손 용 하 교수 지도

석사학위 청구논문

AWS 스토리지 서비스의 보안 강도 및 성능 분석

2025

성신여자대학교 일반대학원

미래융합기술공학과

박 영 선

AWS 스토리지 서비스의
보안 강도 및 성능 분석

손 용 하 교수 지도

이 논문을 석사학위논문으로 제출함

2025년 5월

성신여자대학교 일반대학원
미래융합기술공학과
박 영 선

인 준 서

박영선의 석사학위 논문으로 인준함

2025년 6월

심사위원장 최 현 우 (서명 또는 인)

심 사 위 원 손 용 하 (서명 또는 인)

심 사 위 원 김 성 민 (서명 또는 인)

성신여자대학교 일반대학원

논문 개요

클라우드 컴퓨팅은 기업, 공공기관, 개인 등을 막론하고 활용되고 있는 주요 기반 기술 중 하나이다. IT 자원을 유동적으로 할당받고, 그에 기반한 비용을 지불하는 형태로 편리하게 사용할 수 있다. 하지만 클라우드 서비스 제공 회사에서 자원을 제공받기 때문에 사용자가 모든 것을 제어할 수 없다는 문제가 있다. 또한 기존 온프레미스 환경과는 다른 보안 위협이 발견되고 있으며, 특히 스토리지 서비스에서 잘못된 접근 제어 설정으로 인한 데이터 유출 사고 등이 지속적으로 발생하고 있다. 퍼블릭 접근 허용, 암호화 설정 누락, 과도한 권한 부여 등의 구성 실수로 정보 유출 등의 치명적인 피해로 이어질 수 있어 클라우드 환경에 적절한 보안 설정이 요구하다. 따라서 본 연구는 AWS의 스토리지 서비스인 S3의 보안 설정에 대해 알아본다. 각 설정의 기능과 효과를 분석하고, 디지털 포렌식을 통해 데이터의 유출 가능성을 확인한다. 이를 토대로 보다 안전한 S3 사용을 위한 구성 방안을 제안하고자 한다.

목 차

표 차 례

그림 차례

제 Ⅰ 장 서론

디지털 사회에서 클라우드 컴퓨팅은 다양한 기술의 중심 요소로 자리 잡으며 많은 분야에서 폭넓게 활용되고 있다. 클라우드의 활용으로 사용자들은 시간과 장소에 제약 없이 접근할 수 있게 되었다. 특히, 코로나19의 확산으로 인한 거리두기 제한과 물리적인 이동의 제한을 극복하고자 여러 디지털 기술을 활용하는 사례가 급격히 증가했다. 이 과정에서 컴퓨팅 자원을 유동적으로 사용할 수 있는 클라우드 서비스가 주목받았다. 또한 우리나라를 비롯한 미국 등 여러 나라에서도 기존 온프레미스에서 클라우드 서비스로의 전환을 추진하고 있어 도입이 활발히 이뤄지고 있다. 기업들도 플랫폼과 서비스를 제공할 때 클라우드 서비스를 기반으로 사용자가 편리하게 접근할 수 있도록 구성하고 있다. 클라우드 컴퓨팅은 IT 자원을 필요한 만큼 제공받고 사용한 만큼 비용을 지불하는 시스템으로 확장성, 접근성, 비용 절감 등 다양한 장점을 가지고 있는 서비스이다. 하지만 사용자가 모든 것을 제어할 수 없다는 단점과 이로 인해 발생하는 다양한 보안 문제가 있다. 클라우드는 자체적으로 가지고 있는 기술의 특성으로 인해 기존의 전통적인 보안 대책을 적용하는 것은 적절하지 않다. 특히 인터넷을 통해 접속할 수 있고, 광범위하게 사용되기 때문에 클라우드 보안은 필수적인 요소이다. 클라우드 보안 위협에는 잘못된 구성, 안전하지 않은 API 사용, 내부자 위협, 과도한 권한 부여 등이 있다. 2022년 7월에는 아마존 AWS(Amazon Web Services)의 S3 버킷 설정 오류로 인해 콜롬비아와 페루의 공항 네 군데를 포함한 공항과 관련된 데이터 3TB가 외부로 유출되는 사고가 있었다. 또한 2022년 10월 마이크로소프트의 Azure 스토리지 설정 오류로 인해 Azure Blob 스토리지 버킷이 유출되면서 6만 5천 개 기업과 관련된 데이터가 유출되기도 했다. 이 외에도 다양한 경로로 사고가 발생하고 이로 인한 피해의

규모도 크다. 특히, 스토리지와 관련된 사고에서는 기업 내부 기밀 유출이나 고객들의 민감 정보가 유출될 수도 있기 때문에 중요도가 높다. 따라서 본 논문에서는 클라우드 서비스 중 AWS의 스토리지 서비스 보안 설정에 대해 분석하고자 한다. 보안 설정의 종류와 기능을 알아보고, 각각의 설정의 보안 강도를 비교한다. 활성화했을 때의 소요 시간을 비교하고, 디지털 포렌식을 통해 설정이 다를 때 노출되는 정보가 있는지 확인한다. 결과를 종합해 보안 설정에 따른 성능을 분석하고자 한다.

1. 연구 방법 및 구성

본 논문은 아마존 AWS 클라우드 서비스 중 스토리지 서비스인 S3의 보안 설정을 분석하고 이에 따른 성능을 비교해 보다 유리한 설정 방안을 제시하고자 한다. 제1장 서론에서는 클라우드 서비스가 다양한 분야에 사용되고 있으며 이로 인해 새로운 보안 문제가 발생한 것에 주목하고, 본 연구 방법에 대해 설명한다. 제2장에서는 클라우드의 정의 및 비교, 클라우드 환경에서 발생하는 문제 및 보안 조치를 기술하고, 디지털 포렌식의 개념에 대해 알아본다. 제3장에서 AWS 서비스 중 S3 스토리지 서비스 내 보안 설정에 따른 실험의 방법과 결과를 서술한다. 설정에 따라 소요되는 시간을 비교하고, 디지털 포렌식을 통해 정보의 유출 여부를 분석한다. 마지막으로 제4장에서는 실험에 대한 결론과 시사점을 정리한다.

제 Ⅱ 장 배경 지식

1. 클라우드 컴퓨팅

먼저 온프레미스(On-Premise)는 소프트웨어, 서버, 장비 등을 클라우드와 같은 원격 환경이 아닌 자체적으로 가지고 있는 시설에 직접 설치해 운영하는 방식이다[1]. 클라우드 컴퓨팅 기술이 발전하기 이전에 주로 사용되었던 방식으로 현재는 클라우드와 병행하여 인프라를 구축하는 형태도 많이 활용되고 있다. 클라우드 컴퓨팅은 인터넷을 통해 클라우드라는 가상의 공간에 있는 네트워크, 데이터베이스, 스토리지, OS, CPU 등과 같은 컴퓨팅 자원에 대해 언제 어디서든 접근할 수 있게 주문형 접근을 가능하게 하는 온디멘드(On-Demand Availability) 서비스 모델이다[2]. 온프레미스는 서버나 소프트웨어 등 필요한 것들을 직접 구매해야 하므로 초기에 발생하는 비용이 높고, 인프라를 구축하는데 시간도 상당히 투자해야 한다. 반면, 클라우드 컴퓨팅은 컴퓨팅 자원을 필요한 만큼 제공받고 사용한 만큼 비용을 지불하는 형태로 비용을 절감할 수 있다. 그리고 원하는 인프라를 더 빠르고 유연하게 구현할 수 있다. 하지만 정보가 클라우드 서비스 회사에 저장되어 있기 때문에 위험 요소가 있고, 환경 설정에 대한 전반을 사용자가 제어할 수 없다는 블랙박스 문제가 있다. 그렇기 때문에 중요 정보는 온프레미스에 저장하고, 그 외의 정보는 클라우드에 저장하는 하이브리드 형태의 모델이 많이 사용되고 있다.

미국 국가표준기술연구소(NIST, National Institute of Standards and Technology)에서 제시하는 서비스 모델은 세 가지가 있다[3]. 표 1은 NIST에서 제시한 정의를 기반으로 작성한 것으로 클라우드 서비스 회사에서 관리 및 책임의 범위에 따라 구분된다.

분류	내용
IaaS (Infrastructure as a Service)	소비자에게 제공되는 기능은 처리, 저장소, 네트워크 및 기타 기본적인 컴퓨팅 자원을 프로비저닝(할당)할 수 있는 것이며, 이를 통해 운영체제와 애플리케이션을 포함한 임의의 소프트웨어를 배포하고 실행할 수 있다. 소비자는 기저 클라우드 인프라를 관리하거나 제어하지는 않지만, 운영체제, 저장소, 배포된 애플리케이션에 대해서는 제어할 수 있으며, 호스트 방화벽과 같은 일부 선택된 네트워크 구성 요소에 대해서도 제한적으로 제어할 수 있다.
PaaS (Platform as a Service)	소비자에게 제공되는 기능은 공급자가 지원하는 프로그래밍 언어, 라이브러리, 서비스 및 도구를 사용하여 개발하거나 획득한 애플리케이션을 클라우드 인프라에 배포하는 것이다. 소비자는 네트워크, 서버, 운영체제, 저장소 등 기저 클라우드 인프라를 관리하거나 제어하지 않지만, 배포된 애플리케이션과 애플리케이션을 호스팅하는 환경의 구성 설정에 대해서는 제어할 수 있다.
SaaS (Software as a Service)	소비자에게 제공되는 기능은 클라우드 인프라에서 실행되는 공급자의 애플리케이션을 사용하는 것이다. 이러한 애플리케이션은 웹 브라우저(예: 웹 기반 이메일)와 같은 얇은 클라이언트 인터페이스나 프로그램 인터페이스를 통해 다양한

	클라이언트 장치에서 접근할 수 있다. 소비자는 네트워크, 서버, 운영체제, 저장소를 포함한 기저 클라우드 인프라를 관리하거나 제어하지 않으며, 개별 애플리케이션의 기능조차 제어하지 않는다. 단, 사용자 맞춤형 애플리케이션 설정과 같이 일부 제한적인 설정은 예외적으로 가능할 수 있다.

표 1 NIST에서 제시하는 3가지 서비스 모델

이러한 클라우드 컴퓨팅을 제공하는 회사는 크게 Amazon Web Service(AWS), Microsoft Azure, Google Cloud Service(GCP) 등이 있고, 국내에는 대표적으로 Naver Cloud, KT Cloud, 삼성 SDS 등이 있다. 2024년 클라우드 인프라 서비스 지출은 $330.4B로 전년 대비 20% 증가했고, 2025년에는 19% 더 성장할 예정이다.

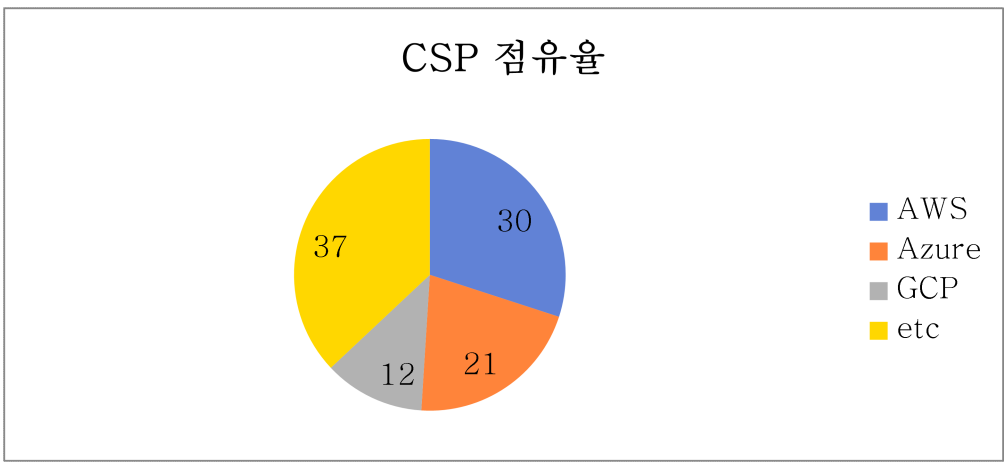

그림 1 글로벌 CSP 점유율

생성형 AI의 확산, 코로나19로 인한 디지털 전환 가속화와 미국 및 국내

공공기관의 기존 시스템을 클라우드 환경으로 전환하는 사업 등이 이뤄졌다. 이로 인해 클라우드의 사용률이 증가했고, 서비스 시장 또한 급격하게 성장하고 있다[4]. 이렇게 해외와 국내 모두 클라우드 서비스를 많이 사용하는 만큼 사고도 빈번하게 발생하고 있다. 2019년 7월에 미국의 금융 기관 Capital One에서 AWS 방화벽 설정 오류로 인해 약 1억 명의 고객 정보가 유출되는 사고가 있었다. 이름, 주소, 전화번호 등 민감한 개인정보가 노출되었다. 2022년 7월에 보안 연구팀 SafetyDetectives는 콜롬비아와 페루의 최소 4개 공항과 관련된 약 3TB의 민감 정보가 공개적으로 접근 가능한 AWS S3 버킷에 저장되어 있는 것을 발견했다. 공항 직원의 개인 식별 정보, 신분증 사진, 직업 정보 등이 노출되었다. 그리고 2022년 10월에는 보안 기업 SOCRader가 Microsoft Azure의 Blob Storage의 설정 오류로 약 2.4TB의 데이터가 외부에 공개적으로 접근이 가능한 상태인 것을 발견했다. 약 65,000개 이상의 기업과 관련된 이름, 이메일 주소, 전화번호, 계약 문서 등이 노출된 상태였다. 이처럼 다양한 설정 오류와 관련된 사고가 지속적으로 발생하고 있다. 클라우드 보안 위협에는 보안 체계와 전략 미흡, 클라우드 데이터 노출, 인프라 확장과 소프트웨어 공급 증가 등이 있다[5]. 이러한 사고를 방지하기 위해서 여러 기관에서 보안 권고 및 가이드라인을 제공하고 있다. 국내의 한국인터넷진흥원(KISA)에서는 안정성과 신뢰성이 검증된 클라우드 서비스를 공급하며 보안 문제에 대한 우려를 해소하며 서비스 경쟁력을 확보하기 위해서 클라우드 보안인증제(CSAP, Cloud Security Assurance Program)를 운영하고 있다. CSAP 인증 기준은 크게 관리적 보호 조치, 기술적 보호조치, 물리적 보호조치 그리고 국가 기관 등이 이용하는 클라우드 컴퓨팅 서비스 보호 조치로 나눠지며 상세한 항목은 다음과 같다. 인증 기준은 IaaS, PaaS, SaaS, DaaS가 있고 기준에 따라 인증 항목이 달라진다[6].

통제 분야	통제 항목
1. 정보보호 정책 및 조직	1.1. 정보보호 정책
	1.2. 정보보호 조직
2. 인적보안	2.1. 내부인력 보안
	2.2. 외부인력 보안
	2.3. 정보보호 교육
3. 자산관리	3.1. 자산 식별 및 분류
	3.2. 자산 변경관리
	3.3. 위험관리
4. 서비스 공급망 관리	4.1. 공급망 관리정책
	4.2. 공급망 변경관리
5. 침해사고관리	5.1. 침해사고 절차 및 체계
	5.2. 침해사고 대응
	5.3. 사후관리
6. 서비스 연속성 관리	6.1. 장애대응
	6.2. 서비스 가용성
7. 준거성	7.1. 법 및 정책 준수
	7.2. 보안 감사
8. 물리적 보안	8.1. 물리적 보호구역
	8.2. 정보처리 시설 및 장비 보호
9. 가상화 보안	9.1. 가상화 인프라
	9.2. 가상 환경
10. 접근통제	10.1. 접근통제 정책
	10.2. 접근권한 관리
	10.3. 사용자 식별 및 인증
11. 네트워크 보안	11. 네트워크 보안
12. 데이터 보호 및 암호화	12.1. 데이터 보호
	12.2. 매체 보안
	12.3. 암호화
13. 시스템 개발 및 도입 보안	13.1. 시스템 분석 및 설계
	13.2. 구현 및 시험
	13.3. 외주 개발 보안
	13.4. 시스템 도입 보안
14. 공공부문 추가 보안요구 사항	14. 공공부문 추가 보안요구 사항

표 2 KISA CSAP 인증 기준

Avneet Kaur 외 3인은 AWS와 IBM 클라우드 서비스를 대상으로 성능과

보안성을 비교하는 연구를 진행했다. Phoronix를 사용해 웹서버 요청 처리 속도, 데이터베이스 성능, 메모리 속도 등을 테스트하고 암호화 알고리즘 등을 비교했다[7].

Wenting Shen 외 5인은 클라우드 스토리지를 안전하게 사용하기 위해서 개인키의 저장 없이 데이터 무결성 감사를 수행하는 체계를 제안했다. 이를 위해서 생체 정보를 사용자의 fuzzy key로 활용해 인증 및 무결성 검증을 가능하게 했다[8].

Ferrag 외 4인은 시큐어 네트워크 코딩 방식을 활용해 동적 데이터에 알맞은 보안 클라우드 스토리지 프로토콜을 제안했다. Secure Network Coding, Homomorphic Signature, Skip List 방식 등을 활용했다[9].

2. 디지털 포렌식

디지털 포렌식은 컴퓨터, 모바일 기기 그리고 네트워크 장비 등과 같은 다양한 디지털 기기로부터 데이터를 수집하고 분석해 범죄 수사에 필요한 증거를 찾아내며, 이를 법적 절차에 활용하는 과정을 말한다[10]. 과거에는 주로 PC나 서버의 하드디스크를 분석하는 '디스크 포렌식'에 집중되었으나, 이제는 스마트폰, 클라우드, 사물인터넷(IoT) 기기, 차량 등 인터넷과 연결된 거의 모든 디지털 장치가 잠재적 분석 대상이 되었다[11]. 특히 해킹이나 정보 유출 등의 보안 사고 발생 시 행위 분석의 중요한 과정이 되었다. 디지털 포렌식은 추후 재판 등에 증거로써 활용되기 때문에 수집 및 분석 과정에서도 다음의 원칙을 철저히 준수해야 한다.

• 정당성의 원칙

디지털 증거물을 수집할 때 정당한 방식으로 증거를 수집해야 한다. 불법으로 수집할 경우, 증거로써의 효력을 잃게 된다.

• 무결성의 원칙

수집된 증거가 분석 및 제출하는 과정에서 위조 또는 변조되지 않았음을 증명할 수 있어야 한다. 주로 증거의 해시(Hash)값을 사용한다.

• 재현의 원칙

동일한 조건에서 동일한 결과가 도출되어야 한다. 조사를 수행했던 하드웨어와 소프트웨어의 버전, 구성 등이 모두 동일한 상황을 말한다.

• 신속성의 원칙

휘발성 증거를 수집하기 위해 지체 없이 진행해야 한다. 메모리 같은 경우, 기기가 종료되면 저장되어 있던 정보가 사라지기 때문에 이를 확보하기 위해서 빠르게 수집 과정이 진행되어야 한다.

• 절차 연속성의 원칙

증거의 수집과 보관, 분석 및 제출 등의 과정을 모두 기록하여 검증의 수단을 제공한다.

위의 5가지 원칙을 지키지 않을 경우, 재판 등에서 증거로 받아들여지지 않을 수 있어 신중하게 증거를 수집해야 한다. 증거로 사용하지 않더라도 이렇게 증거를 수집 및 분석한다면 결과에 대한 신뢰성을 얻을 수 있다.

디지털 포렌식의 종류에는 인터넷 포렌식, 시스템 포렌식, 네트워크 포렌식, 디스크 포렌식, 메모리 포렌식 등이 있다.

• 디스크 포렌식은 물리적인 저장 장치로부터 증거를 수집하는 방식으로

하드 디스크나 플로피 디스크 등에서 확보한다.

• 시스템 포렌식은 운영체제, 응용 프로그램, 프로세스 등을 분석하며 Windows, Linux, NTFS, FAT32 등에서 정보를 확인한다.

• 네트워크 포렌식은 네트워크를 통해 전송되는 데이터나 암호를 분석하는 것이며 패킷 분석, 에러 로그, IP 정보, 평문 분석 등을 통해 수집한다.

• 인터넷 포렌식은 WWW, FTP 등과 같은 인터넷 응용 프로토콜을 활용하는 범위 내에서 증거를 수집하게 된다. 웹 히스토리나 메일 헤더 또는 IP 정보 등을 분석한다.

• 메모리 포렌식은 메모리에 로드되는 정보나 휘발성 정보를 수집해 분석한다. 이를 통해 얻을 수 있는 정보는 네트워크 연결 정보, 악성코드 확인, 프로세스 정보, 사용자 활동 정보 등이 있다.

클라우드 컴퓨팅의 발전으로 클라우드 환경에서 발생하는 범죄가 증가하는 추세이다. 하지만 클라우드는 가상화 기반 환경이라는 점과 대용량 데이터를 저장하고 있는 특성이 있어 기존의 포렌식 방식만으로는 완벽하게 분석이 어려워졌다[12]. 또한 표 1의 클라우드 서비스 모델에 따라서도 포렌식이 다르게 진행될 수 있다[13]. 주로 로그를 수집하고 이벤트를 추적하는 서비스를 이용해 데이터를 수집한 후 이를 분석하게 된다. API 호출 내역, 명령어 내역, 권한 변경, 리소스 상태 변경, 이상 행위 탐지 등을 확인할 수 있다.

Zayyan Umar 외 3인은 이기종 클라우드 환경에서 디지털 포렌식을 위한 프레임워크를 제안했다. 타 CSP 간의 비호환성과 다양한 환경으로 인해 디지털 포렌식 분석이 어려움을 극복하기 위해 중앙 클라우드에 여러 CSP를 연결해 로그를 수집하고, 분석하도록 설계했다[14].

Sankar Thamburasa 외 5인은 IDrive, Mega 클라우드 스토리지 서비스에 대한 디지털 포렌식 흔적을 연구했다. Windows 7 가상 환경에서 RAM, 레지스트리, 로그 파일 등을 활용해 사용자 정보와 흔적을 분석했다[15].

Sameera Almulla 외 2인은 클라우드 내 VM 스냅샷 환경에서 디지털 포렌식을 수행하는 모델을 제안했다. Citrix XenServer 스냅샷을 재구성 없이 직접 분석 가능한 모델을 제안했으며, 도구를 활용해 증거를 수집하고 무결성 검증을 수행했다[16].

제 Ⅲ 장 연구 설계

1. AWS 서비스

AWS는 Amazon의 클라우드 컴퓨팅 서비스로 전 세계에서 가장 많은 점유율을 차지하고 있다. AWS는 클라우드 환경에서 보안 강화 및 유지를 위해 AWS Well-Architected Framework의 보안 원칙과 AWS IAM(Identity and Access Management)을 통한 최소 권한 원칙 적용 등 다양한 가이드와 모범 사례를 제공하고 있다[17]. 또한 서비스별로 자세한 설명과 사례를 포함한 공식 문서와 다양한 가이드라인을 제공해 사용자가 안전하게 운영할 수 있도록 지원하고 있다. AWS에는 크게 컴퓨팅, 네트워킹, 스토리지, 데이터베이스, 보안 등의 서비스가 있고 아래의 표는 주요한 서비스를 간단히 정리한 것이다.

분류	서비스 이름	설명
컴퓨팅	EC2 (Elastic Compute Cloud)	가상 서버를 생성해 애플리케이션을 실행할 수 있는 핵심 컴퓨팅 서비스
	Lambda	서버를 직접 관리하지 않고 코드를 이벤트 기반으로 실행하는 서버리스 서비스
	ECS / EKS	Docker 및 Kubernetes 기반 컨테이너를 실행하고 관리하는 서비스
스토리지	S3 (Simple Storage Service)	확장 가능한 객체 스토리지로 다양한 데이터를 저장하고 인터넷으로 접근 가능
	EBS	EC2에 연결하는 블록 스토리지로 디

	(Elastic Block Store)	스크처럼 사용됨
	Glacier	장기 백업/보관용 저비용 스토리지
네트워킹	VPC (Virtual Private Cloud)	격리된 가상 네트워크 환경을 구성하는 서비스
	Route 53	도메인 이름 등록 및 DNS 트래픽 라우팅 서비스
	CloudFront	정적·동적 콘텐츠를 빠르게 전송하는 글로벌 CDN 서비스
데이터베이스	RDS (Relational Database Service)	관계형 데이터베이스를 자동 백업/패치 포함한 관리형 서비스로 제공
	DynamoDB	서버리스 NoSQL 데이터베이스로 빠른 읽기/쓰기 성능 제공
	Redshift	대규모 데이터 분석을 위한 데이터 웨어하우스 서비스
보안/관리	IAM (Identity and Access Management)	사용자/역할 기반 권한 제어 및 인증 관리 서비스
	CloudTrail	계정 내 API 호출과 활동 로그를 기록하는 서비스
	CloudWatch	모니터링 및 경보 설정이 가능한 통합 관찰 서비스
개발 도구	CodeBuild / CodeDeploy / CodePipeline	CI/CD 자동화 파이프라인 구성 도구
	Cloud9	웹 기반 통합 개발 환경(IDE)

AI/ML	SageMaker	머신러닝 모델의 학습, 배포, 추론을 지원하는 통합 ML 플랫폼
	Rekognition	이미지/영상에서 객체 감지, 얼굴 인식 등의 기능을 제공하는 AI 비전 서비스
메시징	SNS (Simple Notification Service)	이메일, SMS, 모바일 푸시 등으로 알림 전송 가능한 메시징 서비스
	SQS (Simple Queue Service)	비동기 메시지 처리를 위한 안정적인 메시지 큐 서비스
보안/암호화	KMS (Key Management Service)	암호화 키를 생성, 관리, 제어할 수 있는 중앙 집중식 키 관리 서비스로 S3, EBS, RDS 등 다양한 AWS 서비스와 연동됨
	CloudHSM	하드웨어 보안 모듈(HSM)을 통해 FIPS 140-2 수준의 강력한 키 보호 기능을 제공하는 암호화 전용 서비스
	Secrets Manager	데이터베이스 자격 증명, API 키 등 민감 정보를 안전하게 저장하고 주기적으로 자동 교체할 수 있도록 지원하는 비밀 관리 서비스
	AWS Certificate Manager (ACM)	SSL/TLS 인증서를 손쉽게 프로비저닝하고 관리할 수 있는 서비스로, 암호화된 HTTPS 통신 보장을 위해 사용됨

표 3 AWS 주요 서비스

여러 서비스 중 스토리지 서비스를 좀 더 자세히 살펴보면 S3(Simple Storage Service), EFS(Elastic File System), EBS(Elastic Block Storage) 등이 있다. S3는 인터넷 엑세스가 가능한 객체 스토리지 서비스로 사용자가 쉽게 접근할 수 있다. Bucket 단위로 구분하고 Bucket 밑에 Object를 업로드하는 구조이다. 그 과정에서 해당 Object의 접근 권한 설정을 위해 ACL(Access Control List)을 설정한다[18]. EFS는 EC2 인스턴스에서 동시에 마운트 가능한 NFS(Network File System) 기반의 파일 스토리지이고, 여러 인스턴스 간의 파일 공유 시스템이 필요한 경우 사용한다. EBS는 EC2 인스턴스에 연결해 사용하는 디스크 형태의 블록 스토리지이다. 단일 EC2에 종속된다는 점이 EFS와의 차이이다. 본 논문에서는 여러 사용자가 접근할 수 있는 S3 서비스를 기준으로 알아보고자 한다.

S3 서비스에서는 Bucket을 생성할 때 Region을 결정할 수 있다. 또한 버전 관리, 정적 웹 호스팅 등이 가능하다. Object를 보호하기 위해 암호화를 지원하고, 정책 기반의 접근 제어를 통해 권한 없는 사용자의 접근을 방지할 수 있다. 암호화는 서버 측 암호화(SSE, Server Side Encryption)와 클라이언트 측 암호화(Client Side Encryption)가 있다. 서버 측 암호화는 객체를 저장하기 전에 AWS가 자동으로 암호화하고, 접근 시 복호화하게 된다. SSE-S3는 AWS가 관리하는 키를 사용해 암호화하는 방식으로 설정만 하면 자동으로 적용된다. 2023년부터 모든 새로운 S3 Object를 업로드할 때 기본적으로 적용된다. SSE-KMS는 AWS의 KMS(Key Management Service) 서비스를 사용해 사용자가 관리하는 키로 암호화를 적용한다. SSE-C는 사용자가 직접 제공하는 키로 암호화 하고, 키가 AWS에 저장되지 않기 때문에 요청이 있을 때마다 직접 키를 제공해야 한다. 클라이언트 측 암호화는 사용자가 Object를 업로드하기 전에 직접 암호화하고 이를 업

로드하는 방식이다. 본 논문에서는 사용자가 직접 키를 관리하고, 적용할 수 있는 SSE-KMS 암호화를 활용한다.

KMS 서비스는 데이터를 암호화할 때 사용되는 키를 관리하는 서비스로 키를 생성하고 적용해 데이터의 보안을 향상하는 데 사용된다. 키를 생성하는 유형은 3가지로 첫 번째는 AWS managed Key이다. AWS 서비스가 KMS를 통해서 Key를 서비스받는 것으로 내부에서 자동으로 이뤄지기 때문에 사용자는 제어할 수 없다. 두 번째 유형은 Customer Managed Key로 사용자가 직접 Key를 생성 및 관리할 수 있고, IAM 권한을 받으면 제어가 가능하다. 마지막으로 Custom Key Stores라는 AWS의 또 다른 서비스인 CloudHSM을 활용한 Key 관리 서비스가 있다. Custom Key Stores 방식보다 Customer Managed Key가 더 관리가 편리하고 비용도 상대적으로 저렴하므로 Customer Managed Key 방식을 사용하고자 한다.

S3에서 Object를 업로드할 때 암호화 외에 ACL 설정을 추가로 할 수 있다. S3 버킷이나 객체에 대한 접근 권한을 부여하는 방식으로 사용자별 읽기 및 쓰기 권한을 설정할 수 있다. ACL을 활성화하고 퍼블릭 엑세스를 허용하면 객체 URL을 통해 누구나 접근이 가능하다.

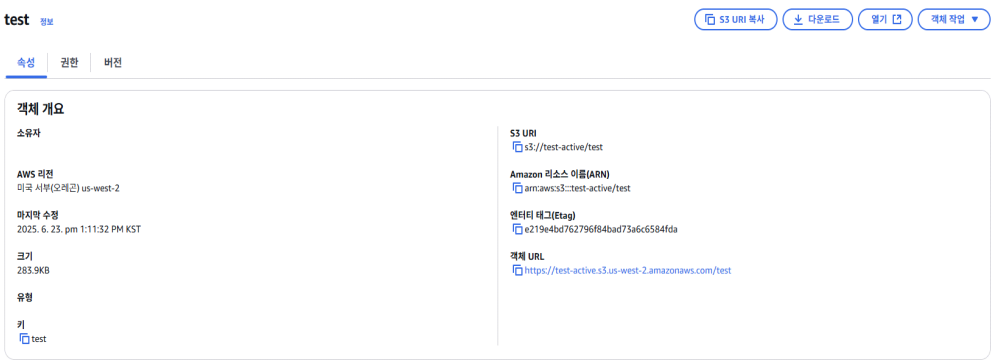

그림 2 S3 객체 URL

비활성화하면 권한을 가진 사용자만이 접근할 수 있다. 또한 퍼블릭 엑세스 차단 설정을 제공해 퍼블릭 접근 여부를 설정할 수 있다. 아래와 같이 체크박스 형태로 원하는 강도를 선택할 수 있다.

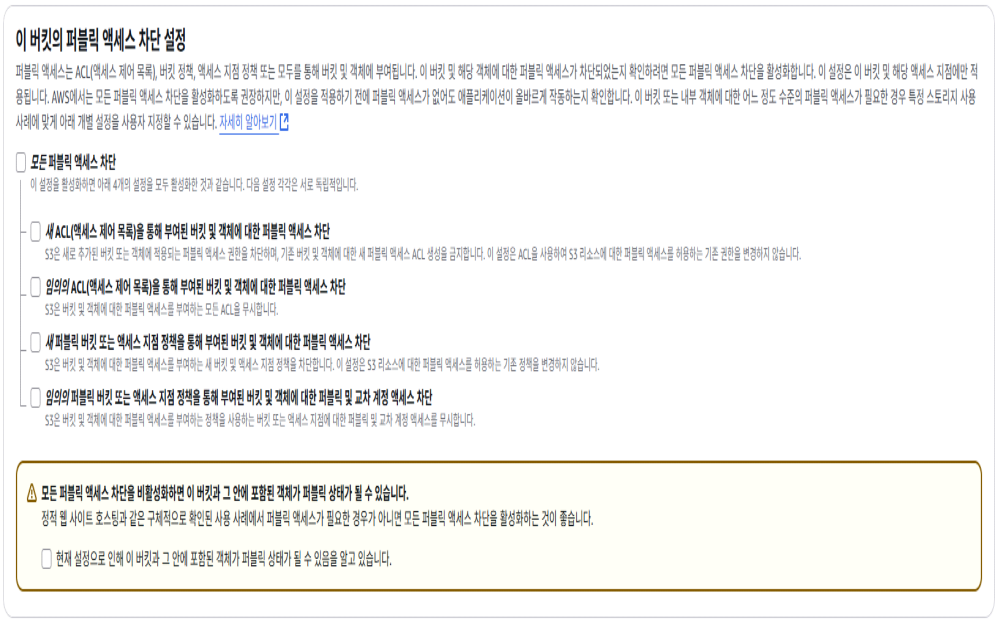

그림 3 S3 Bucket 퍼블릭 엑세스 설정

S3 서비스를 통해 실험에 필요한 보안 설정을 확인해 보았고, 추가로 AWS의 CloudTrail을 디지털 포렌식에 활용할 수 있다. AWS 내에서 발생하는 모든 API 호출과 계정 활동을 기록하는 서비스로, 활성화하면 보안 감사, 모니터링, 디지털 포렌식 분석 등에 사용할 수 있다. Trail을 생성할 때 기록할 API 활동을 선택하고, 리소스 유형에서 원하는 서비스를 선택하면 된다. 이후에 이벤트 기록에 활동한 로그가 남으면 이를 분석할 수 있다. GetObject, PutObject, Decrypt, RunInstance 등의 API 호출과 eventTime, eventName, accountID, aclRequired 등 이벤트 필드 내용을 통해 추적할 수 있다.

2. AWS Architecture 구성

AWS S3 내에서 KMS와 ACL은 보안을 강화하는 데 사용되지만 각각의 연구가 부족한 실정으로, 본 연구에서는 클라우드 환경에서 ACL과 KMS가 제공하는 보안 강도를 비교해 실제 환경에 적용했을 때 차이가 발생하는지 비교한다. AWS Architecture는 S3와 EC2로 구성했고, S3에 접근하는 경로는 두 가지로 설계했다. 사용자가 객체 URL을 사용해 외부에서 직접 S3에 접근하는 경로와 사용자가 EC2 기반 웹서버를 통해 간접적으로 접근하는 경로이다. 버지니아 북부 리전에 S3 버킷과 EC2 인스턴스를 생성하고 웹서버를 구성해 S3에 요청을 보내는 방식으로 구현했다. S3 버킷에는 ACL과 KMS를 각각 다르게 적용한 비교군을 업로드했다. 해당 AWS Architecture와 S3 실험 비교군은 조경현 외 3인의 Amazon S3 제로 트러스트 모델 설계 및 포렌식 분석 논문의 구성을 참고하여 동일하게 구성하였다[19].

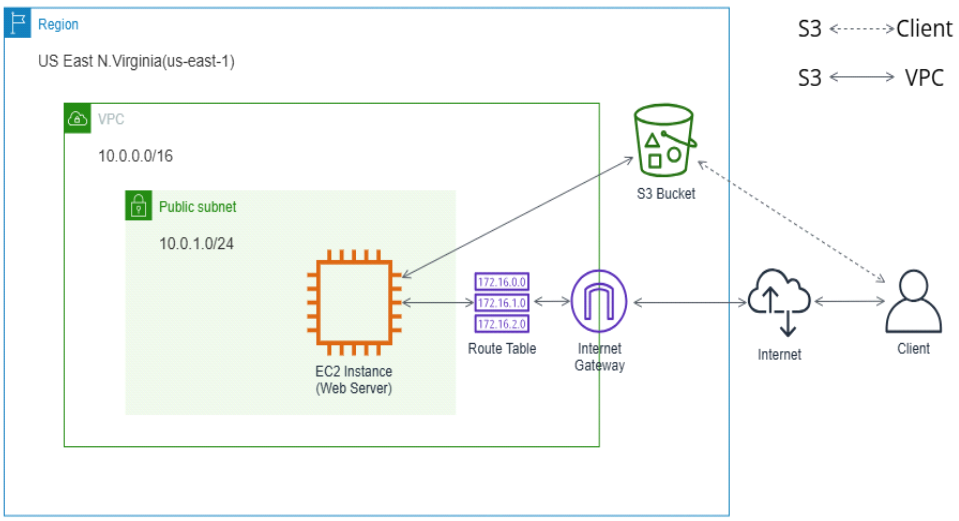

그림 4 AWS Architecture 구성([19]의 그림1에서 재인용)

비교군은 ACL 설정 활성화 여부, KMS 설정 활성화 여부, 리전에 따라

크게 분류한다. 오레곤과 버지니아 북부 리전을 사용한다. 버킷 생성 시에 'ACL 활성화됨'과 '버킷 소유자 선호'를 선택해 ACL을 활성화한다. 객체 업로드 시에 ACL 설정에서 '사전 정의된 ACL에서 선택'과 '퍼블릭 읽기 액세스 권한 부여'를 선택해 객체 ACL을 활성화한다. KMS는 객체 업로드 시에 서버 측 암호화 설정에서 '암호화 키 지정', '기본 암호화 버킷 설정 재정의', 'AWS Key Management Service 키를 사용한 서버 측 암호화(SSE-KMS)', 'AWS KMS 키 중에서 선택' 후 사전에 KMS에서 생성한 키를 선택해 암호화를 적용한다. Access 방식 중 Client는 S3 객체 URL을 통해 접근하는 루트이고, Cloud는 웹서버 홈페이지에 접근해 버튼 클릭을 통해 S3에 연동되어 파일을 다운로드 받을 수 있는 루트이다. Forensic Location에서 Local은 파일을 다운로드 받은 노트북 기기에서 포렌식을 진행한다. Web은 웹서버를 올린 EC2를 포렌식한다. CloudTrail 분석은 Forensic Location에 상관없이 AWS 내에서 로그를 분석하는 방식으로 진행한다.

Scenario	ACL		KMS	Access		Forensic Location	
	Bucket	Object		Client	Cloud	Local	Web
ACL1	O	O	-	O	-	O	-
ACL2	O	-		O	-	O	-
ACL3	-	-		O	-	O	-
ACL4	-	-		-	O	-	O
KMS1	O	O	O	O	-	O	-
KMS2	-	-		-	O	O	-
KMS3	-	-		-	O	-	O

표 4 S3 실험 비교군([19]의 표1에서 재인용)

3. 연구 방법

노트북에 휴대폰 테더링을 연결한 상태와 와이파이를 연결한 상태에서의 2가지 실험을 비교한다. AWS S3에 업로드할 파일은 100MB 크기의

HWPX 파일을 활용한다. 업로드 및 다운로드에 걸리는 시간을 확인하기 위해서 네트워크 속도를 제한해 실험을 진행하며 100KB/s, 200KB/s, 300KB/s, 400KB/s, 500KB/s 별 평균 시간을 비교한다. 네트워크 속도 제한을 위해서 NetLimiter라는 오픈 소스 프로그램(5.3.19.0 버전)을 사용한다.

그리고 파일 다운로드 시 정보 유출 여부를 알아보기 위해서 디지털 포렌식을 진행한다. 웹과 네트워크 포렌식을 통해 캐시, History, Top Sites, Web Data 등과 네트워크 패킷, Access Log와 Error Log를 분석한다. DB Browser for SQLite(3.13.1 버전), Wireshark(4.2.4 버전), ChromeCacheView(2.51 버전)을 사용한다. 더불어 CloudTrail을 통해 실험 과정에서 발생한 로그들을 분석한다.

실험 환경은 노트북 1대, 휴대폰 1대, 와이파이 기기 1대로 이뤄졌으며 자세한 사양은 다음과 같다.

모델명	HFR H734GP
Wi-Fi 규격	IEEE 802.11ac (Wi-Fi 5), backward compatible with 802.11a/n
주파수	5GHz

표 5 Wi-Fi 사양

모델명	Samsung Galaxy S24 Plus
OS	Android 14(One UI 6.1)
연결 방식	Wi-Fi(2.4GHz Hotspot, Galaxy S22 via 5G)
핫스팟 주파수	2.4GHz

표 6 휴대폰 사양

모델명	Lenovo ThinkPad X1 Carbon Gen 11
CPU	Intel Core i5-1335U
RAM	16GB
OS	Windows 11 Home

표 7 노트북 사양

제 Ⅳ 장 연구 결과 및 분석

1. 성능 비교 실험

첫 번째 실험은 휴대폰 테더링을 연결해 100KB/s, 500KB/s 속도로 100MB 크기의 파일을 다운로드 받는 시간을 측정했다. NetLimiter를 사용해 컴퓨터 전체의 네트워크를 제한한다.

Scenario	소요 시간	비고
ACL1	22분 43초 60	
ACL2	0분 2초 3	접속 불가
ACL3	0분 0초 92	접속 불가
ACL4	2시간 30분	
KMS1	0분 1초 20	접속 불가
KMS2	1시간 14분 40초	
KMS3	1시간 14분 40초	KMS2와 동일

표 8 100KB/s 실험 결과

Scenario	소요 시간	비고
ACL1	4분 19초 6	
ACL2	0분 0초 90	접속 불가
ACL3	0분 0초 22	접속 불가
ACL4	3분 19초 12	
KMS1	0분 0초 7	접속 불가
KMS2	3분 21초 3	
KMS3	3분 21초 3	KMS2와 동일

표 9 500KB/s 실험 결과

실험 결과 접속이 불가한 상태에서는 1초 이내에 응답을 받았고, 평균 다운로드 시간은 500KB/s에서 3분 39초 27, 100KB/s에서는 1시간 22분 27초로 확인되었다. 100KB/s 실험에서 ACL4, KMS2, KMS3의 실험 결과가 예상 시간과 매우 큰 차이를 보였다. 속도를 더 제한한 것만으로 이렇게 큰

편차가 나타나지는 않았을 것으로 추정해 다양한 시도를 통해 원인을 분석했다. 크롬 브라우저의 flags 설정에서 다운로드와 관련된 설정을 비활성화해 다시 실험했으나 여전히 100KB/s에서 시간이 과도하게 소요되는 결과가 나타났다. 휴대폰 테더링이라는 불안정한 네트워크와 컴퓨터 전체에 속도 제한을 걸면서 크롬 브라우저의 TCP 연결 유지에 영향이 미쳤을 가능성이 있는 것으로 판단했다. 따라서 이후 실험에서는 와이파이 환경에서 진행하고, 크롬 브라우저에만 속도 제한을 거는 방식으로 변경했다.

이번 실험에서는 100KB/s, 200KB/s, 300KB/s, 400KB/s, 500KB/s 별로 각각 10회씩 100MB 크기의 파일을 다운로드 한다.

Scenario	ACL1	ACL2	ACL3	ACL4	KMS1	KMS2, KMS3	예상 시간
500KB/s	3분 22초 86	0분 0초 29	0분 0초 36	3분 20초 46	0분 0초 35	3분 21초 39	3분 20초
400KB/s	4분 14초 77	0분 0초 32	0분 0초 42	4분 36초 86	0분 0초 40	4분 11초 95	4분 10초
300KB/s	5분 40초 11	0분 0초 40	0분 0초 37	5분 37초 67	0분 0초 40	5분 37초 81	5분 34초
200KB/s	8분 31초 29	0분 0초 43	0분 0초 36	8분 29초 94	0분 0초 39	8분 28초 88	8분 20초
100KB/s	17분 5초 61	0분 0초 46	0분 0초 44	17분 1초 72	0분 0초 46	17분 1초 50	16분 40초
비고		접속 불가	접속 불가		접속 불가		

표 10 속도별 평균값 결과

실험 결과 접속 불가한 상태에서는 1초 이내에 응답을 받았고, 나머지 비교군에서는 예상값에 근사한 결과를 확인할 수 있었다. 실험 중 예상 시간보다 조금 더 오래 걸리는 경우도 있었으나 350회 중 2회 정도의 비율이므로 유의미한 영향을 끼치지 않았다. ACL과 KMS 활성화 여부에 따른 비교군을 봤을 때 크게 속도에서 차이가 나지 않았다. 특히, KMS 활성화 여부에 따라 소요 시간이 더 걸릴 것으로 예상했으나 거의 차이가 없었으므로 S3에 Object를 업로드할 때 보안 강화를 위해 KMS를 통한 암호화 적용을 권장한다.

세 번째 실험은 파일을 업로드 시 ACL 및 KMS 활성화 여부에 따라 속도 차이가 나는지 확인한다. 해당 실험은 사전에 구성한 AWS Architecture와 무관하게 모두 S3 서비스에서 직접 업로드한다. 앞선 실험에서 비교군의 속도 차이가 크게 없는 것으로 확인했으므로 이번 실험은 500KB/s 제한 상태에서만 실험를 진행한다.

	ACL1	ACL2	ACL3	ACL4	KMS1	KMS2, KMS3
1	3분 35초 67	3분 35초 50	3분 31초 10	3분 34초 74	3분 34초 47	3분 33초 20
2	3분 30초 85	3분 29초 17	3분 31초 53	3분 32초 36	3분 33초 1	3분 31초 78
3	3분 36초 81	3분 34초 58	3분 31초 16	3분 32초 99	3분 33초 83	3분 33초 22
4	3분 32초 16	3분 30초 40	3분 34초 18	3분 37초 18	3분 30초 6	3분 32초 41
5	3분 30초 80	3분 31초 12	3분 30초 98	3분 39초 88	3분 36초 28	3분 32초 70

6	3분 29초 97	3분 31초 66	3분 31초 10	3분 32초 92	3분 30초 88	3분 33초 54
7	3분 30초 63	3분 30초 84	3분 31초 43	3분 33초 38	3분 30초 43	3분 32초 74
8	3분 31초 64	3분 31초 86	3분 31초 69	3분 35초 21	3분 31초 34	3분 34초 67
9	3분 31초 32	3분 31초 23	3분 44초 27	3분 35초 23	3분 41초 21	3분 32초 66
10	3분 31초 52	3분 30초 14	4분 10초 95	3분 32초 1	3분 30초 75	3분 32초 58
평균	3분 32초 30	3분 32초 5	3분 35초 44	3분 34초 19	3분 33초 83	3분 33초 15

표 11 500KB/s 업로드 실험 결과

업로드 실험에서도 마찬가지로 ACL과 KMS 활성화 여부에 무관하게 예상 시간과 유사한 결과를 확인할 수 있었다. 다운로드와 업로드 시 ACL과 KMS를 모두 적용해도 시간에 큰 차이가 없으므로 S3에 Object를 업로드할 때는 두 가지 설정을 모두 활성화해 더 강력한 보안을 적용하는 것이 바람직하다.

2. 디지털 포렌식 분석 결과

다음은 디지털 포렌식을 통해 다운로드 과정에서 정보 유출이 있는지 확인하고자 한다.

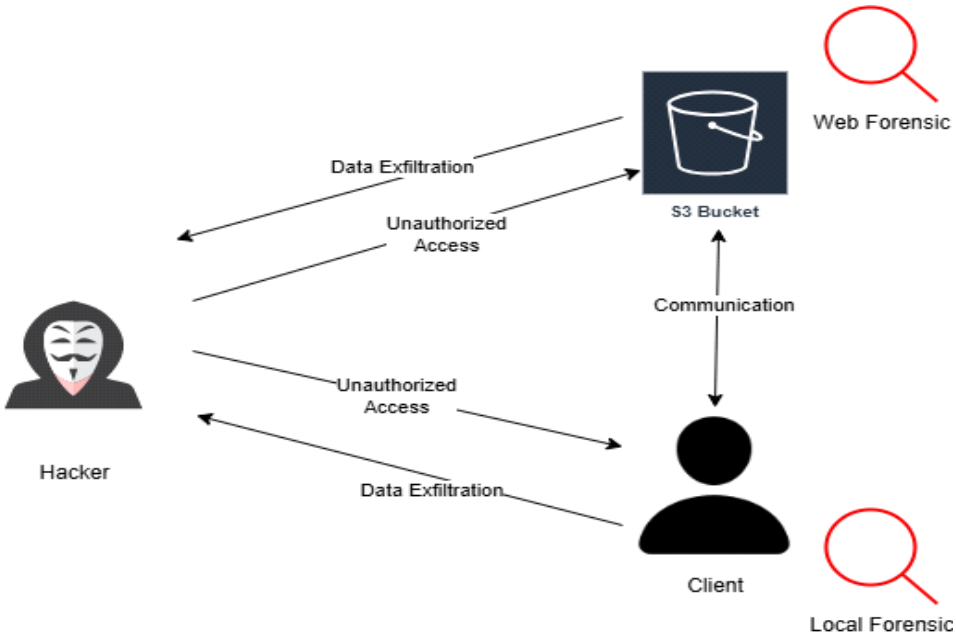

그림 5 디지털 포렌식 위협 시나리오

본 시나리오는 AWS의 S3 서비스에서 발생할 수 있는 보안 설정 실수나 미흡한 구성으로 인해 발생할 수 있는 데이터 유출 가능성을 가정한 것이다. 퍼블릭 접근 허용, 암호화 미적용, 과도한 권한 부여 등의 설정 오류가 존재할 경우, 비인가자가 EC2의 웹페이지 또는 S3 객체 URL을 통해 접근을 시도할 수 있다. 이때 버킷이나 객체에 적절한 접근 제어가 설정되지 않았을 경우, 비인가자는 객체에 접근해 다운로드 및 열람이 가능하고 민감한 정보를 획득할 수 있다. 이러한 정보는 내부 시스템 침입이나 다크웹에 정보를 판매하는 등의 상황으로 이어질 수 있다. 따라서 디지털 포렌식을 통해 침입 경로와 데이터의 유출 여부를 확인해야 한다. 웹 브라우저 캐시, 시

스템 로그, 네트워크 패킷, AWS의 CloudTrial 로그 등을 확인해 해당 내용을 파악한다.

ChromeCacheView 프로그램을 사용해 Cache Data를 분석한 결과 파일명, 접근한 URL, 파일 사이즈, 시간, 응답 내용, IP 등을 확인할 수 있었다.

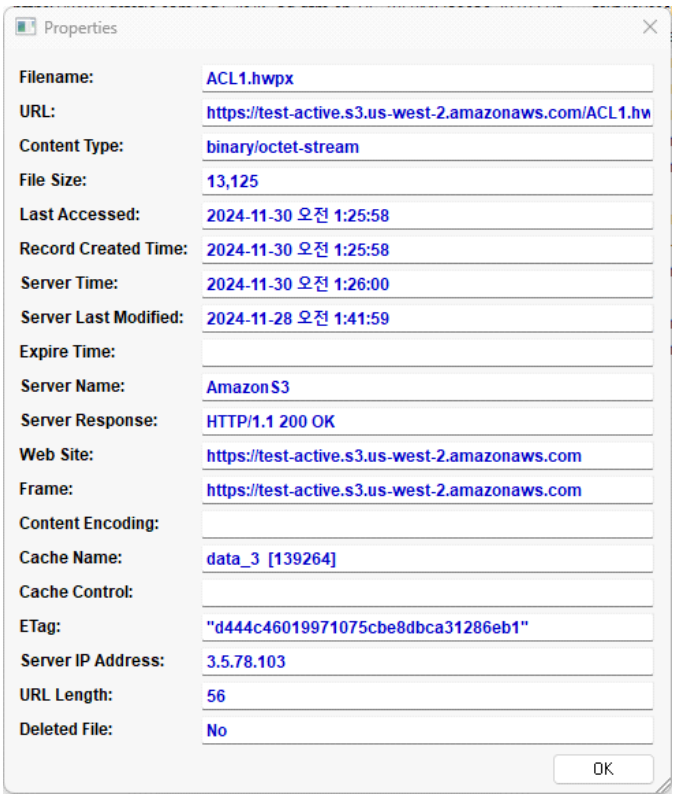

그림 6 Chrome Cache 정보

ACL1의 경우, AWS S3 객체 URL이 확인되고, 성공적으로 접근했기 때문에 200 OK 응답을 받은 것을 확인할 수 있다. 파일 생성 시간과 서버 시간 그리고 서버가 마지막으로 수정된 시간까지 알 수 있었다.

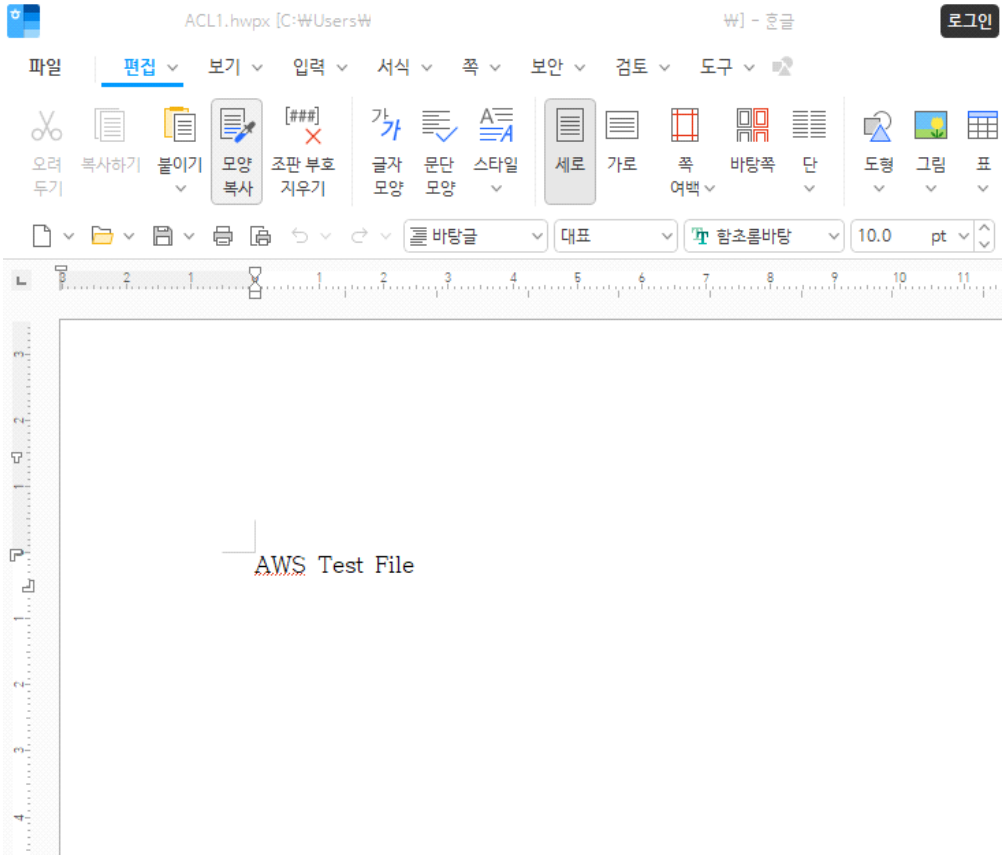

그림 7 ChromeCacheView에서 열어본 파일

사용한 프로그램에는 캐시 파일의 정보를 활용해 다운로드 받은 파일을 열어볼 수 있는 기능이 있어 파일의 내용을 확인할 수 있었다.

ACL2는 ACL1과 거의 동일한 내용을 확인할 수 있었지만 접근이 불가능해 403 Forbidden 응답이 저장된 것을 볼 수 있었다. 다운로드 받지 못했기 때문에 파일은 열어볼 수 없었다. ACL3도 동일하게 확인되었다.

그림 8 ACL4 Access Log

ACL4는 서버에서 포렌식을 진행해 Access Log와 Error Log를 확인했다. 다운로드에 성공해 200 OK 응답과 IP, 접근 URL이 기록되어 있었고, Access Log만 남았고, Error Log는 생성되지 않았다. KMS3도 동일한 내용으로 확인되었다.

KMS1은 접근 시, 'Requests specifying Server Side Encryption with AWS KMS managed keys require AWS Signature Version 4' 라는 메시지와 더불어 400 Bad Request 응답을 받았고, Error Log에도 기록되어 있었다. 이는 Signature Version 4 방식으로 요청하지 않아 접근이 거부되었다는 뜻으로, 오레곤 리전은 Version 4 방식만 허용하고 있기 때문에 거부된 것으로 분석되었다.

KMS2는 버지니아 북부 리전으로 Version 2로 접근하기 때문에 요청 방식으로 인한 거부는 일어나지 않았다. 하지만 KMS 키 권한이 부족해 403 Forbidden으로 접근이 불가했다.

다음은 Wireshark를 사용해 네트워크 패킷을 수집해 분석하는 네트워크 포렌식을 진행했다.

ACL1, ACL2, ACL3와 KMS1의 네트워크 패킷에서는 S3 URL 정보 이외에 다른 정보는 확인할 수 없었다. 반면, ACL4에서는 웹서버 HTML 코드와 S3 URL 정보, 응답 코드가 확인되었고, KMS2와 KMS3에서도 동일한 내용을 확인할 수 있었다.

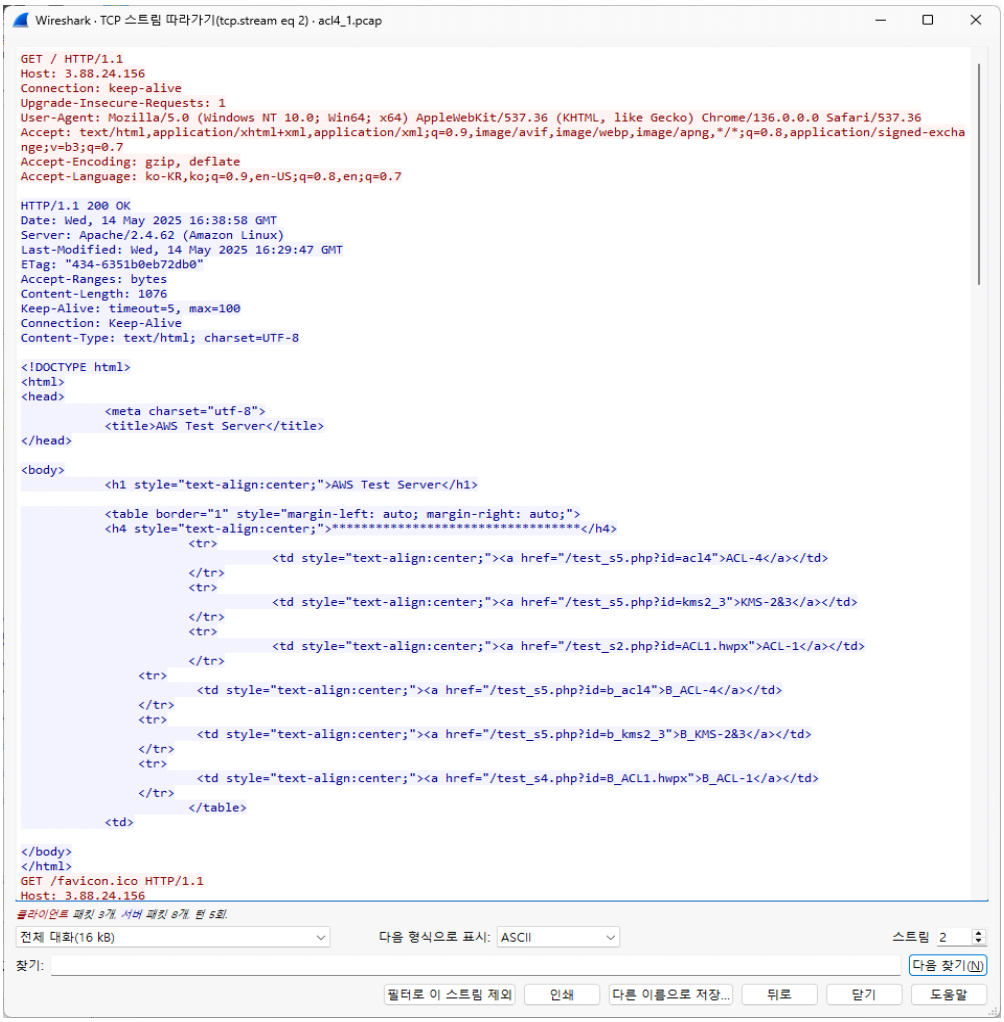

그림 9 네트워크 패킷에서 확인한 웹서버 HTML 코드

정리하자면 웹 포렌식 결과 Local에서는 주로 Cache 정보를 통해 파일명, 접근한 URL, 파일 사이즈, 시간, 응답 코드, IP 주소 등을 확인할 수 있었다. ACL1의 경우 다운로드에 성공했는데, 해당 Cache 파일이 있다면 다운로드 받았던 파일이 삭제되었더라도 열어볼 수 있다는 점에서 유출의 위험이 있다. 그리고 Web에서는 Access Log와 Error Log가 기록되어 있어 시간, 응답 내용, IP, URL 등을 확인할 수 있었다. 네트워크 포렌식에서는

Local에서 확인하는 ACL1, ACL2, ACL3, KMS1의 경우 전송 과정에서 S3 URL 정보만 확인되었고 중요 정보는 보이지 않았다. ACL4, KMS2, KMS3는 Web에서 패킷을 확인했기 때문에 S3 URL 정보와 응답 코드가 나왔고 특이 사항으로는 웹서버의 HTML 코드가 보였다. 하지만 분석을 진행한 위치가 Web, 즉 서버 측이고 Local에서 분석했을 때는 동일한 정보가 보이지 않았으며, 웹서버 HTML 코드는 공개되는 코드로 보안상 민감한 정보라고 생각되지 않기 때문에 정보 유출이라고는 보기 어렵다.

마지막으로 AWS 내의 CloudTrail를 통해 수집한 로그를 분석했다. CloudTrail로 수집한 로그는 S3 내 사전에 명시한 버킷 아래에 저장된다. S3 활동 전부에 대한 로그를 수집했으며 총 24개의 로그 이벤트가 발생했으나 확인해 본 결과 실제로 S3 파일 다운로드와 관련된 로그는 5개로 한정되었다.

그림 10은 ACL1에 접속해 다운로드에 성공한 로그이다. 로그에는 접근한 객체 이름, 버킷 이름, 이벤트 발생 시간, Source IP 주소, 요청자, 다운로드 성공 여부, AWS 리전 등이 기록되어 있다. accountID가 anonymous 즉, 익명으로 표시되어 있는데 AWS 계정으로 접근한 것이 아니고 익명으로 접근했다는 것이다. 하지만 Source IP가 기록되어 있어 S3 객체 URL이나 외부 링크를 통해 접근했다는 것으로 추정할 수 있다. aclRequired 항목이 Yes인 경우는 해당 객체가 퍼블릭 접근을 허용하고 있다는 뜻이다.


```
{
eventVersion : 1.11
userIdentity : {
  type : AWSAccount
  principalId : value
  accountId : anonymous
}
eventTime : 2025-06-08T07:19:15Z
eventSource : s3.amazonaws.com
eventName : GetObject
awsRegion : us-west-2
sourceIPAddress : 221.149.143.4
userAgent : [Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36]
requestParameters : {
  bucketName : test-active
  Host : test-active.s3.us-west-2.amazonaws.com
  key : B_ACL1.hwpx
}
responseElements : null
additionalEventData : {
  aclRequired : Yes
  CipherSuite : TLS_AES_128_GCM_SHA256
  bytesTransferredIn : 0
  x-amz-id-2 : R61J0X4WZ6ii5UmpKvrjoIZA2XRjoi9nCG5IzDD2tbdBNxIJYvEr/Z7Qvjnx97/71JGpPuMtEEQ=
  bytesTransferredOut : 104857600
}
requestID : 2S8WBFX2EB1DP7YX
eventID : 45301998-1034-30a7-b0b2-738454cb5846
readOnly : true
resources : [ 2 items
▼ 0 : {
    accountId : 654654346138
    type : AWS::S3::Bucket
    ARN : arn:aws:s3:::test-active
  }
▼ 1 : {
    type : AWS::S3::Object
    ARN : arn:aws:s3:::test-active/B_ACL1.hwpx
  }
]
```


그림 10 ACL1 JSON 로그 파일

그림 11은 ACL2에 대한 다운로드 시도 로그이다. Json 경로는 ACL1과 유사하게 확인할 수 있고, S3 객체 URL로 접속했기 때문에 anonymous로 기록되어 있다. 다른 점은 해당 시나리오가 접속 불가인 경우로 ACL1에서는 없었던 errorCode와 errorMessage를 확인할 수 있다. Access Denied가 기록되어 있어 접근이 거부된 것으로 확인할 수 있다. aclRequired 경로가 없는 것으로 보아 버킷 또는 객체에 퍼블릭 접근이 허용되지 않았음을 뜻한다.


```
{
eventVersion : 1.11
userIdentity : {
  type : AWSAccount
  principalId : value
  accountId : anonymous
}
eventTime : 2025-06-08T07:21:03Z
eventSource : s3.amazonaws.com
eventName : GetObject
awsRegion : us-west-2
sourceIPAddress : 221.149.143.4
userAgent : [Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36]
errorCode : AccessDenied
errorMessage : Access Denied
requestParameters : {
  bucketName : test-active
  Host : test-active.s3.us-west-2.amazonaws.com
  key : B_ACL2.hwpx
}
responseElements : null
additionalEventData : {
  CipherSuite : TLS_AES_128_GCM_SHA256
  bytesTransferredIn : 0
  x-amz-id-2 : BGRGnGisfmoZOiF4T2SAR7JKISlsiJeqpqo5gL9UevPX3m35gDxD7yd38gGspWLOXWrWvMrpLJbmlsWpo/5buw==
  bytesTransferredOut : 255
}
requestID : DNT5Q3HVPWT2NJGK
eventID : a353b597-8248-3086-92af-175439e23980
readOnly : true
resources : [ 2 items
0 : {
    accountId : 654654346138
    type : AWS::S3::Bucket
    ARN : arn:aws:s3:::test-active
  }
1 : {
    type : AWS::S3::Object
    ARN : arn:aws:s3:::test-active/B_ACL2.hwpx
  }
]
```


그림 11 ACL2 JSON 로그 파일

설정이 유사한 ACL3도 ACL2와 비슷한 결과를 확인할 수 있었다.

ACL4 로그에서는 특별하게 role 이름을 확인할 수 있었다. IAM 권한이 부여되어 있다는 것을 명시하고 있는 항목이다. 또한 Source IP가 AWS EC2의 IP이기 때문에 내부의 요청으로 발생한 이벤트라고 볼 수 있다. bytesTransferredOut는 AWS 리소스에서 클라이언트로 전송된 데이터의 크기를 나타내는 필드로 다운로드가 성공적으로 이뤄졌다는 것을 알 수 있다.


```
{
eventVersion : 1.11
userIdentity : {
  type : AssumedRole
  principalId : AROAZQ3DQSONNWBQQVOXU:i-08c40fa28aadcc0eb
  arn : arn:aws:sts::654654346138:assumed-role/test-s3zt-ec2tos3/i-08c40fa28aadcc0eb
  accountId : 654654346138
  accessKeyId : ASIAZQ3DQSONOZ7CIDYB
▼ sessionContext : {
  ▼ sessionIssuer : {
      type : Role
      principalId : AROAZQ3DQSONNWBQQVOXU
      arn : arn:aws:iam::654654346138:role/test-s3zt-ec2tos3
      accountId : 654654346138
      userName : test-s3zt-ec2tos3
    }
  ▼ attributes : {
      creationDate : 2025-06-08T06:59:53Z
      mfaAuthenticated : false
    }
    ec2RoleDelivery : 2.0
  }
}
eventTime : 2025-06-08T07:25:07Z
eventSource : s3.amazonaws.com
eventName : GetObject
awsRegion : us-east-1
sourceIPAddress : 54.84.39.143
userAgent : [aws-sdk-php/3.325.2 ua/2.0 OS/Linux#6.1.112-122.189.amzn2023.x86_64 lang/php#8.3.10 GuzzleHttp/7]
requestParameters : {
  bucketName : test-kms-passive
  Host : test-kms-passive.s3.amazonaws.com
  key : B_ACL4.hwpx
}
responseElements : null
additionalEventData : {
  SignatureVersion : SigV4
  CipherSuite : TLS_AES_128_GCM_SHA256
  bytesTransferredIn : 0
  AuthenticationMethod : AuthHeader
  x-amz-id-2 : eyEo8R8eLFJqEKYDDsSqG19sQxL7w6Pd5H2zuhX/5S0Gn07O0yKJ1oNfRpcwAmEo3JOanRsA3Rc=
  bytesTransferredOut : 104857600
}
```


그림 12 ACL4 JSON 로그 파일

KMS1에 관한 로그는 ACL2 로그와 마찬가지로 errorCode와 errorMessage 필드가 있는데 InvalidArgument로 기록되어 있다. 이는 파라미터 오류로 인해 접근이 거부되었다는 것이다. errorMessage에 'AWS Signature Version 4로 접근해야 한다'라는 메시지가 남아있다. aclRequired도 Yes로 퍼블릭 접근이 허용되어 있다. 이를 통해 접근 권한과 KMS 권한뿐만 아니라 인증 방식 또한 고려 대상이라는 것을 확인할 수 있었다.


```
{
eventVersion : 1.11
userIdentity :{
  type : AWSAccount
  principalId : value
  accountId : anonymous
}
eventTime : 2025-06-08T07:27:04Z
eventSource : s3.amazonaws.com
eventName : GetObject
awsRegion : us-west-2
sourceIPAddress : 221.149.143.4
userAgent : [Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36]
errorCode : InvalidArgument
errorMessage : Requests specifying Server Side Encryption with AWS KMS managed keys require AWS Signature Version 4.
requestParameters :{
  bucketName : test-active
  Host : test-active.s3.us-west-2.amazonaws.com
  key : B_KMS1.hwpx
}
responseElements : null
additionalEventData :{
  aclRequired : Yes
  CipherSuite : TLS_AES_128_GCM_SHA256
  bytesTransferredIn : 0
  x-amz-id-2 : mdPNC/gx0grm/9WEa0gFQ5NxIi1R+tJI6GZXp3sZ1TJh/697R0IwRBumy7Dv+yqF7KOPh2zHoeI=
  bytesTransferredOut : 376
}
requestID : MJYDX1FT3E2EJD0F
eventID : 84c9d081-b4d8-3669-84ac-21bd779f79b6
readOnly : true
resources : [ 2 items
▼ 0 :{
    accountId : 654654346138
    type : AWS::S3::Bucket
    ARN : arn:aws:s3:::test-active
  }
▼ 1 :{
    type : AWS::S3::Object
    ARN : arn:aws:s3:::test-active/B_KMS1.hwpx
  }
]
```


그림 13 KMS1 JSON 로그 파일

그림 14 KMS2, KMS3 관련 로그에서는 ACL4와 동일하게 role 이름과 EC2의 IP 주소를 확인할 수 있었다. 내부에서 요청이 이뤄졌고, 다운로드에 성공했음을 알 수 있다. ACL4와 다른 점은 KMS 복호화 관련 추가 로그가 존재한다는 것이다. 이벤트 시간과 IAM role, eventName의 Decrypt를 보아 KMS2, KMS3과 관련된 로그라고 추정할 수 있다. 이는 CloudTrail이 서비스를 기준으로 로그를 분리해 남기기 때문이다. 따라서 여러 서비스를 사용하는 경우 관련된 로그를 종합적으로 분석해야 정확하게 상황을 파악할 수 있다.


```
{
eventVersion : 1.11
userIdentity : {
  type : AssumedRole
  principalId : AROAZQ3DQSONNWBQQVOXU:i-08c40fa28aadcc0eb
  arn : arn:aws:sts::654654346138:assumed-role/test-s3zt-ec2tos3/i-08c40fa28aadcc0eb
  accountId : 654654346138
  accessKeyId : ASIAZQ3DQSONFYRKWNJ3
▾ sessionContext : {
  ▾ sessionIssuer : {
      type : Role
      principalId : AROAZQ3DQSONNWBQQVOXU
      arn : arn:aws:iam::654654346138:role/test-s3zt-ec2tos3
      accountId : 654654346138
      userName : test-s3zt-ec2tos3
    }
  ▾ attributes : {
      creationDate : 2025-06-08T06:59:53Z
      mfaAuthenticated : false
    }
    ec2RoleDelivery : 2.0
  }
  invokedBy : AWS Internal
}
eventTime : 2025-06-08T07:29:01Z
eventSource : kms.amazonaws.com
eventName : Decrypt
awsRegion : us-east-1
sourceIPAddress : AWS Internal
userAgent : AWS Internal
requestParameters : {
▾ encryptionContext : {
    aws:s3:arn : arn:aws:s3:::test-kms-passive
  }
  encryptionAlgorithm : SYMMETRIC_DEFAULT
}
responseElements : null
additionalEventData : {
  keyMaterialId : 7ed63629177b3b21bbb7b008625df62199178c54994eae21b6b5734a06767034
}
requestID : beee86d2-e9f6-4670-aee1-70671f0caa56
eventID : e93a40fb-e5e2-4f0d-bc16-94fc7ef35231
readOnly : ☑ true
resources : [ 1 item
▾ 0 : {
    accountId : 654654346138
    type : AWS::KMS::Key
    ARN : arn:aws:kms:us-east-1:654654346138:key/mrk-157b2964aa36499ba48abb97ca6a3801
  }
]
```


그림 14 KMS2, KMS3 JSON 로그 파일

제 V 장 결론 및 향후 계획

전 세계적으로 AWS 클라우드 서비스 사용이 증가하고 있고, 특히 스토리지 서비스인 S3는 널리 사용되고 있는 서비스 중 하나이다. 하지만 스토리지와 관련된 사고들이 지속해서 발생하고 있으며 기업 내부 기밀이나 고객들의 개인정보 등 민감한 정보가 유출되는 사례도 있다. 본 연구에서는 S3 스토리지 서비스의 보안 설정을 중심으로 분석해 보았으며, 크게 접근 권한을 부여하는 ACL 설정과 암호화 적용을 위한 KMS 서비스 설정을 비교 및 분석했다. 두 가지 설정의 활성화 여부에 따라 속도의 차이를 측정하고 디지털 포렌식 분석을 통해 보안의 강도를 비교해 보았다. 100KB/s, 200KB/s, 300KB/s, 400KB/s, 500KB/s 별로 네트워크 속도를 제한해 다운로드 및 업로드 실험을 진행했고, 예상 시간과 근사한 결과를 얻었다. 2회 정도 조금 느리게 다운로드 되는 경우가 있었으나 전체의 0.57% 비율이기 때문에 전체 결과에 영향을 주는 정도는 아니었다. 디지털 포렌식 분석에서는 Cache 정보와 네트워크 패킷을 통해 접근 URL, IP, 응답 코드, 시간 등을 공통적으로 확인할 수 있었다. ACL1의 경우 다운로드에 성공했고, Cache 파일에서 다운로드 받은 파일 내용을 확인할 수 있었다. 만약 공격자가 해당 기기에 접속할 수 있다면 민감한 정보가 외부로 유출될 가능성이 있다. 또한 오레곤 리전은 신규 리전으로 Signature Version 4 방식만 접근할 수 있어 Version 2인 경우는 접근이 불가하다. Web에서 포렌식을 진행한 ACL4, KMS2, KMS3에서는 웹서버의 HTML 코드가 패킷에서 확인되었으나, 이는 공개되는 정보로 유출로 보기에는 어렵다. 이러한 결과를 종합하자면, ACL과 KMS 설정을 모두 적용해 권한을 부여받은 사용자만 접근할 수 있도록 하고, 파일이 외부로 유출되어도 KMS Key가 없으면 복호화하지 못해 원본 내용을 확인할 수 없도록 적용하는 것이 보안 측면에서 효과적이다. 한편,

AWS 내 CloudTrail 서비스를 통해 수집한 로그에서는 Account 여부, IP 주소, 시간, 다운로드 성공 여부, 퍼블릭 접근 허용 여부, Role 이름, 복호화 내용 등 더 자세한 로그를 확인할 수 있었다. 분석 결과, 계정이 없는 익명 형태로 접근한 기록, IAM 권한이 부여된 경우 role의 이름, 다운로드에 성공했을 때 다운받은 파일의 크기, 복호화에 사용된 키 ARN과 알고리즘 등이 기록되어 있어 사용자의 흔적을 추적할 수 있다. 이렇게 Client와 Server에는 기록되지 않은 것들이 CloudTrail에서는 확인할 수 있기 때문에 보안 감사 및 디지털 포렌식 측면에서는 CloudTrail 활성화를 필수적으로 권장한다. 향후 연구에서는 AWS 외의 Azure, GCP나 국내 클라우드 서비스의 스토리지 설정을 분석해 이기종 클라우드 환경에 적용할 수 있는 대응 방안을 도출하고자 한다.

참고 문헌

[1] 정보영, “공공 금융서비스의 클라우드 전환시 보안이 미치는 영향에 관한 연구” *건국대학교 정보통신대학원*, 2024.2.

[2] 이일현, “클라우드 서비스 보안인증제도(CSAP) 개선에 관한 연구” *중앙대학교 보안대학원*, 2024.8.

[3] NIST, “The NIST Definition of Cloud Computing,” [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf. [Accessed: May 10, 2025].

[4] 최가영, “주요국 클라우드 정책 비교 연구: 산업 진흥과 규제 측면에서 정부의 역할” *고려대학교 정보보호대학원*, 2025.2.

[5] 이글루코퍼레이션, “클라우드 환경의 보안사고 사례분석을 통한 대응방안,” [Online]. Available: https://www.igloo.co.kr/security-information/클라우드-환경의-보안사고-사례분석을-통한-대응방/. [Accessed: May 10, 2025].

[6] 한국인터넷진흥원, 클라우드 서비스 보안 인증 제도 안내서, [Online]. Available: https://www.kisa.or.kr/. [Accessed: May 10, 2025].

[7] A. Kaur, G. Raj, S. Yadav, and T. Choudhury, “Performance evaluation of AWS and IBM cloud platforms for security mechanism,” in *Proc. Int. Conf. on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 2018, pp. 516 – 520. doi: 10.1109/CTEMS.2018.8769215.

[8] W. Shen, J. Qin, J. Yu, R. Hao, J. Hu, and J. Ma, “Data integrity auditing without private key storage for secure cloud storage,” *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1408 – 1421, 2021. doi: 1

0.1109/TCC.2019.2921553.

[9] B. Sengupta, A. Dixit, and S. Ruj, “Secure cloud storage with dat a dynamics using secure network coding techniques,” *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 2090 – 2101, 2022. doi: 10.1109/TC C.2020.3000342.

[10] 조광훈, “디지털 증거의 압수・수색의 문제점과 개선방안” *서울시립대학교 법학연구소,* no.21, pp.699-738, 2014.

[11] 우창인, “디지털 포렌식에서 엔드포인트 보안 솔루션 로그 활용방안” *성균관대학교 일반대학원*, 2025.2.

[12] H. Y. Chen, “A digital forensics method in cloud computing enviro nment,” *Appl. Mech. Mater.*, vol. 635 – 637, pp. 1471 – 1475, 2014. do i: 10.4028/www.scientific.net/amm.635-637.1471.

[13] S. Bhatia and J. Malhotra, “Forensic based cloud computing archit ecture – exploration and implementation,” in *Proc. Int. Conf. Comp ut. Commun. Technol. (ICCCT)*, 2019, pp. 37 – 45. doi: 10.1109/ICCC T2.2019.8824813.

[14] Z. Umar, D. U. Ebem, F. S. Bakpo, and M. Ezema, “A digital fore nsic framework design for joined heterogeneous cloud computing en vironment,” Int. *J. Comput. Sci. Netw. Secur.*, vol. 24, no. 6, pp. 20 7 – 215, 2024. doi: 10.22937/IJCSNS.2024.24.6.24.

[15] S. Thamburasa, S. Easwaramoorthy, K. Aravind, S. B. Bhushan, a nd U. Moorthy, “Digital forensic analysis of cloud storage data in I Drive and Mega Cloud Drive,” in *Proc. Int. Conf. Inventive Compu t. Technol. (ICICT)*, 2016, pp. 1 – 6. doi: 10.1109/INVENTIVE.2016.7 830159.

[16] S. Almulla, Y. Iraqi, and A. Jones, “Digital forensic of a cloud bas ed snapshot,” in *Proc. Int. Conf. Innovative Comput. Technol. (INT ECH)*, 2016, pp. 724 – 729. doi: 10.1109/INTECH.2016.7845140.

[17] 김상구, “AWS 환경에서 개인정보보호를 위한 기술적 보호조치에 관 한 연구” *건국대학교 정보통신대학원*, 2024.8.

[18] A. Gupta, A. Mehta, L. Daver, and P. Banga, “Implementation of s torage in virtual private cloud using simple storage service on AW S,” in *Proc. Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA),* 2020, pp. 213 – 217. doi: 10.1109/ICIMIA48430.2020.9074899.

[19] 조경현, “Amazon S3 제로 트러스트 모델 설계 및 포렌식 분석” *한 국정보보호학회 논문지,* vol. 33, pp. 295-303, 2023.

[20] Amazon Web Services, “Amazon Elastic Compute Cloud (EC2),” [Online]. Available: https://aws.amazon.com/ec2/. [Accessed: May 1 0, 2025].

[21] Amazon Web Services, “Amazon Simple Storage Service (S3),” [O nline]. Available: https://aws.amazon.com/s3/. [Accessed: May 10, 2 025].

[22] Locktime Software, “NetLimiter (Version 5.3.19.0),” [Online]. Availa ble: https://www.netlimiter.com/. [Accessed: May 10, 2025].

[23] DB Browser for SQLite, “DB Browser (Version 3.13.1),” [Online]. Available: https://sqlitebrowser.org/. [Accessed: May 10, 2025].

[24] Wireshark Foundation, “Wireshark (Version 4.2.4),” [Online]. Avail able: https://www.wireshark.org/. [Accessed: May 10, 2025].

[25] NirSoft, “ChromeCacheView (Version 2.51),” [Online]. Available: ht tps://www.nirsoft.net/utils/chrome_cache_view.html. [Accessed: May

10, 2025].

[26] HFR, “H734GP 무선 공유기,” [Online]. Available: https://www.hfrnet.com/. [Accessed: May 10, 2025].

[27] Samsung Electronics, “Galaxy S24 Plus,” [Online]. Available: https://www.samsung.com/sec/smartphones/galaxy-s24/. [Accessed: May 10, 2025].

[28] Lenovo, “ThinkPad X1 Carbon Gen 11,” [Online]. Available: https://www.lenovo.com/kr/ko/. [Accessed: May 10, 2025].

ABSTRACT

Security and Performance Analysis of AWS Storage Services

Young Sun Park
Department of Future Convergence Technology Engineering
Graduate School of
Sungshin Women's University

Cloud computing is one of the core foundational technologies used across enterprises, public institutions, and individuals. It allows for the flexible allocation of IT resources and a pay-as-you-go cost model, making it convenient to use. However, because resources are provisioned by cloud service providers, users face limitations in controlling all aspects of the environment. Moreover, security threats unique to cloud infrastructure—distinct from those found in traditional on-premise systems—continue to emerge. In particular, data breaches caused by misconfigured access controls in storage services are occurring frequently. Configuration errors such as enabling public access, failing to

apply encryption, or granting excessive permissions can lead to severe incidents, including information leakage. As such, it is essential to establish appropriate security configurations tailored to the cloud environment. This study investigates the security settings of Amazon S3, AWS's storage service. It analyzes the functions and effectiveness of each setting and examines the potential for data exposure through digital forensics. Based on the findings, this study proposes configuration strategies for more secure usage of Amazon S3.