사이드카 인젝션 공격: 쿠버네티스 환경에서 Mutating Webhook 악용

조치현[1], 이승수[2]

인천대학교 (학부생[1], 교수[2])

Sidecar Injection Attack: Exploiting the Mutating Webhook in Kubernetes

Chi Hyeon Jo[1], Seungsoo Lee[2]

Incheon National University (Undergraduate student[1], Professor[2])

요 약

쿠버네티스의 많은 프로젝트에서 Mutating Webhook을 사용함으로써, 자동화 및 리소스의 일관성 유지 등의 이점을 얻고 있다. 하지만, 공격자에 의해 악용되는 Mutating Webhook은 클러스터 전반의 보안을 위협하는 수단이 될 수 있다. 이러한 공격적인 특성은 리소스에 대한 요청을 변형(mutate)할 수 있는 Mutating Webhook의 기능으로부터 기인한다. 본 논문은 Mutating Webhook을 악용하는 사이드카 인젝션 공격 시나리오를 통해 공격자가 클러스터에 미칠 수 있는 영향력을 보여준다. 이를 통해, 안전한 Mutating Webhook 사용의 중요성을 강조하고, 악성 Mutating Webhook으로부터 클러스터를 안전하게 보호하기 위한 전략도 소개한다.

I. 서론

쿠버네티스는 현재 클라우드 네이티브 환경에서 가장 대표적인 컨테이너 오케스트레이션 플랫폼 중 하나이다. 많은 기업이 효율적인 리소스 관리 및 용이한 확장성을 비롯한 여러 장점으로 인해 쿠버네티스를 채택하고 있다.

Admission Control은 쿠버네티스 클러스터에 대한 API 요청을 검증 및 제어하는 과정이며, Mutating Webhook[1]은 이러한 과정을 확장할 수 있는 메커니즘이다. 특히, Mutating Webhook은 클러스터 리소스에 대한 요청을 특정한 값으로 변경할 수 있으며, 자동화, 리소스의 일관성 유지, 보안 강화 등의 기능을 제공한다. 이러한 Mutating Webhook을 활용해 쿠버네티스 기반의 대표적인 프로젝트인 Istio는 사이드카 프록시를 파드에 자동으로 추가하여 서비스메시 및 추가 보안 기능을 제공한다.

하지만, Mutating Webhook이 공격자에 의해 악용된다면, 클러스터 전반의 보안을 심각하게 위협하는 수단이 될 수 있다. 본 논문에서는, Mutating Webhook을 악용한 사이드카 인젝션 공격을 소개한다. 공격자가 미리 준비한 악의적인 Mutating Webhook은 피해자의 클러스터에 악성 컨테이너를 루트 권한의 사이드카 컨테이너로 실행하도록 클러스터 리소스 사양을 수정한다. 이후 사이드카 컨테이너는 은밀하게 피해자 호스트에 공격자의 서버와 통신할 수 있는 에이전트를 심어 공격자가 피해자 호스트와 연결을 지속적으로 유지할 수 있도록 한다. 최종적으로 에이전트를 통해 수집한 클러스터 내부의 자격 증명 정보를 바탕으로 피해자 클러스터의 제어 권한을 갖게 된다.

본 논문이 기여하는 바는 다음과 같다.

1. Mutating Webhook을 악용한 사이드카 인젝션 공격 시나리오를 소개하여, 안전한 Mutating Webhook 사용의 중요성을 강조한다.
2. 악의적인 Mutating Webhook으로부터 클러스터를 보호하기 위한 방어 기법을 소개한다.

II. 배경지식

2.1 Mutating Webhook

Mutating Webhook은 쿠버네티스에서 요청이 처리되기 전에 클러스터 리소스를 규칙에 맞게 수정할 수 있는 기능을 제공한다. 쿠버네티스의 다양한 프로젝트들이 이 기능을 활용해 자동화, 일관성 유지, 보안 강화 등의 이점을 누리고 있다. Mutating Webhook을 사용하기 위해, 리소스를 수정하는 Mutating Webhook 서버와 이를 쿠버네티스가 인식할 수 있도록 정의한 구성 파일을 배포해야 한다.

2.2 사이드카 패턴

사이드카 패턴은 메인 컨테이너 외에 보조적인 기능을 수행하는 서브 컨테이너를 파드에 포함하는 패턴이다. 이러한 서브 컨테이너를 사이드카 컨테이너라고 부르며, 이는 네트워크 트래픽 제어, 모니터링 등의 다양한 목적을 위해 쿠버네티스에서 활용된다. 특히, InitContainers로 배포된 사이드카 컨테이너는 메인 컨테이너가 시작되기 전에 실행되어, 파드에 필요한 초기 작업을 수행하고 종료된다.

III. 사이드카 인젝션 공격

3.1 가정

본 논문에서는 악의적인 Mutating Webhook이 미리 클러스터에 배포되었다고 가정하며, 이는 최근 발표된 클라우드 보안 보고서[2]에 기인한다. 해당 보고서는 클라우드 환경의 가장 큰 보안 위협 중 하나로 사용자의 클러스터 구성 및 리소스 배포의 부주의로 인한 구성오류(misconfiguration)에 대한 심각성을 강조한다.

따라서 공격자는 사전에 사이드카 인젝션 공격을 위해 악의적인 Mutating Webhook을 준비한 뒤, 정상적인 쿠버네티스 리소스 배포 파일에 심어 공개적인 저장소에 배포하여 사용자의 부주의를 유도해 해당 Mutating Webhook 리소스를 사용자의 클러스터에 설치한다.

3.2 공격 시나리오

그림 3은 악의적인 Mutating Webhook을 활용한 사이드카 인젝션 공격 흐름을 나타낸다.

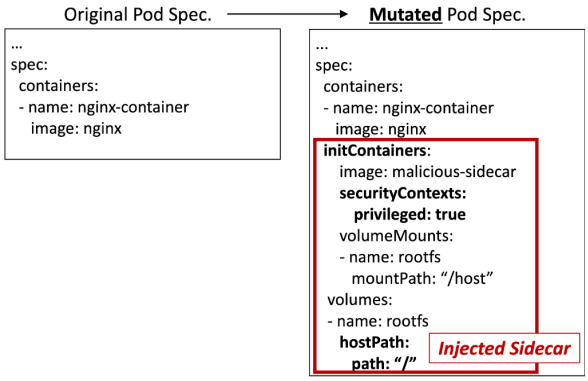

그림 1. 사이드카 인젝션 전/후 파드 사양 비교

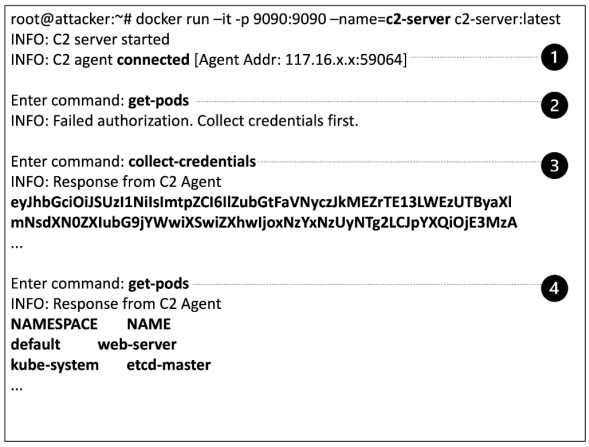

그림 2. 공격자 터미널

1) 악의적인 사이드카 컨테이너 배포. (1) 클러스터 관리자는 클러스터 제어 도구인 kubectl을 사용해 파드 생성을 요청한다. (2) 미리 설치된 악의적인 Mutating Webhook 서버는 쿠버네티스 API 서버로부터 파드 생성 요청을 전송받는다. 이후, 파드의 사양에 악성 사이드카 컨테이너를 추가하여 API 서버로 반환한다. 그림 1은 악성 사이드카 컨테이너를 주입함으로써 변경된 파드 사양을 나타낸다. 악성 사이드카 컨테이너는 특수 권한을 가지며, 호스트의 루트 파일시스템을 마운트한다. (3) 쿠버네티스 API 서버는 수정된 사양으로 파드를 배포한다.

2) C2 에이전트 배포. (4) 악의적인 사이드카 컨테이너는 컨테이너에서 호스트로 탈출하여, 원격 공격자로부터 명령을 전달받아 수행할 루트권한을 갖는 C2 에이전트를 마스터 노드에 배포한다. (5) 이때, InitContainers 사양으로 배포된 악의적인 사이드카 컨테이너는 C2 에이전

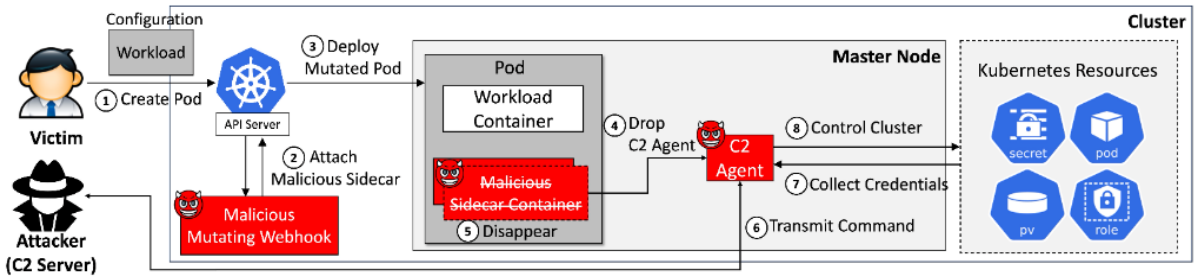

그림 3. 사이드카 인젝션 공격 시나리오

트를 노드에 설치하고 즉각적으로 종료한다.

기본적으로, 클러스터 관리자는 "kubectl get pods" 명령어를 통해 파드에 포함된 컨테이너의 개수를 확인할 수 있다. 하지만, Init 컨테이너는 지정한 명령어를 파드 생성 단계에서 수행하고 빠르게 종료되며, 컨테이너 개수에 포함되지 않아 클러스터 관리자에 의해 쉽게 감지되지 않는다는 특징이 있다.

3) 클러스터 제어 권한 획득. (6) 원격 공격자는 C2 서버를 통해 C2 에이전트에게 자격 증명 수집과 같은 명령을 전달한다. (7) C2 에이전트는 호스트의 루트 권한을 가지고 있으므로, 노드에서 실행 중인 파드의 자격 증명을 수집할 수 있다. (8) 최종적으로, 공격자는 C2 에이전트를 통해 획득한 자격 증명을 이용하여 전체 클러스터를 제어할 수 있다.

3.3 공격 수행 결과

그림 2는 공격자가 C2 서버를 사용해 클러스터를 제어하는 터미널 화면을 나타낸다. (1) 공격자가 피해자 노드에 배포한 C2 에이전트가 C2 서버와 세션을 연결한다. (2) 이후, 공격자가 C2 에이전트를 배포하는 것에 성공하였지만, 초기 단계에는 파드를 조회할 수 있는 권한이 없는 것을 볼 수 있다. (3) 이어서 공격자는 C2 에이전트에게 인가를 위한 자격 증명을 수집할 것을 명령한다. (4) 공격자는 획득한 자격 증명을 사용해 피해자 클러스터 내부에서 실행 중인 파드를 조회한다. 이를 통해, C2 서버가 악의적인 Mutating Webhook 서버에 의해 배포된 C2 에이전트에게 제어 명령을 전달함으로써, 공격자가 클러스터 제어 권한을 획득할 수 있다는 것을 보였다

IV. 방어 기법

은밀하게 수행되는 사이드카 인젝션 공격을 방어하기 위해 Validating Webhook을 활용할 수 있다. Validating Webhook은 쿠버네티스 요청이 처리되기 전에 요청의 사양이 관리자가 정의한 규칙을 준수하는지 검증한다. 3.2 절의 공격 시나리오에서 공격자는 악성 사이드카 컨테이너 실행을 위해 파드의 사양을 수정하였다. Validating Webhook은 정책을 기반으로 파드의 사양이 적절한지 검증할 수 있으며, 특수 권한을 가진 사이드카 컨테이너의 실행을 거부할 수 있다. 따라서, Validating Webhook을 적절하게 사용한다면, 악의적인 Mutating Webhook으로부터 클러스터를 안전하게 보호할 수 있다.

V. 결론

본 논문에서는 Mutating Webhook이 공격자에 의해 악용되었을 때, 클러스터 전체에 악의적인 영향을 미칠 수 있는 사이드카 인젝션 공격 및 방어 기법을 소개했다. 향후 연구에서는 기밀 컨테이너를 대상으로 보다 정교한 사이드카 인젝션 공격과 방어 기법에 대해 연구할 예정이다.

[참고문헌]

[1] Mutating Admission Webhook, May 2024, [online] Available: https://kubernetes.io/docs/reference/access-authn-authz/admission-controllers/#mutatingadmissionwebhook

[2] Red hat, The state of Kubernetes security report, 7p, June, 2024