정 보 보 호 학 회 지
제35권 제1호, 2025. 2

소프트웨어 공급망 공격에서의 클라우드 악용 사례 및 완화 방안 분석

김 미 연*, 최 상 훈**, 박 기 웅***

요 약

최근 클라우드 컴퓨팅의 활용도가 급증하면서 클라우드 보안 위협과 공격 표면의 확대가 주요한 우려 사항으로 대두되고 있다. 특히, 클라우드 기반 시스템의 특성상 소프트웨어 모듈 간 의존성이 증가함에 따라 소프트웨어 공급망의 취약점이 더욱 부각되고 있다. 본 논문에서 소프트웨어 공급망 공격과 연결되는 클라우드 악용 위험 및 사례를 분석한 결과, 외부 소프트웨어 패키지, 시스템 이미지 및 자원 링크의 미흡한 검증과 부족한 보안 경각심이 공급망 공격의 위협요인이 될 수 있었다. 또한, 클라우드에서 제공하는 API의 오류 및 자격 증명 메커니즘의 미비가 공급망 공격을 유발할 수 있음을 확인하였다. 분석된 사례와 같은 보안 사고를 완화하기 위한 연구에 따르면, 다중 인증 활성화, 검증된 기기 및 컨테이너 이미지 사용, 사용자의 보안 인식을 높이는 등의 노력이 클라우드 악용 사례를 효과적으로 감소시킬 수 있음을 보여준다.

Ⅰ. 서 론

클라우드 네이티브화, 마이크로서비스 아키텍처와 같은 클라우드 기반의 시스템적 특성은 여러 소프트웨어 모듈 간에 의존성을 갖도록 하며, 소프트웨어 공급망 공격에 대한 위협을 증대시킨다. Snyk 보고서에 따르면, 2031년 기준 소프트웨어 공급망 공격으로 인한 전 세계 기업의 손실액이 약 1,380억 달러에 이를 것으로 예측된다 [1]. 소프트웨어 공급망에는 여러 조직과 다양한 요소가 상호 연결되어 있다는 특성으로 인하여 구조적 복잡성이 따른다 [2]. 이로 인해 어느 한 지점에서 공격이 발생한다면, 그 피해는 공급망으로 연결된 모든 곳에 전파될 수 있다. 클라우드의 구조 또한 매우 복잡하다. 클라우드의 경우, 글로벌 데이터 센터와 사용자 간의 데이터 이동이 광범위하게 발생하며, 다양한 기기를 통해 접근할 수 있기 때문에 보안 사고로 인한 피해가 크게 확대된다. 그럼에도 불구하고 IDC는 클라우드 시장이 지속적으로 성장하여 2028년의 전 세계 매출이 1조 6,000억 달러에 이를 것으로 전망하고 있다 [3]. 이처럼 소프트웨어 공급망과 클라우드의 복잡성은 보안 사고에 따른 피해자나 피해 규모를 쉽게 특정할 수 없도록 한다. 또한, 피해가 겉으로 드러나지 않는 잠재적 피해자의 경우에는 대응조차 하기 어려운 실정이다. 따라서 본 논문에서는 소프트웨어 공급망과 클라우드에 중점을 두고 소프트웨어 공급망 공격에서 클라우드가 악용되었던 사례와 파급력을 파악한다. 또한, 관련 연구를 분석하여 이러한 위협을 완화하는 방안을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에 필요한 배경지식을 기술하고, 3장에서는 클라우드 시스템에서의 소프트웨어 공급망 공격 사례를 분석한다. 4장에서는 소프트웨어 공급망에서의 클라우드 악용을 완화하는 방안을 기술하고, 5장에서 결론 및 향후 연구에 대하여 작성한다.

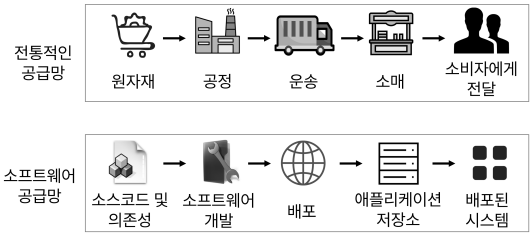
본 연구는 과학기술정보통신부의 재원으로 정보통신기획평가원(IITP)의 국방ICT융합연구(Project No. 2022-0-00701, 30%), 실감콘텐츠 핵심기술개발(Project No. RS-2023-00228996, 30%), 정보통신방송혁신인재양성사업(Project No. 2021-0-01816, 10%) 및 한국연구재단(NRF) 중견후속연구사업(Project No. RS-2023-00208460, 30%)의 지원을 받아 수행된 연구임.

* 세종대학교 시스템보안연구실 (대학원생, etcherish@naver.com)
** 세종대학교 시스템보안연구실 (연구교수, csh0052@gmail.com)
*** 세종대학교 정보보호학과 (교수, woongbak@sejong.ac.kr)

Ⅱ. 배경지식

2.1. 소프트웨어 공급망(Software Supply Chain, SSC)

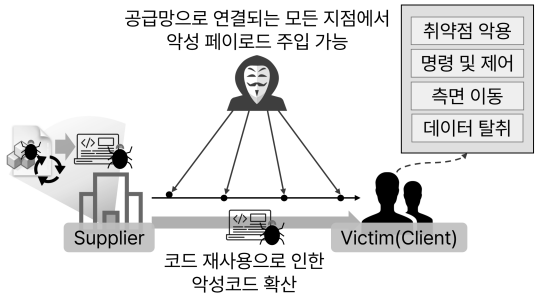
공급망이란 제품이나 서비스가 클라이언트에게 공급될 때까지의 전반적인 프로세스를 포함하고, 이와 관련된 사람, 조직, 활동, 자원 등이 서로 상호작용하며 통합된 시스템을 의미한다 [4]. 전통적인 공급망과 소프트웨어 공급망의 흐름은 [그림 1]과 같다 [5]. 먼저, 전통적인 공급망에는 원자재부터 공정, 운송, 소매를 통해 소비자에게 전달되는 모든 과정과 요소가 포함된다. 소프트웨어 공급망은 애플리케이션을 개발하고 배포하여 유지 및 관리하는 모든 과정과 소스 코드, 플랫폼, 데이터 저장소 등의 모든 구성 요소를 일컫는다. 소프트웨어 공급망은 특정한 기능을 제공하기 위해 서로 연관되거나 네트워크로 연결된 상호 의존적인 시스템의 한 예시이며, 독립적인 여러 조직은 소프트웨어 공급망을 통해 여러 기술적 역량을 제공할 수 있다 [6].



〔그림 1〕 전통적인 공급망과 소프트웨어 공급망의 흐름

2.2. 소프트웨어 공급망 공격

소프트웨어 공급망을 통해 개발 및 배포되는 제품의 복잡성은 취약점이나 악성코드 삽입의 가능성을 증가시킨다. 소프트웨어 공급망 공격이 발생 경로는 [그림 2]와 같다. 공격자들은 애플리케이션이 조직에 배포되기 전 단계에서 소프트웨어 공급망을 악용하여 악성 페이로드를 전달한다. G. Betul et al.에 따르면 [7], 악성 페이로드가 조직의 시스템에서 활성화된 이후, 다음의 단계를 거친다: *(1) 취약점 악용 (2) 명령 및 제어 (3) 측면 이동 (4) 데이터 탈취* 이러한 일련의 단계를 통해 공격자는 조직 내에서 치명적인 보안 사고를 유발할 수 있게 된다.



〔그림 2〕 소프트웨어 공급망 공격 발생 경로

또한, 소프트웨어 공급망에서의 취약점은 코드 재사용으로 인해 악화될 수 있다[8]. 코드를 재사용하는 것 자체가 문제가 되는 것은 아니지만, 소스 검증을 통하여 최종 제품에는 신뢰성 문제가 발생하지 않도록 해야 한다. J. Vojnović에 따르면[9], 오픈소스 라이브러리로 인한 코드 재사용은 개발자의 작업량을 줄여주기 때문에 소프트웨어 개발에서 널리 사용되고 있다. 그러나 이러한 의존성 문제는 불완전함, 취약점, 악성코드 등을 전파할 수 있어 보안적 위험이 야기될 수 있다.

Ⅲ. 클라우드에서의 소프트웨어 공급망 공격 사례 분석

본 장에서는 클라우드가 소프트웨어 공급망 공격에 어떻게 악용되는지 살펴보고, 공급망 공격에 악용된 클라우드의 파급력에 대하여 분석한 내용을 기술한다.

3.1. 컨테이너의 악용

3.1.1. Docker hub의 악용

2021년 8월, 클라우드 네이티브 및 컨테이너 보안 기업 아쿠아시큐리티의 Team Nautilus는 악성 컨테이너 이미지를 통해 발생하는 컨테이너 공급망 공격 사례를 발표했다 [10]. 모두 Docker hub에서 발견되었으며, 발견된 악성 이미지의 개수는 총 5개였다. Docker hub는 컨테이너 이미지를 생성 및 저장하고 다운로드할 수 있는 세계 최대 규모의 클라우드 기반 레지스트리이다. 본 연구팀이 발견한 5개의 악성 컨테이너 이미지의 목록은 [표 1]과 같다. 이미지 ①은 10만회 이상으로 가장 많이 다운로드 되었다. 나열된

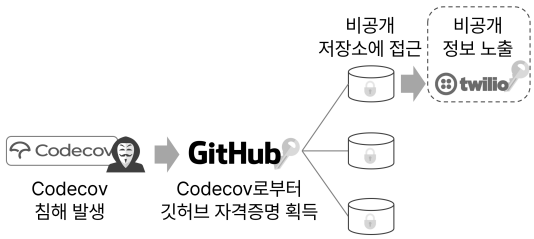
〔표 1〕 Docker hub에서 발견된 악성 이미지의 목록

컨테이너 이미지	악성 요소		다운로드
① thanhtudo/thanhtudo:latest	crypto miner	dap.py script	10만+
② thieunutre/thieunutre:latest	crypto miner	dap.py script	11
③ chanquaa/chanquaa:latest	crypto miner	dap.py script	18
④ c43602f9cc95/openjdk:0da242bcd93b7f	-	Execute xmring	1만+
⑤ 700888880a0/golang:e2e26c727b88	-	Execute xmring	1만+

악성 이미지 중 상위 3개(①, ②, ③)의 이미지는 dao.py 스크립트를 통해 xmrig 바이너리를 실행시킨다. 나머지 2개의 이미지(④, ⑤)는 Docker hub에서 공유되고 있는 공식 이미지와 유사한 이름을 사용하여 무해한 컨테이너처럼 보이도록 하지만, 실행 이후에 xmrig 바이너리가 실행된다. 5개의 이미지 모두 리소스 하이재킹을 통해 암호화폐를 채굴하는 데 사용되며, 특히 상위 2개의 이미지(①, ②)는 또 다른 공급망 공격에 사용될 가능성이 높은 것으로 알려졌다. 이처럼 공격자는 Docker hub를 직접적으로 악용하여 공급망 공격의 발생을 초래한다.

3.1.2. Codecov 이미지 오류로 인한 데이터 유출

2021년 1월, Codecov의 도커 이미지 취약점을 악용한 공격자가 민감한 데이터에 무단으로 접근했다 [11]. 해당 공격은 약 두 달 가량 탐지되지 않다가 같은 해 2021년 4월, 고객에 의해서 발견되었다. 공격자는 Codecov 이미지의 오류를 악용하여 자격 증명을 추출한 뒤, Bash uploder 스크립트에 악성코드 한 줄을 추가했다. 이 악성코드는 CI(Continuous Integration) 환경의 모든 환경 변수를 공격자가 소유한 원격 서버로 전송한다. 공격자에게 전송된 환경 변수에는 애플리케이션을 실행하기 위해 필요한 민감 정보가 포함되어 있었다. Codecov는 약 2만 3천여 명의 고객을 보유하고 있기 때문에 피해 조직이나 규모를 명확하게 특정하기는 어려웠지만, 사건이 공개된 지 7일 후에 깃허브(GitHub)에서 해당 사건과 관련된 의심스러운 활동이 감지되었다. 깃허브에는 Codecov의 고객사이자 미국의 클라우드 통신 회사인 Twilio의 공개되지 않은 저장소가 복제되어 있었고, 저장소 내부에는 Twilio 사용자 토큰의 일부가 노출되어 있었다. 공격 경로는 [그림 3]과 같으며, Twiilo는 해당 공격과 관련하여 자체적인 성명을 발표했다 [12].



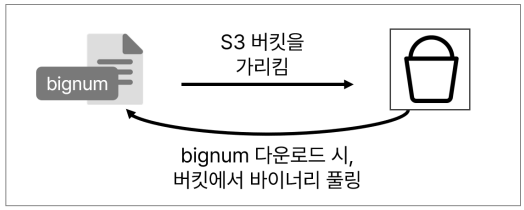
〔그림 3〕 Codecov 침해로 인한 Twillo 공격 경로

또한, 고객사 중 하나인 HashiCorp도 공식 성명을 통해 피해 사실을 밝혔다 [13]. HashiCorp에 따르면, HashiCorp CI 파이프라인의 일부로 문제가 발생한 Codecov의 구성 요소를 사용하고 있었는데 이로 인해 제품 검증에 사용되는 GPG 개인 키가 노출되었다. Codecov 침해 사건은 잠재적 피해자의 규모를 특정할 수 없고, 피해의 정도조차 예측할 수 없는 공급망 공격의 파급력을 명확하게 보여주는 사례이다.

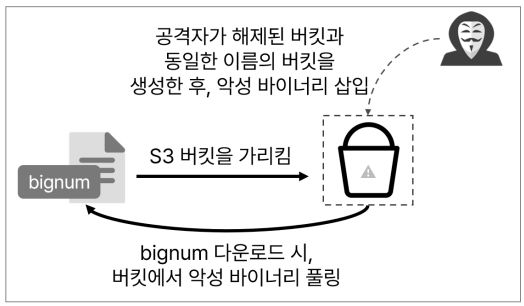
3.2. 클라우드 스토리지의 악용

3.2.1. AWS S3 버킷 하이재킹

2023년, NPM 패키지 중 하나인 bignum에서 악성코드가 발견되었다는 보고가 Github에 게시되었다다 [14]. bignum v.0.12.2~0.13.0은 AWS 스토리지인 S3에서 호스팅되던 바이너리에 의존하였는데 이 바이너리는 [그림 4]의 ①과 같이 패키지 설치 시에 버킷에서 풀링되어 패키지의 기능을 지원했다. AWS의 S3 버킷은 전 세계적으로 고유한 이름을 가져야 하므로 동일한 이름의 버킷을 생성할 수 없지만, 버킷이 해제되면 해당 버킷의 이름을 사용할 수 있게 된다. 사용자는 버킷을 해제할 때 버킷을 가리키고 있던 포인터도 제거해야 하는데 버킷만 해제하고 포인터는 제거하지 않아 bignum 패키지는 여전히 해제된 버킷을 참조하고 있던 상태였다. 한때 활성화 상태였던 버킷이 해제되었다는 사실을 인지한 공격자는 bignum 패키지가 가리키고 있는 버킷과 동일한 이름의 버킷을 생성했다. 공격자는 장악한 버킷에 악성 바이너리를 삽입하여 [그림 4]의 ②와 같이 사용자가 bignum 패키지를 설치할 때 악성 바이너리가 풀링되도록 했다.



① 정상 바이너리가 존재하는 버킷



② 공격자가 주입한 악성 바이너리가 존재하는 버킷

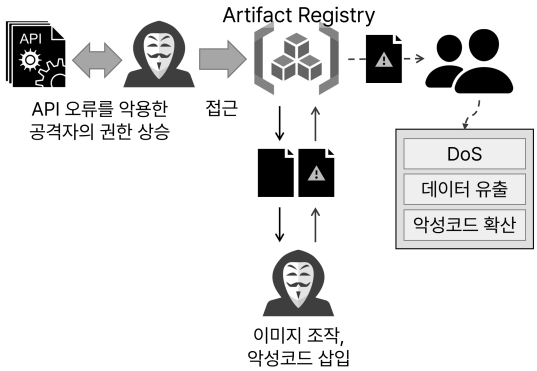
〔그림 4〕 bignum의 바이너리 풀링과 버킷 하이재킹의 흐름

악성 바이너리는 정상적으로 실행되는 패키지처럼 보였지만, 사용자가 감지하기 어려운 방식의 악성 페이로드를 추가했다. 이 악성 페이로드는 사용자의 자격 증명을 탈취하여 패키지가 풀링되었던 버킷으로 전달되도록 설계되었다. 공격자는 단 한 줄의 코드 변경 없이 사용자의 자격 증명을 획득할 수 있었다.

3.3. 자격 증명의 악용

3.3.1. 자격 증명의 취약점을 악용한 레지스트리 무단 접근

2023년, 클라우드 보안 회사인 Orca Security는 Google Cloud Build 서비스에서 설계적 결함(Bad.Build)을 발견했다고 발표했다 [15]. 발견된 결함은 권한 상승을 통해 Google의 Artifact Registry에 무단 접근할 수 있는 것으로 알려졌다. GCP(Google Cloud Platform)에서 setIamPolicy 메서드는 IAM 정책을 업데이트할 때 사용되고, 권한이 업데이트되면 setIamPolicy API가 호출된다. 통상적으로 API에는 편집된 권한에 대한 내용만 포함되어야 하는데 해당 결함은 전체 프로젝트에 대한 권한이 포함된다는 것이 문제였다. 전체 권한이 노출되면 공격자 입장에서

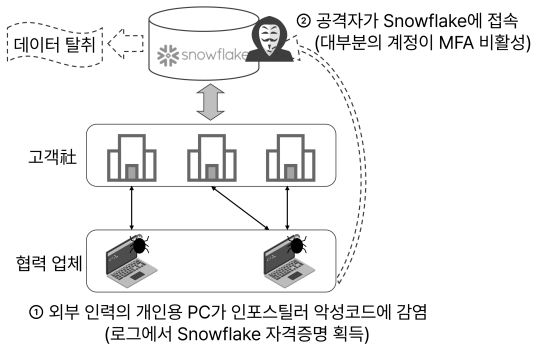


〔그림 5〕 Bad.Build를 악용한 공급망 공격 시나리오

는 측면 이동과 권한 상승 공격을 비교적 쉽게 수행할 수 있다. 공격자가 권한 상승 공격에 성공하면 Google의 Artifact Registry에 접근하여 보관된 이미지를 조작할 수 있고, 악성코드를 삽입할 수 있게 된다. 이러한 이미지를 기반으로 빌드된 애플리케이션은 DoS, 데이터 유출, 악성코드 확산 등의 결과를 초래할 수 있고, 변조된 애플리케이션이 여러 사용자의 환경으로 배포되는 경우, [그림 5]와 같이 공급망 공격으로 확대될 수 있다. Orca Security는 구글이 취약점에 대한 보고를 받고 패치를 통해 일부 권한을 제한하기는 했지만, 취약점 자체가 제거된 것은 아니기 때문에 공급망 공격에 대한 위험성은 여전히 존재한다고 언급했다.

3.3.2. Snowflake 데이터 유출

2024년, 사이버 보안 회사인 맨디언트가 데이터베이스 레코드와 관련한 위협 정보를 수집했는데, 이는 클라우드 기반 스토리지 회사인 Snowflake에서 유출된 것이었다[16]. Snowflake 데이터 유출 사고의 경로는 [그림 6]과 같다. 맨디언트가 조사한 결과에 따르면, 공격의 시작은 고객이 고용한 외부 인력의 개인용 PC였고, 해당 PC에는 민감 정보를 훔치는 인포스틸러 악성코드(Infostealer Malware)가 설치되어 있었다. 게임이나 불법적인 활동에 사용되었거나 모니터링이 제대로 되지 않는 개인용 PC는 초기 침입 경로를 확대시킬 수 있기 때문에 각별한 주의를 기울여 사용해야 한다. 인포스틸러 악성코드를 통해 유출된 Snowflake 자격 증명은 공격자가 Snowflake에 접속하여 데이터를 탈취하는 데 사용되었다. 공격자는 탈



〔그림 6〕 Snowflake 데이터 유출 사고의 경로

취한 데이터를 사이버 범죄 포럼에 판매하거나 피해 기업의 금전을 탈취하기 위해 협박하는 용도로 사용했다. 맨디언트는 해당 공격이 성공할 수 있었던 이유를 *(1) 공격의 대상은 다중 인증이 비활성화된 계정 (2) 오래된 자격 증명을 사용 (3) 네트워크 위치에 따른 접근 허용 목록 미설정*과 같이 크게 세 가지로 분류했으며, 해당 사건은 Snowflake의 문제가 아닌 고객사들의 개인정보 유출로 인한 것이라고 밝혔다. TechCrunch가 제공한 정보에 따르면, 해당 사고로 유출된 미국의 이동통신사 AT&T 데이터에는 고객의 전화번호와 6개월간의 통화/메시지 기록이 포함되어 있었고, 피해자의 수는 약 1억 1천만 명에 달한다 [17].

3.4. 분석 결과

본 논문에서는 소프트웨어 공급망 공격과 연결되는 다섯 가지의 클라우드 악용 위험 및 사례를 분석했다. 분석 결과에 따르면, 본 연구에서 분석한 사례 3.1.1, 3.1.2, 3.2.1은 외부에서 도입되는 소프트웨어 패키지, 시스템 이미지, 외부 자원 링크에 대한 검증이 미흡하거나 보안에 대한 경각심이 부족한 경우가 소프트웨어 공급망 공격으로 이어질 수 있음을 보여준다. 또한, 사례 3.3.1과 3.3.2는 클라우드 환경에서 제공되는 API의 오류, 설계적 결함, 자격 증명 메커니즘의 미비 등으로 인해 소프트웨어 공급망 공격이 언제든지 발생할 수 있음을 시사한다.

소프트웨어 공급망과 클라우드의 복잡한 특성상, 다양한 요인으로 인해 클라우드가 소프트웨어 공급망 공격의 일환으로 악용된다면, 피해 조직이나 피해의 정도를 정확하게 특정할 수 없기 때문에 대응 방안이나 적절한 조치를 수행하기 어렵다는 한계가 있었다. 또한, 피해 사례가 직접적으로 드러나지 않았지만, 공격의 영향을 받은 잠재적 피해자나 조직의 수도 상당할 것으로 예측된다.

Ⅳ. 소프트웨어 공급망에서의 클라우드 악용 완화

본 장에서는 3장에서 언급되었던 공급망 공격에서의 클라우드 악용을 완화하기 위해 여러 연구를 분석하고 분석 결과를 바탕으로 한 완화 방안을 기술한다.

4.1. 컨테이너 악용의 완화

Docker 이미지에 악성코드를 주입하여 Docker hub를 통해 배포하는 공격자들이 점점 증가하고 있으며 [18], 사용자의 부주의로 인해 민감 정보가 포함된 이미지를 생성되어 공유되기도 한다 [19]. 이러한 유형의 보안 사고는 이미지 제작자와 사용자 모두를 위협한다. D. Markus et al.의 연구에 따르면[19], Docker 이미지의 구조는 Github처럼 쉽게 확인할 수 없으므로 유출된 정보를 직관적으로 파악하는 것은 매우 어렵다. 이러한 문제를 완화하기 위해 이미지 생성자가 이미지를 공유할 때 비밀 정보가 포함되어 있을 수 있다는 알림을 통해 이미지 공유에 신중할 수 있도록 하고, 사용자에게는 이미지에 민감 정보가 포함되어 공격이 대상이 될 수 있다는 경고를 제공하여 이미지 사용에 대한 경각심을 갖도록 한다. 또한, 비밀 정보를 찾아내는 데 도움을 주는 도구인 TruffleHog와 SecretScanner를 사용하거나 Docker hub 자체적으로 이미지에 포함된 파일을 시각화하여 보여주는 인터페이스를 제공하는 것도 도움이 될 수 있다고 언급했다.

H. M. Sadun et al.은[20] 기존의 컨테이너 방어를 위한 스캔 도구와 이상 탐지 알고리즘을 사용하여 정적/동적 분석에 대한 평가를 진행하고, 최신화된 공격을 위한 동적 이상 탐지 모델을 평가하기 위해 새로운 데이터셋을 제안했다. 해당 연구팀의 분석에 따르면 정적 스캔 도구를 사용하여 이상을 탐지하는 것은 비교적 간단하지만, 50% 이상의 탐지율을 가진 도구는 없었으므로 스캔 도구에만 의존하여 컨테이너를 보호하는 것에는 한계가 있었다. 또한, 이상 탐지 알고리즘을 활용한 동적 분석의 경우에도 공격 패턴이 복잡한 경우가 많아서 탐지율이 낮은 편이었다.

Supervised 알고리즘 중에서는 Random Forest(RF)와 Decision Tree(DT)가 일부 데이터에서 80%의 탐지율을 기록했는데 새로 생성된 데이터셋에서는 20% 가량 감소하는 것으로 나타났다. 본 연구팀은 연구 결과를 바탕으로 정적/동적 솔루션을 결합하면 다양한 공격을 완화하는 데 도움이 되지만, 여전히 컨테이너를 보호하기 위해서는 추가적인 연구가 시급하다고 강조했다.

4.2. 클라우드 스토리지 악용의 완화

도메인이나 패키지가 이미 제거된 스토리지를 참조하지 않도록 각별한 주의가 필요하다. 특히, 공격자들은 클라우드 자산과 연결된 도메인이나 패키지 등의 좋은 평판을 이용하는 경우가 많기 때문에 보안 사고가 발생하면 조직의 이미지나 평판에도 악영향을 미칠 수 있다 [21]. 이러한 문제를 완화하기 위해 가장 근본적으로 진행해야 할 방안은 버킷의 소유자가 버킷을 해제할 때 자신의 애플리케이션이 더 이상 버킷을 가리키지 않도록 참조를 해제하고, 접근 제어 정책을 갱신해야 한다 [22]. J.Frieß et al.은[21] 공격자가 해제된 버킷의 이름을 예측하여 악용하지 못하도록 무작위 식별자(randomized identifiers)의 사용을 제안했다. 버킷의 이름을 사용자가 지정하는 경우 직관적이고 예측하기 쉬운 이름으로 설정하는 경우가 많기 때문에 공격자가 하이재킹하기에 유리하다. 그러나 IP pool에서 IP를 할당받는 것과 같이 네임서버에서 리소스의 이름을 무작위로 할당받는다면 공격자는 리소스의 이름을 예측하는 것이 거의 불가능하거나 어려워지기 때문에 리소스를 악용한 하이재킹이 감소할 수 있다.

4.3. 자격 증명 악용의 완화

클라우드 보안을 강화하기 위해 다중 인증(Multi-Factor Authentication, MFA) 설정은 필수적이다 [23]. 다중 인증은 아이디와 패스워드로 로그인하는 것 이상의 추가적인 인증을 제공하는 보안 메커니즘으로 가지고 있는 것(OTP, 스마트 카드 등), 사용자가 가진 고유한 특성(지문, 홍채 등) 등의 요소가 포함된다 [24]. 최근 CSP는 자체적으로 다중 인증 솔루션을 제공하고 있기 때문에 사용자는 별도의 인프라 관리 없이도 클라우드 시스템에 대한 다중 인증을 적용할 수 있다 [24]. 다중 인증은 접근 권한이 있는 사용자나 조직만 인증하므로 특정 클라우드 시스템이나 사용자 계정에 대해 보다 향상된 수준의 보안을 제공하여 사이버 공격의 가능성을 크게 낮출 수 있다 [23, 25]. 또한, 역할을 기반으로 접근을 제어하고 최소 권한 원칙(Principle of Least Privilege, PoLP)을 적용하면, 사용자가 담당 업무에 필요한 데이터와 리소스에만 접근할 수 있기 때문에 공격의 표면이 줄어든다 [26]. M. M. Alani에 따르면[27], 자격 증명을 완화하기 위해 조직원의 인식 제고를 통하여 사회공학 공격을 방지하고 모든 시스템이나 패치 프로그램을 최신 상태로 유지하는 것이 중요하다. 또한, 사용자와 서비스 사이에서 자격 증명이 재사용되는 것을 금지해야 한다. 그러나 이러한 설정이 클라우드 보안 사고에 대한 위험을 감소시키는 것은 맞지만, 완벽한 보안을 제공할 수는 없다고 강조했다.

Ⅴ. 결론 및 향후 연구

본 논문에서는 클라우드 서비스(컨테이너, 스토리지, 자격 증명 등)가 소프트웨어 공급망 공격에 악용된 실제 사례에 대하여 파악하고, 그로 인한 파급력을 분석했다. 분석 결과에 따르면, 외부에서 도입되는 소프트웨어 패키지, 시스템 이미지, 외부 자원 링크에 대한 검증이나 보안에 대한 경각심 부족이 소프트웨어 공급망 공격으로 이어질 수 있음을 발견했다. 또한, 클라우드 환경에서 제공되는 API의 오류나 자격 증명 메커니즘의 설정 미흡은 공급망 공격이 언제든지 발생할 수 있음을 시사한다. 이러한 소프트웨어 공급망에서의 클라우드 악용을 완화하기 위해 우리는 다양한 연구 내용을 기반으로 완화 방안을 제안하였다. 소프트웨어 공급망 공격에서의 클라우드 악용을 완화하기 위해서는 클라우드 사용자가 안전한 기기만을 이용하여 시스템에 접근해야 한다. 또한, 역할을 기반으로 최소한의 권한만 부여하고, 검증된 소프트웨어나 이미지를 사용하는 등의 노력을 기울여야 한다.

향후 연구에서는 MITRE Att&ck^{TM} Matrix와 소프트웨어 공급망 공격의 매핑을 통해 공격 벡터와 완화 방안을 체계화할 수 있는 연구를 진행할 것이다.

참 고 문 헌

[1] Synk, “2023 Software Supply Chain Attack Report”

[2] C. Alberts, J. Haller, C. Wallen and C. Woody, “Assessing DoD system acquisition supply chain risk management,” CrossTalk, vol. 30, no. 3, pp-4-8, 2017

[3] IDC, “Worldwide Public Cloud Services Revenues Grew 19.9% Year Over Year in 2023”, Oct. 2024

[4] X. Wang, “On the feasibility of detecting software supply chain attacks,” MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM), pp. 458-463, Nov. 2021

[5] M. J. H. Faruk, M. Tasnim, H. Shahriar, M. Valero, A. Rahman and F. wu, “Investigating Novel Approaches to Defend Software Supply Chain Attacks,” 33rd IEEE Int. Symp. Softw. Reliab. Eng, pp. 283-288, 31 Oct.-3 Nov. 2022

[6] C. J. Alberts, A. J. Dorofee, R. Creel, R. J. Ellison and C. Woody, “A systemic approach for assessing software supply-chain risk,” 2011 44th Hawaii International Conference on System Sciences, pp. 1-8, Jan. 2011

[7] B. Gokkaya, L. Aniello and B. Halak, “Software supply chain: review of attacks, risk assessment strategies and security controls,” 2023

[8] H. Hauser, “Hardening the Software Supply Chain: Developing a System to Prevent Dependency Confusion Attacks in Cloud Based Continuous Integration and Deployment Processes”, Master’s Thesis, Hochschule Wismar University of Applied Sciences Technology, Sep. 2022

[9] J. Vojnović, “Mitigating Supply Chain Attacks through Detection of High-Risk Software Dependencies,” Radboud Universiteit, Mar. 2023

[10] Aqua Security, “Threat Alert: Supply Chain Attacks Using Container Images”, Aug. 2021

[11] J. Mackenzie, “Codecov supply chain breach - explained step by step”, GitGuardian BLOG, Jun. 2021(https://blog.gitguardian.com/codecov-supply-chain-breach/)

[12] Twilio, “Twilio’s Response to the Recent Codecov Vulnerability”, May. 2021

[13] HashiCorp, “HCSEC-2021-12 - Codecov Security Event and HashiCorp GPG Key Exposure”, Apr. 2021

[14] N. Guy, “Hijacking S3 Buckets: New Attack Technique Exploited in the Wild by Supply Chain Attackers”, Checkmarx, Jun. 2023(https://checkmarx.com/blog/hijacking-s3-buckets-new-attack-technique-exploited-in-the-wild-by-supply-chain-attackers/).

[15] Orca Security, “Bad.Build: A Critical Privilege Escalation Design Flaw in Google Cloud Build Enables a Supply Chain Attack”, Jul. 2023

[16] Mandiant, “UNC5537 Targets Snowflake Customer Instances for Data Theft and Extortion”, Google Cloud, Jun. 2024

[17] W. Zack, “AT&T says criminals stole phone records of ‘nearly all’ customers in new data breach”, TechCrunch, Jul. 2024(https://techcrunch.com/2024/07/12/att-phone-records-stolen-data-breach/)

[18] M. Aldiabat, Q. M. Yaseen and Q. A. Ein, “An Efficient Random Forest Classifier for Detecting Malicious Docker Images in Docker Hub Repository,” IEEE Access, vol. 12, 2024

[19] M. Dahlmanns, C. Sander, R. Decker and K. Wehrle, “Secrets revealed in container images: an internet-wide study on occurrence and impact,” Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security, pp. 797- 811, Jul. 2023.

[20] M. S. Haq, T. D. Nguyen, A. Ş. Tosun, F. Vollmer, T. Korkmaz and A. R. Sadeghi, “SoK: A comprehensive analysis and evaluation of docker container attack and defense mechanisms,” 2024 IEEE Symposium on Security and Privacy (SP), pp. 465-183, May. 2024

[21] J. Frieß, T. Gattermayer, N. Gelernter, H. Schulmann and M Waidner, “Cloudy with a Chance of Cyberattacks: Dangling Resources

Abuse on Cloud Platforms," 21st USENIX Symposium on Networked Systems Design and Implementation, pp. 1977-1994, Apr. 2024

[22] A. Continella, M. Polino, M. Pogliani and S. Zanero, "There's a hole in that bucket! a large-scale analysis of misconfigured s3 buckets," Proceedings of the 34th Annual Computer Security Applications Conference, pp. 702-711, Dec. 2018

〈저 자 소 개〉

김 미 연 (Mi-Yeon Kim)

2024년 8월 : 부산외국어대학교 스마트융합보안학과 학사
2024년 9월~현재 : 세종대학교 정보보호학과 석사과정
<관심분야> 정보보호, 클라우드

최 상 훈 (Sang-Hoon Choi)

정회원

2014년 : 대전대학교 정보보안학과 학사
2016년 : 대전대학교 정보보안학과 석사
2023년 : 세종대학교 정보보호학과 박사
2024년 : 세종대학교 시스템보안연구실 연구교수
<관심분야> 하이퍼바이저, 시스템 모니터링, 시스템 메모리, 악성코드 분석, 딥러닝

박 기 웅 (Ki-Woong Park)

종신회원

연세대학교 Computer Science 학사
KAIST Electrical Engineering 석사
KAIST Electrical Engineering 박사
2009년 10월 : Microsoft Research, Graduate Research Fellow
2012년 8월 : 국가보안기술연구소 연구원
2016년 8월 : 대전대학교 정보보안학과 교수
2016년 9월~현재 : 세종대학교 정보보호학과 교수
<관심분야> 클라우드 시스템 보안, 초고속 보안 시스템, 시스템 인스펙션, 디지털포렌식