DevSecOps 를 위한 AWS CloudFormation 기반 코드형 인프라 취약성 스캐닝 효율성 분석

채시윤 [1], 홍지원 [1], 김정아 [1], 박승현 [1], 김성민 [2]
[1] 성신여자대학교 융합보안공학과 학부생
[2] 성신여자대학교 융합보안공학과 교수

{20200140, 20211107, 20200913, 20211058, sm.kim}@sungshin.ac.kr

Comparative analysis of IaC Vulnerability Scanning Efficiency with AWS Cloudformation for DevSecOps

Siyun Chae[1], Jiwon Hong[1], Junga Kim[1], Seunghyun Park[1], Seongmin Kim[2]
[1]Dept. of Convergence Security Engineering, Sungshin Women's University
[2]Dept. of Convergence Security Engineering, Sungshin Women's University

요 약

최근 클라우드 컴퓨팅 인프라 및 소프트웨어의 지속적인 발전으로 인한 복잡성 증가로 인해 신속한 확장성과 유연성에 대한 요구가 증가하고 있다. 이에 클라우드 네이티브 환경과의 호환성뿐만 아니라 개발과 운영의 효율성을 높이고자 코드로 인프라를 정의하여 자동화된 환경을 구축해 주는 코드형 인프라(Infrastructure as Code, IaC) 기술이 주목받고 있으며, AWS CloudFormation 은 대표적인 솔루션 중 하나이다. 그러나 IaC 형태로 배포되는 템플릿에 취약성이 존재할 경우, 인스턴스가 실행되기 전까지 보안 취약점을 미리 발견하기 어려워 DevOps 사이클에서의 보안 이슈를 야기할 수 있다. 이에 본 논문은 CloudFormation 템플릿의 보안 취약성 스캔이 가능하다고 알려진 오픈 소스 도구의 효율성을 평가하기 위한 사례 연구를 수행한다. 분석 결과를 바탕으로, DevSecOps 달성을 위한 IaC 기반 환경에서 취약성 사전 탐지의 필요성과 세분화된 접근 방식을 제시하고자 한다.

1. 서론

최근 기업들이 서버 운영 방식을 전통적인 온프레미스(on-premise) 환경에서 클라우드 기반으로 전환하는 움직임이 가속화됨에 따라, 이를 신속하고 일관되게 개발, 배포 및 관리할수 있게 자동화하는 DevOps 기술이 주목받고 있다. 이중, 코드형 인프라(Infrastructure as Code, IaC)[1] 기술은 클라우드 인프라를 코드 형태로 배포 및 관리할 수 있게하는 DevOps 구축을 위한 핵심기술 중 하나이다. 대표적인 클라우드 플랫폼인 AWS(Amazon Web Services)의 경우, DevOps 구축을 위해 효율적인 개발 및 인프라 관리가 가능한 IaC 도구인 CloudFormation[2]을 제공하고 있다.

그와 동시에, DevOps 에 보안 중심의 접근 방식을 통합하여 보안성 강화에 목적을 둔 새로운 방법론인 DevSecOps[3] 달성을 위한 움직임이 최근 AWS 를 포함한 메이저 클라우드 서비스 제공자를 중심으로 나타나고 있다. 그러나, 개발, 배포 및 운영의 자동화 과정에서의 전주기적 보안성 구축을 지향하는 DevSecOps 를 위한 클라우드 환경에서의 취약성 탐지 기술은 개선이 필요하다. 구체적으로, CloudWatch 와 같은 AWS 에서 제공하는 보안 서비스들은 인스턴스 배포 이후에 대한 보안 탐지만을 수행하여 CloudFormation 템플릿의 구문 오류나 보안 취약성을 사전에 탐지하는데 한계가 있다. 이에 본 논문은 대표적 코드형 인프라 중 하나인 AWS CloudFormation 템플릿을 대상으로, 오픈소스 정적 분석 도구별 성능 및 실효성을 비교 평가하고자 한다. 이를 통해, 코드형 인프라 취약성 탐지 분류체계를 마련하고, 세분화된 접근 방식을 제시하고자 한다.

2. CloudFormation 취약성 정적 분석 도구

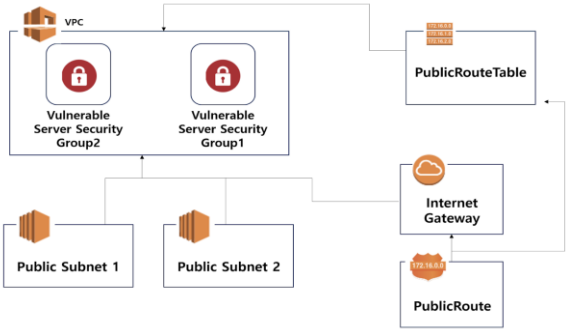
본 연구는 CloudFormation 템플릿의 취약점을 사전에 탐지하여 IaC 보안을 강화하는 데 초점을 두고 있다. 이를 위해 CloudFormation 템플릿을 분석하고 오류를 진단하는 Snyk[4], CFRipper[5], cfn-nag[6]를 활용하고자 한다. 이때, 오픈 소스로 제공되는 정적 분석 도구 중 높은 사용 빈도수를 기준으로 도구들을 선별하였다.

먼저 Snyk 는 보안 키 관리 및 암호화 설정과 같은 추가적인 취약점을 탐지할 수 있으며, 각 취약점에 대한 심각성 수준과 개수를 제공하여 취약점의 중요성을 명시한다. CFRipper 는 리소스 간의 종속성과 네이밍 규칙을 중점으로 CloudFormation 템플릿의 구조적인 문제를 식별하여 코드 보안 강화에 도움을 준다. 마지막으로, cfn-nag 는 IAM 정책 및 보안 그룹 설정 외에도 네트워크 구성 및 리소스 태깅과 같은 다양한 측면에서 취약점을 식별할 수 있다.

Snyk, CFRipper, 그리고 cfn-nag 세 도구 모두 스캐닝대상이 되는 AWS CloudFormation 템플릿 중 대부분을 차지하는 YAML 및 JSON 형식에 대한 분석을 지원하며, 이는 인스턴스 배포 전 정적 분석의 형태로 수행된다. 실효성 평가를 위해 본 연구에서는 동일한 YAML 파일을 대상으로 세 도구의 분석 결과와 산출된 항목을 비교하였다.

결과 비교를 위한 평가 기준은 AWS 보안 핵심 요소들로 선정하였다. 구체적으로, 인증과 권한 부여를 관리하는 IAM(Identity and Access Management) 정책, 리소스에 대한 네트워크 트래픽을 제어하여 보안을 강화하는 보안 그룹 설정, 데이터 보호 강화를 담당하는 리소스 암호화 여부 등을 분석 결과의 평가 기준으로 삼았다. 비교 분석 결과, 동일한 대상임에도 제공 기능과 탐지 범위가 상이하여 도구별 고유의 형식으로 보안 취약점을 식별한다. 예를 들어, 취약점에 대한 구체적인 해결 방법은 Snyk 만 제공하는 반면, CFRipper 와 cfn-nag 는 취약점 탐지와 식별에 중점을 두는 등 취약점 탐지 범위에서도 차이를 보인다.

3. IaC 스캐닝 성능 평가: 취약한 VPC 구성



(그림 1) Project-Vpc.yaml

앞서 언급한 평가기준과 관련하여 IaC 기반의 VPC(Virtual Private Cloud) 구축 과정에서 취약성이 존재할 경우, 각 도구의 탐지 결과를 비교 평가하였다. 그림 1 은 VPC 를 구성하는 동시에 보안 취약성을 가진 YAML 파일의 인프라 구성도를 도식화한 것이다. 두 개의 퍼블릭 서브넷을 설정하고 각 서브넷에 보안 그룹 구성을 나타내지만, 보안 그룹 설정에서 모든 IP 주소(0.0.0.0/0)로부터의 트래픽을 특정 포트에 대해 허용하고 있다. 이는 외부에서 EC2(Elastic Compute Cloud) 인스턴스에 접근할 수 있는 취약성을 내포하고 있다. 위 취약성은 OWASP Cloud-Native Application Security Top 10[7]의 CNAS-6 에 해당한다.

<표 1> 도구별 탐지한 취약점

Vulnerability \ Tool		Snyk	CFRipper	Cfn-nag
Security Groups	Authorized Ports Only	o	o	o
	High-Risk Port Restrictions	x	o	o
	Port Ingress Restriction	o	o	o
	VPC Default Security Group Restrictions	x	o	o
CloudFormation Stacks	CloudFormation Integration with Simple Notification Service (SNS)	x	x	x
EC2 Instances and Subnets	No Automatic Public IP Assignment	o	o	x

표 1 은 취약점 유형을 기준으로 보안 요구 사항에 가장 적합한 도구를 결정하는 데 이바지한다. 예를 들어, 포트 수신 제한에 대한 취약점을 분석한다면 Snyk 또는 CFRipper 가 유용한 반면, EC2 인스턴스 및 서브넷에 퍼블릭 IP 를 자동으로 할당하는 데 중점을 둔다면 cfn-nag 가 적합할 것이다.

이러한 정적 분석 도구의 특정 기능을 비교함으로써 잠재적 보안 문제를 다루는 세분화된 보안 접근 방식을 만드는 데 이바지할 수 있다.

향후 연구에서는 OWASP Cloud-Native Application Security Top 10 중 클라우드 컨테이너와 사용자 계정 인증 요소와 관련한 취약한 파일 또한 분석할 예정이다. 본 연구에서 제시한 표와 추후 진행할 연구 기반으로 클라우드 환경 내에서 보안 위협을 사전 탐지할 수 있는 솔루션을 산출하고자 한다.

참고문헌

[1] Kumar M., Mishra S., Lathar N.K., and Singh P., “Infrastructure as Code (IaC): Insights on Various Platforms”, Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022, pp. 439-449, Jan. 2023.

[2] “AWS, AWS CloudFormation”, accessed on Mar. 20, 2024, https://aws.amazon.com/ko/cloudformation/.

[3] Juncal Alonso, Radosław Piliszek, and Matija Cankar, “Embracing IaC Through the DevSecOps Philosophy Concepts, Challenges, and a Reference Framework”, 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 56-62, Jan. 2023.

[4] “Snyk”, accessed on Mar. 20, 2024, https://github.com/snyk/snyk-iac-cloudformation.

[5] “CFRipper”, accessed on Mar. 20, 2024, https://github.com/Skyscanner/CFRipper.

[6] “Cfn-nag, Stelligent”, accessed on Mar. 20, 2024, https://github.com/stelligent/cfn_nag.

[7] “OWASP, OWASP Cloud-Native Application Security Top 10”, accessed on Mar. 20, 2023, https://owasp.org/www-project-cloud-native-application-security-top-10/.