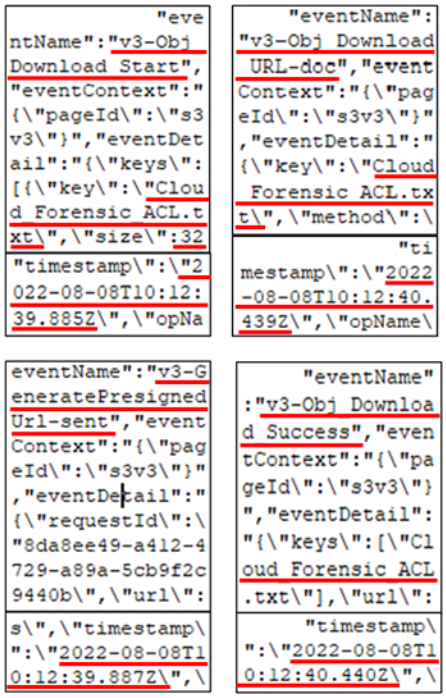

Fig. 2. Experimental Results of Memory Forensics for ACL-1, ACL-2, and ACL-3

을 확인하였다.

KMS를 적용한 KMS-1은 S3에 직접 접근하여 메모리 덤프를 실행하였으며 ACL-1부터 3까지의 포렌식 덤프 파일에서 발견되는 데이터의 정보가 유사하게 나타났다. ACL-4와 KMS-3의 경우에는 클라우드 내부의 웹 서버에서 메모리 덤프를 실행하였으며 클라이언트가 제작된 VPC를 통하여 S3에 접근하였을 때 로컬에서 수행한 메모리 포렌식 결과에 비해 기록된 흔적이 약 절반으로 줄어든 것을 확인하였다. 또한, ACL-4, KMS-2, KMS-3 메모리 포렌식 분석 결과, 웹 서버 정보, 다운로드 요청 시간, 다운로드 요청 파일명 등의 정보가 저장되며 Fig. 3.과 같은 〈ZoneTransfer〉도 정보도 확인 가능하였다.

〈ZoneTransfer〉의 〈ZoneId〉 1부터 4의 값으로 객체 접근에 대한 연결 방식을 표현하는데 1은 로컬 인트라넷, 2는 신뢰할 수 있는 사이트, 3은 인터넷, 마지막으로 4는 제한된 사이트에서 다운로드를 실행한다는 의미를 가진다. 본 연구 결과에서는 Fig. 3.과 같이 〈ZoneId〉 값이 3으로 기록된 것을 확인할 수 있다.

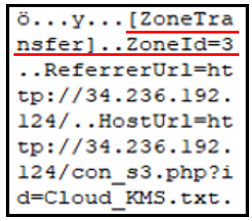

Fig. 3. ZoneTransfer Information for ACL-4, KMS-2, and KMS-3

4.2 웹 포렌식

클라우드 외부의 클라이언트가 S3에 직접 접근하는 경우에는 클라이언트의 로컬 시스템에 설치된 크롬(chrome) 웹 브라우저에서 쿠키 파일 변경 사항, 히스토리 기록을 포렌식하고, 클라이언트가 클라우드 내부의 웹 서버를 통해 S3에 접근하는 경우에는 클라우드 내부의 Apache 웹 서버에서 access_log 파일과 error_log 파일을 확인하는 방법으로 포렌식을 수행하였다. 클라이언트가 S3 객체에 접근하는 방법으로는 객체 URL을 통해 접근하는 방법, AWS의 관리 콘솔을 통해 접근하는 방법이 있다. ACL-1, ACL-2 ACL-3의 경우, URL을 통해 접근했을 때에는 Fig. 4.와 같이 S3 객체의 URL 정보를 히스토리에서 확인할 수 있었고, AWS 관리 콘솔을 통하여 접근하였을 때에는 Fig. 5.와 같이 S3 접속 과정에서 내부적으로 접속한 다수의 URL이 기록된 것을 확인할 수 있었다.

ACL-1과 ACL-3의 쿠키를 비교하면 7개의 쿠키가 추가되고 쿠키 유효기간 및 쿠키 명 또한 변경되는 것을 확인하였다. KMS-1에서 객체 다운로드

Fig. 4. URL history for ACL-1 when accessing S3 using URL

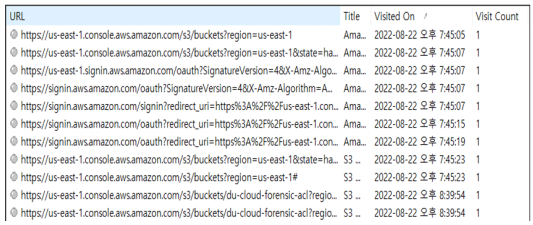

Fig. 5. URL history for ACL-1 when accessing S3 using the AWS management console

전·후의 쿠키를 비교하였을 때에는 ACL-1부터 ACL-3과는 다르게 초기 지역 설정인 〈noflust_Region〉이 us-west-2에서 us-east-1로 지역이 변동된 것을 확인할 수 있었다. 클라이언트가 클라우드 내의 웹 서버를 통하여 S3에 접근하는 ACL-4와 KMS-3의 경우, 웹 서버 내의 '/var/log/httpd' 에서 로그 파일을 확인하였으며 기간별로 저장되어있는 로그 파일도 확인할 수 있었다. 폴더 내에는 ssl_access_log, ssl_error_log, ssl_ request_log 파일도 존재하지만, 본 연구에서는 암호화 기반 인터넷 프로토콜을 허용하지 않았기 때문에 해당 파일들은 공백 파일로 존재하므로 access_ log와 error_log 파일에 대해서 분석하였다.

access_log 파일에는 접속 요청이 들어온 IP와 요청 시간, 요청 방식, 상태 코드, 전송 데이터 크기 등이 기록되어 있으며 기록된 시간은 메모리 포렌식 결과와 마찬가지로 한국 시간보다 9시간 느리게 기록되는 것을 확인할 수 있었다. KMS-2의 경우, 쿠키값이 전혀 생성되지 않았으며 history는 최초 접속을 시도한 S3 객체의 URL만 기록되었다. ACL-4, KMS-2와 KMS-3은 클라이언트가 모두 클라우드 내부에 구축된 웹 서버를 통하여 S3에 저장된 객체에 대해 다운로드를 시도하지만, ACL과 KMS 모두 접근 권한이 없는 외부 사용자로 판단되어 정상적으로 파일이 다운로드 되지 않는다. 예를 들어, 텍스트 파일을 다운로드 할 경우, 클라이언트의 로컬 시스템에 텍스트 파일이 다운로드 되지만, 다운로드 된 파일을 열어보면 파일이 비어있는 것을 확인할 수 있다. 이는 웹 서버가 KMS 권한을 부여받지 않아, KMS로 암호화된 객체에 대한 접근 권한이 없기 때문이다.

4.3 네트워크 포렌식

클라우드 외부의 클라이언트 시스템에서 네트워크 포렌식을 수행하기 위해서는 WireShark를 사용하여 패킷 캡처를 수행하였고, 클라우드 내부의 웹서버에서 네트워크 포렌식을 수행할 때는 TShark를 사용하여 패킷을 실시간 수집하였다. 각 시나리오 유형별 패킷의 통계를 분석한 결과는 Table 2.와 같다.

클라우드 외부에서 AWS 관리 콘솔을 통하여 S3 객체를 다운로드하며 네트워크 포렌식을 수행한 ACL-1, ACL-2, ACL-3, KMS-1 네 가지 시나리오에서는 파일에 접근하는 과정에서 패킷의 개수가

Table 2. Experimental Results of Network Forensics

Scena-rio	Cou-nt	Aver-age	Mini-mum	Maxi-mum
ACL-1	22	698.73	54	9297
ACL-2	16	797.94	54	9319
ACL-3	16	753.94	54	9312
ACL-4	88	175.61	54	4350
KMS-1	19	137.47	54	564
KMS-2	239	649.00	54	16467
KMS-3	78	170.00	54	5020

평균 18개가 발견되었으며 패킷 길이가 40 이상 79 이하인 패킷이 전체 패킷의 50%이상을 차지하는 것을 확인할 수 있었다. 클라우드 내부의 웹 서버를 통해 S3 객체에 접근한 KMS-2 의 경우에는 239개의 패킷을 확인할 수 있었으며, 패킷 길이가 40 이상 79 이하인 패킷이 전체 패킷의 60% 이상을 차지하는 것을 확인할 수 있었다. 클라우드 외부의 로컬 시스템에서의 S3 객체를 다운로드 받는 과정에서 네트워크 포렌식을 수행한 결과, 4가지의 시나리오 모두에서 SSL Handshaking이 진행되는 것을 확인하였다. 따라서 네트워크 패킷이 암호화되어 대부분의 정보가 보이지 않았지만, Fig. 6.과 같이 S3 버킷명은 확인할 수 있었다.

클라우드 내부의 웹 서버에서 다운로드가 완료된 이후, 파일 객체를 열람하는 과정에서 네트워크 포렌식을 수행한 결과, KMS를 적용하지 않은 ACL-4의 경우, 파일에 접근하는 과정에서 TCP 통신이 이루어지고, 이 과정에서 Fig. 7.과 같이 HTTP 페이로드에 본 연구에서 구축한 웹 서버의 소스 코드가 유출되는 것을 확인하였다.

TCP 통신이 종료된 이후에는 UDP 패킷을 송수신하였으며 UDP 패킷에서는 Queries와 Answers의 정보를 확인할 수 있었다. 또한, 이후에는 TLS 패킷을 주고받으며 SSL Handshaking 과정을 수행하고, 이 과정이 종료되면 HTTP 패킷을 전송하는 것을 확인할 수 있다. HTTP 패킷에는 Fig. 8과

Fig. 6. Payload data for a TLS Packet

```
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
        <title>DU-Cloud Forensic</title>
</head>

<body>
        <h1 style="text-align:center;">DU-Cloud Forensic</h1>

        <table border="1" style="margin-left: auto; margin-right: auto;">
        <h4 style="text-align:center;">..... ........</h4>
                <tr>
                        <td style="text-align:center;"><a href="/con_s3.php?
id=Cloud_Forensic.txt">KMS-2</a></td>
                </tr>
                <tr>
                        <td style="text-align:center;"><a href="/con_s3.php?
id=Cloud_KMS.txt">KMS-3&4</a></td>
                </tr>
                </table>
        <td>

</body>
</html>
```

Fig. 7. Payload of a HTTP packet

```
GET /con_s3.php?id=Cloud_Forensic.txt HTTP/1.1
Host: 34.236.192.124
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/108.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://34.236.192.124/
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7,ja;q=0.6

HTTP/1.1 200 OK
Date: Sun, 08 Jan 2023 10:06:04 GMT
Server: Apache/2.4.37 (Red Hat Enterprise Linux) OpenSSL/1.1.1k
X-Powered-By: PHP/7.2.24
Content-Disposition: attachment; filename="Cloud_Forensic.txt"
Cache-Control: max-age=0
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/octet-stream
```

Fig. 8. Analysis Result of HTTP Packet

같이 S3 버킷의 IP, 다운로드 받은 객체 파일명, 다운로드 받은 시간, S3 버킷 정보를 확인할 수 있었다.

KMS를 적용한 KMS-3의 경우에는 ACL-4와 다르게 파일에 접근하는 과정에서 TCP 통신이 이루어지지 않았으며, 웹 서버의 소스 코드가 HTTP 페이로드에 유출되는 문제는 발생하지 않았다. 이후 과정에서의 포렌식 결과는 ACL-4와 동일하게 관찰되어 KMS-3에서도 Fig. 8.과 같이 S3 버킷의 IP 및 정보, 다운로드 받은 객체 파일명 및 시간 등을 확인할 수 있었다.

V. 결 론

본 논문에서는 AWS에서 제공하는 스토리지 서비스인 S3를 위한 제로 트러스트 모델을 ACL 및 KMS 기술을 활용하여 설계하고, 포렌식을 통해 적용된 보안 기술의 결과 및 S3 자원 사용에 대한 흔적을 분석하였다. 실험을 위해 S3의 ACL 활성화 여부 및 데이터의 KMS 기반 암호화 여부를 고려하여 7개의 실험 시나리오를 개발하였고, 클라우드 내·외부에서의 S3 자원 접근 요청을 모두 발생시키기 위하여 AWS 클라우드 내부에 웹 서버를 구축하였다. 또한, 제로 트러스트 기술 적용 여부에 따른 S3 자원 접근 가능 여부 및 자원 사용 흔적 분석을 위해 메모리 포렌식, 웹 포렌식, 네트워크 포렌식을 수행하였다.

실험 결과, S3 버킷에 저장된 객체는 기본적으로 버킷 및 객체 모두 ACL 활성화가 된 경우에만 다운로드 가능하고, KMS 기반으로 암호화된 객체에는 해당 KMS 권한을 가진 사용자만 접근이 가능한 것을 확인하였다. 그러나 자원 이용 여부와 관계없이, S3 객체 정보 및 접근 기록은 메모리 포렌식을 통해 흔적을 추적할 수 있고, 클라우드 외부의 클라이언트 웹 브라우저에서는 S3 객체에 대한 URL 접속 흔적을 웹 포렌식을 통해 찾을 수 있었다. 또한, 클라우드 내부의 웹 서버에서 수행한 포렌식 결과, S3 자원에 대한 접근 요청 및 성공 여부가 웹 서버에 흔적으로 기록되는 것을 확인하였다. 마지막으로 네트워크 포렌식을 통해서는 클라우드 외부의 클라이언트가 직접 S3에 접근할 때보다 클라우드 내부의 웹 서버를 통해 접근할 때 더 많은 패킷 교환이 이루어지는 것을 확인하였고, ACL 비활성화 및 KMS 적용 여부와 상관없이 클라우드 외부에서 S3에 접근할 때에는 기본적으로 SSL 통신이 이루어져 기본적인 S3 버킷 명 외에는 다른 흔적이 발견되지 않았다. 이에 비해, AWS 클라우드 내부의 웹 서버를 통해 S3에 접근하는 경우에는 웹 서버와 S3 간의 HTTP 패킷을 통해 S3 버킷 주소, 객체 명 및 다운로드 시간 등 더 많은 사용 흔적을 발견할 수 있었다.

본 논문은 클라우드 스토리지 서비스의 보안성 강화를 위한 제로 트러스트 참조 모델로 활용될 수 있으며 향후에는 클라우드 스토리지 서비스 외의 다양한 컴퓨팅 및 네트워크 자원을 위한 제로 트러스트 모델을 설계하고, 직접 공격을 주입하며 설계된 제로 트러스트 모델을 검증할 예정이다.

References

[1] T.G. Lee and S.H. Lee, "Information interaction between cloud computing and smart computing," Korean Institute of Information Technology Magazine, 10(1), pp. 45-52, Mar. 2012.

[2] H.C. Gwon, D.Y. Jeong, B.H. Jeong and J.N. Kim, "Cloud security overview," The Journal of The Korean Institute of Communication Sciences, 32(10), pp. 71-76, Sep. 2015.
[3] Ki Hyun Jung and Seung Jung Shin. "Key management server design for providing cryptographic service in cloud computing environment (services in a cloud environment)," International journal of advanced smart convergence, 5(4), pp. 26-31, 2016.
[4] Abdullah Albuali, Tessema Mengistu and dunren Che, "ZTIMM: A zero-trust-based identity management model for volunteer cloud computing," International Conference on Cloud Computing, LNISA vol. 12403, pp. 287-294, Sep. 2020.
[5] S. Rodigari, D. O'shea, P. McCarthy, M. McCarry and S. McSweeney, "Performance analysis of aero-trust multi-cloud," 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), pp. 730-732, Sep. 2021.
[6] Saima Mehraj and M. Tariq Banday, "Establishing a zero trust strategy in cloud computing environment," 2020 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-6, Jan. 2020.
[7] Eric Ackley, "Zero trust networking in a cloud environment," M.S. Thesis, California State University San Marcos, Aug. 2022.
[8] F. Zhang and X. Jiang, "The zero trust security platform for data trusteeship," 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), pp. 1014-1017, Mar. 2021.
[9] Monjur Ahmed and Krassie Petrova. "A zero-trust federated identity and access management framework for cloud and cloud-based computing environments," WISP 2020 Proceedings, vol. 4, pp. 1-6, Dec. 2020.
[10] S.H. Seo, J.E. Kim and C.H. Lee, "Trend of cloud storage analysis research in terms of digital forensic," Review of KIISC, 32(2), pp. 29-36, Apr. 2022.
[11] H.J. Chung, J.H. Park and S.J. Lee, "Digital forensic investigation of devices using cloud storage service," Journal of Digital Forensics, (8), pp. 1-25, Nov. 2011.
[12] Josiah Dykstra and Alan 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, Aug. 2012.
[13] Sandesh Achar, "Cloud computing forensics," International Journal of Computer Engineering and Technology(IJCET), vol. 13, no. 3, pp. 1-10, Dec. 2022.
[14] AWS, EC2 Security, Ensure that an IAM profile is associated with an EC2 instance, Available online: https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/ensure-that-an-iam-profile-is-associated-with-an-ec2-instance.html?did=pg_card&trk=pg_card (Accessed on March 9 2023).
[15] AWS, S3 Security, Encryption best practices and features for AWS services, Available online: https://docs.aws.amazon.com/prescriptive-guidance/latest/encryption-best-practices/welcome.html (Accessed on March 9 2023).

〈저자소개〉

조 경 현 (Kyeong-Hyun Cho) 학생회원
2023년 3월~현재: 대구대학교 일반대학원 컴퓨터정보공학과 석사과정
2023년 2월: 대구대학교 정보통신대학 컴퓨터정보공학부(컴퓨터소프트웨어전공) 공학사
〈관심분야〉 클라우드 컴퓨팅, 사이버 보안, 암호학

조 재 한 (Jae-Han Cho) 학생회원
2023년 3월~현재: 부산대학교 일반대학원 정보융합공학과(컴퓨터공학전공) 석사과정
2023년 2월: 대구대학교 정보통신대학 컴퓨터정보공학부(컴퓨터공학전공) 공학사
〈관심분야〉 클라우드 컴퓨팅, 사이버 보안, 네트워크 보안, Zero Trust

이 현 우 (Hyeon-Woo Lee) 학생회원
2021년 2월~현재: 대구대학교 정보통신대학 컴퓨터정보공학부(컴퓨터공학전공) 학사과정
〈관심분야〉 클라우드 컴퓨팅, 인공지능, 사이버 보안

김 지 연 (Jiyeon Kim) 정회원
2021년 3월~현재: 대구대학교 컴퓨터정보공학부 컴퓨터공학전공 조교수
2019년 3월~2021년 2월: 서울여자대학교 소프트웨어교육혁신센터 전담교수
2014년~2017년: Carnegie Mellon University 전기컴퓨터공학과 박사후연구원
2013년 8월: 서울여자대학교 일반대학원 컴퓨터공학과 이학박사
2007년 2월: 서울여자대학교 정보통신대학 정보보호공학과, 컴퓨터공학과 공학사
〈관심분야〉 사이버 보안, 클라우드 보안, 사물인터넷 보안, 사이버 수사, 디지털 트윈