SUSE

Guide

쿠버네티스 보안을 위한 최상의 가이드

쿠버네티스 보안에 대한 최상의 가이드

쿠버네티스 파이프라인 보호하기

컨테이너 배포 보안의 중요성

쿠버네티스와 같은 컨테이너와 툴을 사용하여 기업은 애플리케이션 배포의 많은 부분을 자동화할 수 있으며, 이는 많은 비즈니스상의 이점을 제공합니다. 그러나 이러한 새로운 전략은 기존의 시스템 운용 환경과 마찬가지로 해커 및 내부자 공격에 취약합니다. 랜섬웨어, 암호화폐 채굴 악성코드, 데이터 도난 및 서비스운영 방해와 같은 공격들은 프라이빗 클라우드와 퍼블릭 클라우드의 새로운 컨테이너 기반 가상화 환경에서도 계속될 것입니다.

이러한 상황을 악화시키는 것은, 기업의 소중한 자산을 탈취하기 위해 쿠버네티스나 퍼블릭클라우드의 관리 컨테이너 서비스와 같은 새로운 툴과 테크놀로지 자체가 공격자들이 통과하는 게이트웨이로서 공격을 받게 된다는 것입니다. 최근에 일어난 쿠버네티스 중간자 취약성 그리고 테슬라 시스템에 대한 취약성 공격은 향후 수개월, 수년에 걸쳐 급증할 것으로 예상되는 많은 컨테이너 기술 기반을 타겟으로하는 잠재적 공격의 시작에 불과합니다.

극적으로 동적인 컨테이너의 특성으로 인해 다음과 같은 보안 문제가 발생할 수 있습니다.

1. **CI/CD 파이프라인에서 시작된 취약성.** 오픈 소스 구성요소의 과도한 사용과 계속되는 중대한 취약성 문제의 발견은 빌드 단계, 레지스트리 및 프로덕션에서의 컨테이너 이미지에 영향을 미칩니다.
2. **서버간 트래픽의 폭발적 증가.** 전통적인 애플리케이션은 방화벽과 호스트 보안 툴로 보호가 가능하지만, 컨테이너는 서버간 트래픽이나 인터넷 트래픽을 동적으로 증가시킬 수 있어 공격에 대한 감시가 어려울 수 있습니다.
3. **공격받을 수 있는 범위의 증가.** 각 컨테이너는 서로 다른 공격 범위와 각각의 다른

취약성을 가질 수 있습니다. 쿠버네티스와 Docker와 같은 컨테이너 오케스트레이션 툴에 대한 추가적인 공격 범위도 고려해야 합니다.

4. **변화를 따라잡기 위한 보안의 자동화.** 오래된 보안 모델과 툴들은 지속적으로 변화하는 컨테이너 환경을 따라갈 수 없습니다. 쿠버네티스의 자동화된 특성으로 인해 컨테이너와 포드가 몇 분 또는 몇 초 만에 나타났다 사라질 수 있습니다. 새로운 네트워크 접속을 포함할 수 있는 어플리케이션의 작동방식은 강화된 보안 정책에 즉시 반영되어야 합니다. 컨테이너의 보안을 확보하기 위해서는 차세대 자동 보안 툴이 필요하며, 파이프라인 초기에 보안 정책을 선언하고 코드로서 관리해야 합니다.

컨테이너가 제한된 기능과 특수 인터페이스를 가지고 있기 때문에 기본적으로 기존 애플리케이션보다 더 안전하다는 주장 있지만, 이는 사이버 범죄자와 해커들이 취약성이 없으며 가능한 모든 위협 벡터에 대해 잠긴 코드와 인프라에 대해 오래된 기술을 사용하여 공격을 할 경우에만 해당됩니다. 물론 해커의 이와 같은 공격을 하지는 않겠지만, 이러한 공격에 대한 실시간 모니터링이 필요합니다. 시간이 경과하고 경험이 쌓여감에 따라, 공격자의 정교함은 항상 새로운 인프라의 대응에도 불구하고 그보다 더 앞서 나가게 됩니다. 공격자들은 컨테이너를 공격하는 새로운 방법을 끊임없이 개발하고 있습니다.

보안 관련하여 귀사의 쿠버네티스 팀에게 해야할 질문들

- 구축 단계를 포함하여, 파이프라인의 초기 단계에서 중요한 취약성을 제거하는 (해결방법과 함께)프로세스가 있습니까?
- 쿠버네티스 포드가 도입되어 있는지 여부를 확인할 수 있습니까? 예를 들어 애플리케이션 포드 또는 클러스터가 서로 어떻게 통신하는지 알고 계십니까?
- 컨테이너 사이의 서버간 트레픽에서 비정상적인 작동을 탐지할 수 있는 방법이 있습니까?
- 모든 포드가 정상적으로 작동하는지 확인하는 방법을 알고 있습니까?
- 내부 서비스 포드 또는 컨테이너가 내부적으로 포트를 검색하기 시작하거나 외부 네트워크에 무작위로 연결하려고 할 때 어떻게 경고를 받습니까?
- 공격자가 컨테이너, 포드 또는 호스트에 대한 기반을 확보했는지 어떻게 알 수 있습니까?

- 예를 들어 Layer 7에서 네트워크 접속 확인과 컨테이너화 되지 않은 배포와 동일한 수순의 검사가 가능합니까?
- 포드 또는 컨테이너 내부에서 무슨 일이 일어나고 있는지 감시하여 악용 가능성이 있는지 여부를 판단할 수 있습니까?
- 쿠버네티스 클러스터에 대한 액세스 권한을 검토하여 잠재적인 내부자 공격 벡터를 파악했습니까?
- 쿠버네티스 서비스, 액세스 제어(RBACs) 및 컨테이너 호스트를 잠그기 위한 체크리스트가 있습니까?
- 컴플라이언스 정책이 있는 경우 런타임에 컴플라이언스를 어떻게 적용합니까? 예를 들어 내부 포드 통신에 대한 암호화가 필요함에도 불구하고 포드가 암호화 채널을 사용하지 않는 것을 알아낼 방법이 있습니까?
- 애플리케이션 통신에 대해 트러블 슈팅 하거나 포렌식 데이터를 기록할 때 문제가 되는 포드를 찾아 로그를 캡처하려면 어떻게 해야 합니까? 가공되지 않은 원시 트래픽을 캡처하고 사라지기 전에 신속하게 분석하려면 어떻게 해야 합니까?

이 가이드에서는 런타임 보안 자동화에 특히 중점을 두고 쿠버네티스 및 컨테이너 배포를 보호하기 위한 개요를 제공합니다.

> 먼저, 쿠버네티스가 어떻게 작동하고 네트워킹이 어떻게 처리되는지 이해하는 것이 중요합니다.

쿠버네티스는 어떻게 작동하는가?

기본원리

쿠버네티스에 익숙하지 않으신 분들을 위한 주요 개념과 용어에 대한 소개입니다.

쿠버네티스는 컨테이너의 배포, 업데이트 및 모니터링을 자동화하는 조정 툴입니다. 쿠버네티스는 다음과 같은 대부분의 주요 컨테이너 관리와 클라우드 플랫폼에서 지원됩니다. Red Hat OpenShift, Docker EE, Rancher, IBM Cloud, AWS EKS, Azure, SUSE

CaaS, Google Cloud. 다음은 쿠버네티스에 대해 알아야 할 주요 사항입니다.

- **마스터 노드.** 쿠버네티스 워커 노드 클러스터 및 노드상의 포드 전개를 관리하는 서버. 노드는 물리적 시스템이나 가상 시스템입니다.
- **워커 노드**. 또는 미니언이라고도 불리는 이들 서버는 일반적으로 응용 프로그램 컨테이너 및 에이전트나 프록시 등의 기타 쿠버네티스 컴포넌트를 실행합니다.
- **포드.** 쿠버네티스에서의 배치와 주소 지정의 단위. 포드는 자체 IP 주소를 가지며 하나 이상의 컨테이너(일반적으로 1개)를 포함할 수 있습니다.
- **서비스.** 서비스는 기본 포드에 대한 프록시 역할을 하며 복제된 포드 사이에서 요청을 로드 밸런싱할 수 있습니다. 서비스는 또한 외부로부터 다음을 제공할 수 있습니다. 또한 서비스는 외부 IP 또는 노드 포트를 정의하여 하나 이상의 포드에 액세스할 수 있는 외부 액세스 엔드포인트를 제공할 수 있습니다. 쿠버네티스는 DNS 서비스, 라우터 및 로드 밸런서도 제공합니다.

쿠버네티스 클러스터 관리에 사용되는 주요 컴포넌트에는 API 서버, Kubelet 등이 있습니다. 쿠버네티스는 브라우저 기반 관리 콘솔인 쿠버네티스 Dashboard도 지원합니다(옵션). 이러한 컴포넌트는 공격 대상이 될 수 있습니다. 예를 들어, 테슬라 해킹 사건에서는 보호되지 않은 쿠버네티스 콘솔을 이용하여 암호 마이닝 소프트웨어를 설치했습니다.

쿠버네티스 역할 기반 액세스 제어

쿠버네티스 역할 기반 액세스 제어(RBACs)는 시스템 리소스의 세밀한 관리를 제공합니다. RBACs는 애플리케이션 워크로드와 쿠버네티스 시스템 리소스에 대한 액세스를 관리할 수 있습니다. OpenShift와 같은 관리 툴은 추가적인 제어기능을 사용할 수 있지만, 기본적으로

쿠버네티스의 자체 보안 제어 기능을 사용합니다. API 서버나 애플리케이션 워크로드와 같은 쿠버네티스 구성 요소에 대한 무단 액세스를 방지하기 위해 액세스 제어를 적절하게 구성하는 것이 중요합니다.

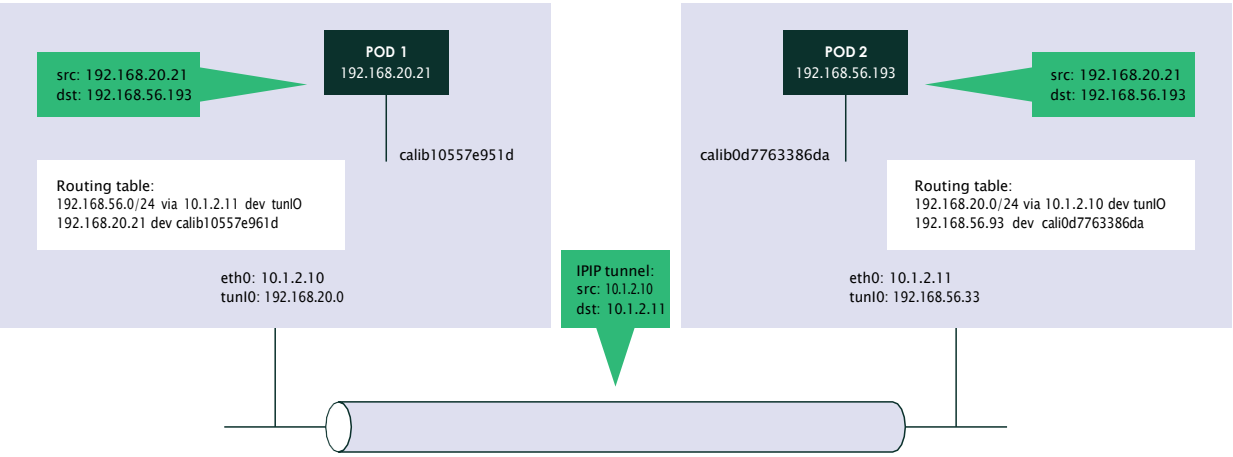
기본 쿠버네티스 네트워킹

쿠버네티스의 주요 네트워킹 개념은 모든 포드에 자체 라우팅 가능한 IP 주소가

할당된다는 것입니다. 쿠버네티스(실제로는 쿠버네티스의 네트워크 플러그인)는 호스트 간의 모든 요청을 내부적으로 적당한 포드에 라우팅합니다. 쿠버네티스 포드에 대한 외부로부터의 액세스는 쿠버네티스가 적절한 포드로 라우팅하는 서비스, 로드 밸런서 또는 입력 컨트롤러를 통해 가능합니다.

쿠버네티스는 iptables를 사용하여 포드 간(그리고 노드 간) 네트워크 연결을 제어하고 많은 네트워킹 및 포트 포워딩 규칙을 처리합니다. 이렇게 하면 클라이언트는 쿠버네티스 서비스에 접속하기 위해 IP 주소를 추적할 필요가 없습니다. 또, 각 포드에는 고유의 IP 주소가 있어 컨테이너가 네이티브 포트로 수신할 수 있기 때문에 포트 맵핑은 큰 폭으로 심플화(대부분 제거)됩니다.

이러한 오버레이 네트워킹이 모두 쿠버네티스에 의해 동적으로 처리되기 때문에 네트워크 트래픽을 모니터링하는 것은 매우 어려우며 보안은 더더욱 어렵습니다. 다음은 쿠버네티스 네트워킹의 동작 예를 보여줍니다.



위의 그림은 패킷이 다른 노드의 포드 사이를 통과하는 방법을 보여 줍니다. 이 예에서는 Calico CNI 네트워크 플러그인이 사용됩니다. 네트워크 플러그인마다 포드 IP 주소 할당 방법(IPAM), iptables 규칙 및 크로스노드 네트워킹 설정 방법 및 노드 간 라우팅 정보 교환 방법에 대한 접근 방식이 다릅니다.

1. CNI 네트워크 플러그인은 쿠버네티스로부터 컨테이너가 배포되었음을 통지 받으면 IP 주소를 할당하고 노드에서 적절한 iptables 및 라우팅 규칙을 설정합니다.
2. Pod1은 Pod2의 IP 또는 Pod2의 서비스 IP 중 하나를 수신처로 사용하여 패킷을 Pod2로 전송합니다 (그림에서는 Pod2의 IP를 사용하고 있습니다).

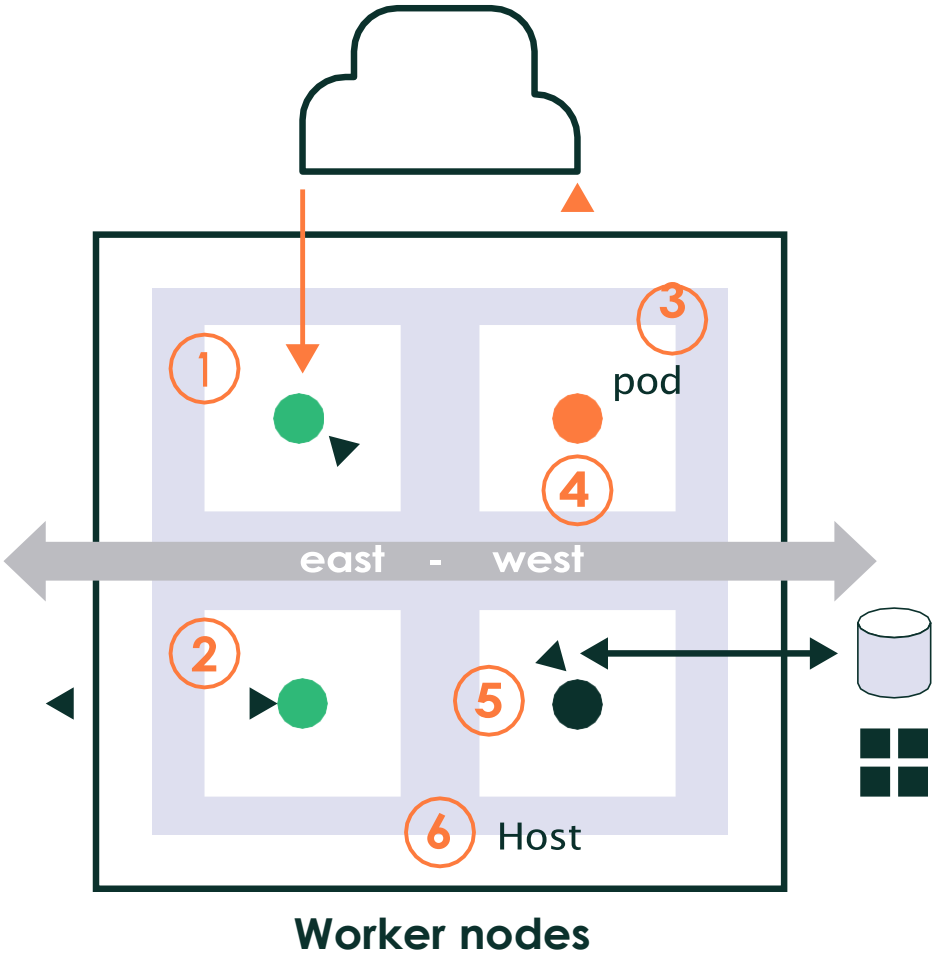
3. 서비스 IP 가 사용되고 있는 경우, kube-proxy는 로드밸런싱과 DNAT를 실행하고, 행선지 IP를 리모트 포드의 IP 로 변환합니다.
4. 노드의 라우팅 테이블에 따라 패킷이 어디로 라우팅 되어야 할지가 결정됩니다.
    a. 수신처가 같은 노드상의 로컬 포드일 경우 패킷은 포드의 인터페이스로 직접 전송됩니다.
    b. 그렇지 않으면 네트워크 플러그인에 의해 오버레이 네트워킹 또는 L3 라우팅 메커니즘이 사용되는지에 따라 패킷이 적절한 인터페이스로 전송됩니다.
    c. 위의 그림에서 패킷은 IPIP 터널인터페이스로 송신되어 IPIP 터널헤더로 캡슐화 됩니다.
5. 패킷이 행선지 노드에 도달하면, 캡슐화가 제거됩니다.
6. 리모트 노드의 라우팅 테이블은 패킷을 수신처인 Pod2에 라우팅합니다.

이러한 복잡한 라우팅과 가능한 NAT 그리고 캡슐화의 발생과 네트워크 플러그인에 의한 관리로 인해, 네트워크 트래픽에 공격 및 접속 위반이 없는지 검사하고 모니터링하는 것은 매우 어렵습니다.

쿠버네티스의 취약성과 공격 벡터

포드에서 실행되고 있는 쿠버네티스 컨테이너에 대한 공격은 네트워크를 통해 외부에서 발생하거나 내부자에 의해서도 발생할 수 있는데, 내부자에는 피싱 공격의 피해자도 포함되며 이때는 내부자의 시스템이 공격의 도관이 됩니다. 다음은 몇 가지 예입니다.

1. **컨테이너의 손상.** 애플리케이션 설정 오류 또는 취약성에 의해 공격자는 컨테이너에 들어가 네트워크, 프로세스 제어 또는 파일 시스템의 취약점 탐색을 시작합니다.
2. **포드 간의 무단 연결.** 손상된 컨테이너는 시스템을 탐색하거나 공격을 감행하기 위해 동일한 호스트 또는 다른 호스트에서 실행 중인 다른 포드와 연결을 시도할 수 있습니다. 비록 Layer 3 네트워크 제어에서는 화이트리스트 포드의 IP 주소를 어느 정도 보호할 수 있지만 신뢰할 수 있는IP 주소에 대한 공격은 Layer 7 네트워크 필터링으로만 적발할 수 있습니다.
3. **포드에서 데이터 유출.** 데이터 탈취는 명령과 제어 서버에 연결된 포드에서의 리버스 쉘과 민감한 데이터를 숨기기위한 네트워크 터널링과 같은 기술의 조합을 사용하여 이루어집니다.

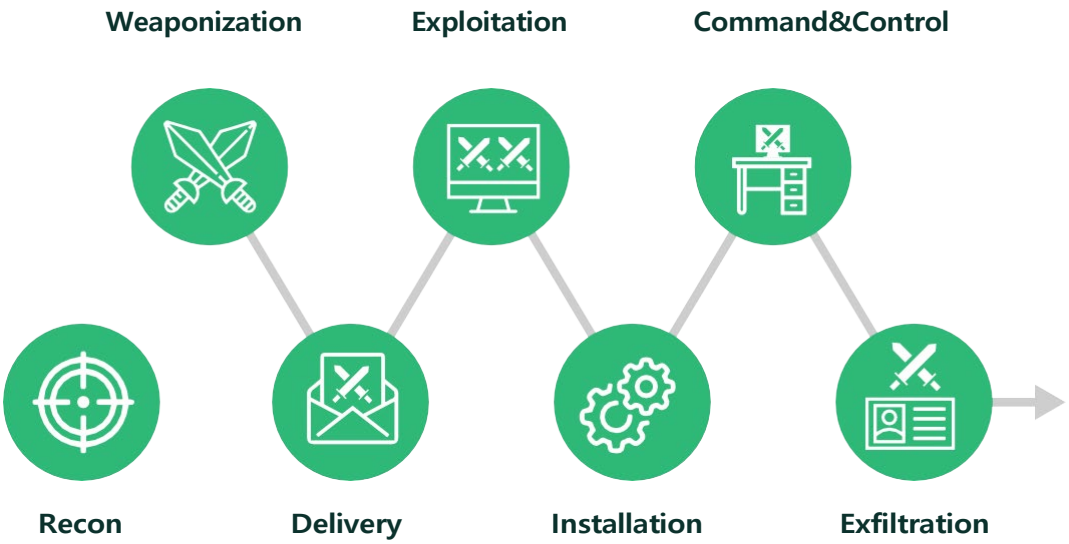


4. **손상된 컨테이너가 악성 프로세스를 실행.** 컨테이너는 일반적으로 제한된 프로세스 세트를 실행하지만, 손상된 컨테이너는 암호 마이닝과 같은 악성 프로그램이나 네트워크 포트 검색과 같은 의심스러운 프로세스를 시작하거나, 혹은 이전에 볼 수 없었던 바이너리(프로세스 악용)를 주입할 수 있습니다.
5. **컨테이너 파일 시스템의 손상.** 공격자는 취약한 라이브러리/패키지 버전을 설치하여 컨테이너를 공격할 수 있습니다. 민감한 파일들도 또한 변경할 수 있는데, 일단 파일 변경이 가능해지면, 공격자는 루트 권한 탈취 등을 시도할 수 있습니다.
6. **손상된 워커 노드.** 운용 중인 컨테이너와 마찬가지로 호스트 자체도 손상될 수 있습니다. 일례로, Dirty Cow 리눅스 커널 취약성으로 인해 사용자가 루트 권한을 갖게 되는 일도 발생했었습니다.

킬체인 공격

공격자는 가장 피해가 큰 공격인 킬 체인 혹은 일련의 악의적 행동들을 통해 공격의 목표를 달성합니다. 이러한 이벤트는 몇 초 이내에 빠르게 발생하거나 며칠, 몇 주 또는 몇 달에 걸쳐 발생할 수 있습니다.

킬 체인에서 이벤트를 탐지하려면 서로 다른 리소스가 사용되므로 여러 계층의 보안 모니터링이 필요합니다. 프로덕션 환경에서 탐지의 가능성을 최대한 높이기 위해 감시해야 할 가장 중요한 벡터는 다음과 같습니다.



- **네트워크 검사.** 공격자는 일반적으로 네트워크 연결을 통해 침입하여 네트워크를 통해 공격 범위를 넓혀갑니다. 네트워크는 공격을 감지할 수 있는 첫 번째 기회와, 공격자의 움직임을 탐지할 수 있는 기회, 그리고 데이터 탈취 활동을 포착할 수 있는 마지막 기회를 제공합니다.
- **컨테이너.** 각 컨테이너의 프로세스 및 syscall 활동을 모니터링하여 의심스러운 프로세스가 시작되었는지 또는 권한을 상승시키고 컨테이너에서 빠져나오려는 시도가 있었는지 여부를 확인하여 응용프로그램 또는 시스템 탈취를 탐지할 수 있습니다. 파일 무결성 모니터링 및 액세스 제한은 파일, 패키지 또는 라이브러리를 수정하려는 시도도 탐지할 수 있습니다.
- **호스트 모니터링**. 전통적인 호스트(엔드포인트) 보안도 커널 또는 시스템 리소스에 대한 공격을 탐지하는데 유용합니다. 단, 적절한 보안 능력을 확보하려면 호스트 보안 툴이 쿠버네티스 및 컨테이너를 인식할 수 있어야 합니다. 예를 들어 새 호스트는 쿠버네티스 클러스터에 동적으로 진입할 수 있으며 쿠버네티스가 관리하는 보안 설정 및 툴을 유지할 수 있어야 합니다.

위의 위협 벡터와 더불어, 공격자는 쿠버네티스 API 서버나 콘솔 등의 배포 툴을 손상시켜 시크릿에 액세스하거나 실행 중인 포드를 제어할 수 있습니다.

쿠버네티스 인프라 자체에 대한 공격

응용 프로그램을 비활성화 혹은 중단시키거나 시크릿, 리소스 또는 컨테이너에 대한 액세스를 얻기 위해 해커는 API 서버나 Kubelets 같은 쿠버네티스 리소스를 손상시킬 수도 있습니다. 예를 들어, 테슬라 해킹은 보호되지 않은 콘솔을 이용하여 기반 인프라에 액세스하고 암호화 마이닝 소프트웨어를 실행했습니다.

API 서버 토큰을 탈취/해킹하거나 ID가 도난당하여 해커가 인증된 사용자를 가장하여 데이터베이스에 액세스할 수 있게 된 경우, 해커는 악성 컨테이너를 배포하거나 중요한 응용 프로그램을 중지시킬 수 있습니다.

오케스트레이션 툴 자체를 공격함으로써 해커는 실행 중인 응용 프로그램을 중단시키고 컨테이너를 실행하는 데 사용되는 기본 리소스를 제어할 수 있습니다. 쿠버네티스에는 etcd 또는 서비스 토큰에 대한 액세스와 같은 몇 가지 공개된 권한 상승 메커니즘이 있으며, 이를 통해 해커가 손상된 컨테이너에서 클러스터 관리자 권한을 얻을 수 있습니다.

이렇게 쿠버네티스의 중간자 취약성과 같은 보안 문제는 보안 전문가들이 우려하는 비교적 새로운 악성 보안 문제이며, 앞으로 더 새로운 보안 문제들이 생겨날 것입니다. 시스템의 취약성으로 인해 공격자는 외부 IP나 자주 사용되지 않는 옵션을 가진 쿠버네티스 기본 서비스 정의를 악용하여 중간자 공격을 시작할 수도 있습니다.

전체 파이프라인 보안

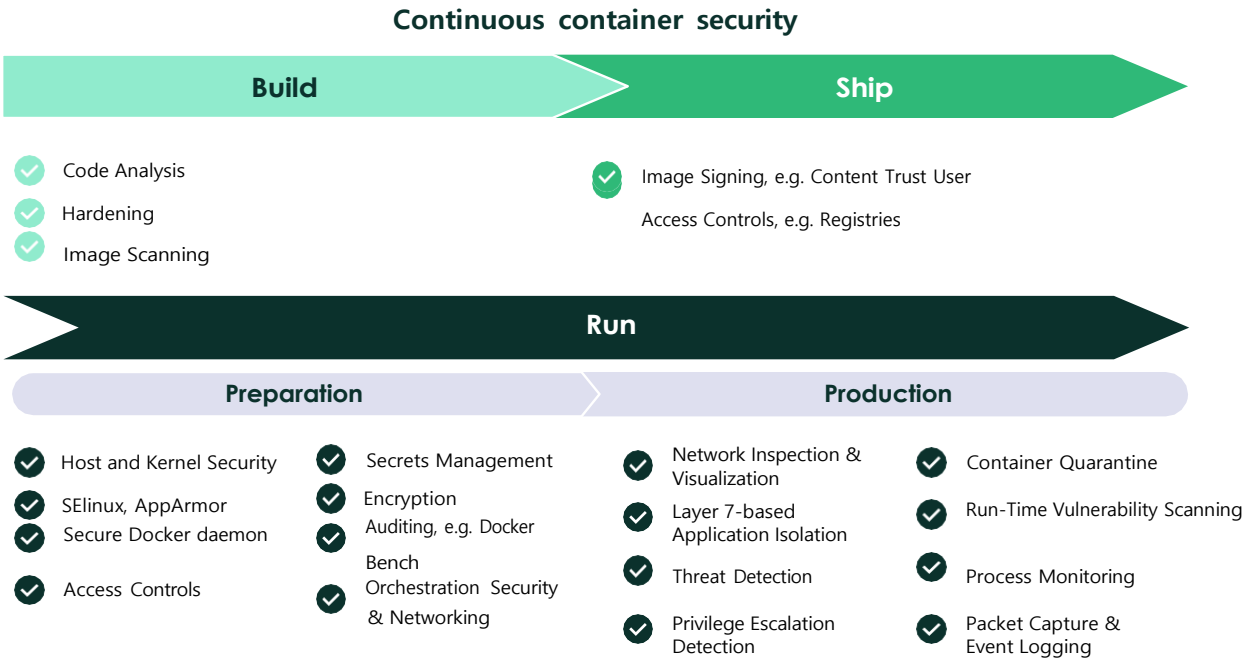
런타임 보안이 중간자 공격이나 다른 공격으로부터 어떻게 시스템을 보호할 수 있는지 알아보기 전에 보안을 전체 CI/CD 파이프라인에 어떻게 적용할 수 있는지를 살펴보겠습니다.

빌드 단계에서 코드 및 이미지 분석은 이미지가 배포용으로 승인되기 전에 알려진 취약성 및 컴플라이언스 위반을 제거하는 데 중요합니다.

출하 단계에서 적절한 액세스 제어를 활성화하고 이미지 배포를 제한하는 것은 파이프라인의 후반부에서 의도적이거나 실수로 취약성이 발생하지 않도록 하는 데 매우 중요합니다.

실행 단계에서는 준비 중인 호스트와 오케스트레이션 툴을 적절히 잠그고 분리시키는 것이 좋으며, 새로운 공격의 위험에 대한 노출을 줄이고 위험을 완화시키기 위해서는 컨테이너 환경에 대한 실시간 모니터링이 매우 중요합니다.

보안 팀은 항상 모든 단계에 걸쳐 완전한 보안을 제공하는 하나의 만능 보안툴을 원하지만, 파이프라인에는 보안을 적용해야 할 많은 계층과 단계가 있으며, 어떤 툴 하나가 이를 모두 담당해 낼 수 없습니다. 일반적으로 Red Hat OpenShift, Docker EE, Rancher, SUSE CaaS 및 AWS EKS와 같은 플랫폼은 빌드에서부터 출하 및 배포 이전 단계에 초점을 맞춘 보안 툴들과 기능을 제공하는 반면, 독립 보안 벤더는 런타임 보안을 포함하는 모든 단계에서 필요한 보안 툴들을 제공할 수 있습니다. 런타임 보안 툴은 정교한 네트워크 및 컨테이너 기반 공격 탐지 및 방지를 전문으로 해야 합니다. 이 안내서의 뒷부분에는 몇 가지 오픈 소스 컨테이너 보안 프로젝트에 대한 요약을 제공합니다.



CI/CD 파이프라인 보안

보안은 가능한한 초기에 CI/CD 파이프라인에 적용되어야 합니다. 대부분의 회사에서 보안 작업은 개발자가 컨테이너 이미지를 빌드할 때 시작됩니다.

개발자는 검사를 통해 이미지가 파이프라인을 통과하여 다음 단계로 이동하기 전에 수정해야 하는 중요한 취약성 또는 컴플라이언스 위반이 있는지 여부를 알 수 있습니다. 다음과 같은 기술적인 문제와 프로세스 문제가 모두 고려되어야 합니다.

- 파이프라인에서 검사가 어떻게 시작하고 실시되어야 하는가? Jenkins와 같은 대부분의 툴에서는 검사를 트리거하는 플러그인 또는 확장프로그램이 있습니다. 다른 툴들은 스크립트를 호출하여 API를 통해 검사를 시작하게 할 수 있습니다.
- 취약성을 평가하고 교정하기 위해 승인된 프로세스는 무엇입니까? 누가 통보를 받고 검토해야 합니까?
- 취약성 교정 요청이 이루어지기 위해 어떤 기준이 사용되어야 합니까? 중요도(높은 CVSS 점수 임계 값)를 기반으로 합니까?
- 언제 빌드 작업이 중단되도록 해야 합니까? 중요한 취약성에 대한 수정이 가능할 경우에는 빌드 작업을 멈추지만, 수정이 가능하지 않을 경우에는 빌드 작업 중단을 하지 않도록 합니까?
- 컴플라이언스 위반에 대해 유예 기간 및/또는 예외(면제)가 있어야 합니까? 예외 프로세스는 어떻게 됩니까?
- 프로덕션 레지스트리 또는 실행 중인 컨테이너와 같은 파이프라인 후반에서 취약성이 발견되면 이를 해결하기 위해 어떤 행동을 하도록 해야 합니까?

CI/CD 파이프라인에 검사 프로세스를 구축하는 것 외에도 다음과 같은 다른 보안 조치가 포함되어야 합니다.

- 내부자 악용 가능성을 줄이는 파이프라인 툴과 레지스트리에 대한 액세스 제어
- 취약하거나 허가되지 않은 이미지의 배포를 방지하거나 파이프라인에서 이미지가 더 이상 사용되지 않도록 하는 승인 제어
- 오픈 소스 구성 요소 및 코드 검색 툴에 대한 라이센스 제어와 같은 기타 회사 소프트웨어 관리 정책 시행

프로덕션을 위해 쿠버네티스 노드 준비

애플리케이션 컨테이너를 배포하기 전에 쿠버네티스 워커 노드에 대한 호스트 시스템을 잠가야 합니다. 다음 섹션에서는 호스트를 잠그는 가장 효과적인 방법을 설명합니다.

권장되는 배포 전 보안 단계

- 네임스페이스를 사용합니다.
- Linux 기능을 제한합니다.
- SELinux를 활성화합니다.
- Seccomp를 활용합니다.
- Cgroup을 구성합니다.
- R/O 마운트를 사용합니다.
- 최소한의 Host OS를 사용합니다.
- 시스템 패치를 업데이트합니다.
- CIS 벤치마크 보안 테스트를 실행합니다.

업데이트 및 확장 중에 보안 구성이 실수로 잘못되지 않도록 스테이징 및 프로덕션 환경에서 쿠버네티스 호스트를 지속적으로 감사하고 검사해야 합니다.

쿠버네티스 런타임 컨테이너 보안

프로덕션 운영 환경에서 컨테이너가 실행되면 컨테이너를 보호하기 위한 세 가지 중요한 보안 벡터는 네트워크 필터링, 컨테이너 검사 그리고 호스트 보안입니다.

네트워크 검사와 보안

컨테이너 방화벽은 기존 네트워크 보안 기술을 새로운 클라우드 네이티브 쿠버네티스 환경에

적용하는 새로운 유형의 네트워크 보안 제품입니다. 방화벽으로 컨테이너 네트워크를 보호하는 몇 가지 방법들은 다음과 같습니다.

- • IP 주소 및 포트를 기반으로 하는 Layer 3/4 필터링. 이 방법은 쿠버네티스 네트워크 정책이 규칙을 동적으로 업데이트하여 배포가 변경되고 확장될 때 이를 보호하도록 합니다. 단순한 네트워크 분할 규칙은 비즈니스 크리티컬 컨테이너 배포에 필요한 강력한 모니터링, 로깅 및 위협 탐지 기능을 제공하도록 설계되지 않았지만 승인되지 않은 연결로부터 어느 정도 보호할 수 있습니다.
- • WAF(Web Application Firewall) 공격 탐지는 웹 애플리케이션 방화벽의 기능과 유사하게 일반적인 공격을 탐지하는 방법을 사용하여 웹 대면 컨테이너(일반적으로 HTTP 또는 HTTPS 기반 애플리케이션)를 보호할 수 있습니다. 그러나 이는 HTTP를 통한 외부 공격에 대해서만 보호할 수 있으며, 내부 트래픽에 대한 멀티 프로토콜 필터링이 없습니다.
- • Layer 7 컨테이너 방화벽. 포드 사이의 트래픽에 대한 Layer 7 필터링 및 심층 패킷 검사가 포함된 컨테이너 방화벽은 네트워크 애플리케이션 프로토콜을 사용하여 컨테이너를 보호합니다. 컨테이너 방화벽은 또한 쿠버네티스와 같은 오케스트레이션 툴들과 통합되며, 자동화된 정책 작성을 위해 행동 학습을 활용합니다. 애플리케이션 프로토콜 화이트리스트와 DDoS, DNS 및 SQL 주입과 같은 일반적인 네트워크 기반 애플리케이션 공격에 대한 기본 제공 탐지 기능을 통한 보호가 이루어집니다. 또한 컨테이너 방화벽은 컨테이너 프로세스 모니터링 및 호스트 보안을 모니터링되는 위협 벡터에 통합할 수도 있습니다.

심층 패킷 검사(DPI) 기술은 컨테이너 방화벽에서 심층 네트워크 보안을 위해 필수적입니다. 일반적으로 공격에는 예측 가능한 공격 벡터가 사용됩니다. 즉, 잘못된 형식의 헤더를 가진 악성 HTTP 요청을 보내거나, XML(Extensible Markup Language) 개체 내에 실행 가능한 쉘 명령을 포함하는 것입니다. Layer 7 DPI 기반 검사는 이러한 방법을 찾고 인식할 수 있습니다. 이러한 기술을 사용하는 컨테이너 방화벽은 각 포드 연결을 통과하도록 허용할지 또는 차단되어야 할 공격인지 여부를 결정할 수 있습니다.

컨테이너의 동적 특성과 쿠버네티스 네트워킹 모델을 고려할 때, 기존 툴들을 네트워크 가시성, 포렌식 및 분석에 사용할 수 없습니다. 애플리케이션 디버깅을 위한 패킷 캡처나 보안 이벤트 조사와 같은 간단했던 작업들은 이제 더 이상 간단하지 않습니다. 네트워크 보안, 검사 및 포렌식 작업을 수행하려면 새로운 쿠버네티스와 컨테이너를 인식하는 툴들이 필요합니다.

컨테이너 검사

사이버 공격은 자주 권한 상승과 악의적인 프로세스를 활용하여 공격을 시작하거나 확산시킵니다. Linux 커널(예: Dirty Cow), 패키지, 라이브러리 또는 응용 프로그램 자체의 취약성 공격은 컨테이너 내에서 의심스러운 활동을 발생시킬 수 있습니다.

컨테이너 프로세스 및 파일 시스템 활동을 검사하고 의심스러운 동작을 탐지하는 것은 컨테이너 보안의 중요한 요소입니다. 포트 검색 및 리버스 쉘 또는 권한 상승과 같은 의심스러운 프로세스가 모두 탐지되어야 합니다. 내장된 탐지프로세스와 이전 활동을 기반으로 비정상적인 프로세스를 식별할 수 있는 기본 행동 학습 프로세스의 조합이 필요합니다.

컨테이너형 응용프로그램을 마이크로서비스 원칙을 염두에 두고 설계할 경우, 컨테이너의 각 응용프로그램은 제한된 기능 집합을 가지고 있으며 컨테이너는 필요한 패키지와 라이브러리만으로 구축되므로 의심스러운 프로세스와 파일 시스템 활동을 훨씬 쉽고 정확하게 탐지할 수 있습니다.

호스트 보안

컨테이너가 실행되는 호스트(예: 쿠버네티스 워커 노드)가 손상되면 여러 가지 유형의 부정적인 결과가 발생할 수 있습니다. 여기에는 다음이 포함됩니다.

- 루트로 권한 상승
- 보안 애플리케이션 또는 인프라 액세스에 사용되는 시크릿 도난
- 클러스터 관리자 권한 변경
- 호스트 리소스 손상 또는 하이잭킹(예: 암호화 마이닝 소프트웨어)
- API 서버 또는 Docker 데몬과 같은 중요한 오케스트레이션 툴 인프라의 종료
- 컨테이너 검사에 대해 이전 섹션에서 논의된 의심스러운 프로세스의 시작

컨테이너와 마찬가지로 호스트 시스템에서 이러한 의심스러운 활동을 모니터링해야 합니다. 컨테이너는 호스트와 같은 운영 체제 및 애플리케이션을 실행할 수 있으므로 컨테이너 프로세스 및 파일 시스템 작업을 모니터링하려면 호스트를 모니터링하는 것과 동일한 보안 기능이 필요합니다. 네트워크 검사, 컨테이너 검사 및 호스트 보안을 함께 사용하면 여러 벡터에서 킬 체인을 탐지할 수 있습니다.

쿠버네티스 시스템과 리소스 보안

쿠버네티스와 같은 오케스트레이션 툴과 그 위에 구축된 관리 플랫폼은 보호되지 않으면

공격에 취약해질 수 있습니다. 컨테이너 배포가 이전에는 존재하지 않았던 잠재적인 새로운 공격 표면을 노출되면 해커의 침입에 취약해지게 됩니다. Tesla 해킹과 Kublet 공격은 새로운 기술에 대한 지속적인 공격과 패치의 사이클이 될 것으로 예상되는 첫 번째 사례 중 하나입니다.

쿠버네티스와 관리 플랫폼을 공격으로부터 보호하려면 시스템 리소스에 대한 RBACs를 적절하게 구성하는 것이 중요합니다. 다음은 적절한 액세스 제어를 위해 검토하고 구성할 영역입니다.

1. **API 서버 보호.** API 서버에 대한 RBAC를 구성하거나 수동으로 방화벽 규칙을 만들어 무단 액세스를 방지합니다.
2. **kubelet 권한 제한.** Kubelets을 위해 RBAC를 구성하고 인증서 로테이션을 관리하여 Kubelet을 보호합니다.
3. **모든 외부 포트에 대한 인증.** 외부에서 액세스할 수 있는 모든 포트를 검토하고 불필요한 포트를 제거합니다. 필요한 외부 포트에 대한 인증을 실시합니다. 인증되지 않은 서비스의 경우 화이트리스트 소스에 대한 액세스를 제한합니다.
4. **콘솔 액세스를 제한하거나 제거.** 강력한 암호 또는 이중 인증으로 사용자 로그인이 적절하게 구성되지 않은 경우 콘솔/프록시 액세스를 허용하지 않습니다.

일반적으로 모든 역할 기반 액세스 제어는 신중하게 검토되어야 합니다. 예를 들어 클러스터 관리자 역할이 할당된 서비스 계정은 검토되어야 하고 반드시 필요한 계정에게만 주어져야 합니다.

이전에 논의한 것처럼, 워커 노드를 잠그기 위해 쿠버네티스 배포 인프라를 강력한 호스트 보안과 결합하면 공격으로부터 보호할 수 있습니다. 또한 모니터링 툴을 사용하여 인프라 서비스에 대한 액세스를 추적하여 무단 연결 시도 및 잠재적인 공격을 탐지하는 것이 권장됩니다.

Tesla Kubernetes 콘솔 공격의 예를 보면, 일단 워커 노드에 대한 액세스가 손상되면서 해커들이 암호화 마이닝 소프트웨어를 제어하기 위해 중국으로의 외부 연결을 만들었습니다. 컨테이너, 호스트, 네트워크 및 시스템 리소스에 대한 실시간 정책 기반 모니터링을 했다면 의심스러운 프로세스와 무단 외부 연결을 탐지할 수 있었을 것입니다.

쿠버네티스 환경을 위한 감사 및 컴플라이언스 – 보안 상태

컨테이너 기술과 쿠버네티스와 같은 툴의 급속한 발전으로 기업은 컨테이너 환경을 지속적으로 업데이트, 업그레이드 및 마이그레이션할 것입니다. 쿠버네티스 환경을 위해 설계된 일련의 보안 테스트를 실행하면 시스템 변경 시마다 보안이 저하되지 않도록 할 수 있습니다. 이렇게 함으로서 공격의 위험에 대해 인프라의 보안 상태를 평가할 수 있습니다. 더 많은 기업이 컨테이너로 마이그레이션함에 따라 인프라, 툴 및 배치의 변경으로 PCI와 같은 컴플라이언스 표준에 대한 재인증이 필요할 수도 있습니다.

다행히도 쿠버네티스용 CIS 벤치마크 및 Docker 벤치 테스트를 통해 쿠버네티스 및 Docker 환경에 대한 포괄적인 보안 상태를 점검할 수 있습니다. 이러한 테스트를 정기적으로 실행하고 예상 결과를 확인하는 작업을 자동화해야 합니다.

벤치마크 테스트가 가능한 영역들

- 호스트 보안
- 쿠버네티스 보안
- Docker 데몬 보안
- 컨테이너 보안
- RBACs의 올바른 구성
- 미사용 및 전송 중인 데이터 보안

또한 이미지 검사에는 이미지 보안과 관련된 CIS 벤치마크 테스트가 포함되어야 합니다. 추가 이미지 컴플라이언스 테스트를 통해 내장된 시크릿 및 파일 액세스(setuid/setgid) 위반이 있는지 이미지를 검사할 수 있습니다. 레지스트리와 프로덕션에서 이미지 및 컨테이너에 대한 취약성 검사는 알려진 공격을 방지하고 컴플라이언스를 달성하기 위한 핵심 구성 요소이기도 합니다. 빌드 프로세스와 CI/CD 파이프라인에 검사 프로세스를 통합하여 프로덕션으로 이동하는 모든 이미지가 검사되었는지 확인할 수 있습니다. 프로덕션에서 실행 중인 컨테이너와 호스트는 정기적으로 취약성 검사를 해야 합니다. 그러나 취약성 검사만으로는 컨테이너 런타임 배포를 보호하는 데 필요한 다중의 보안 벡터를 제공하기에는 충분하지 않습니다.

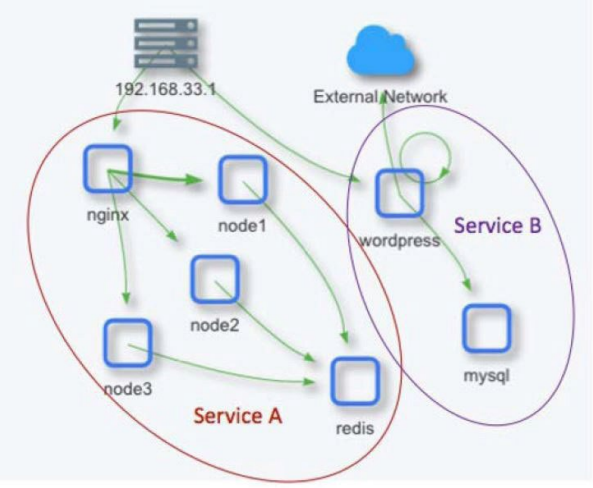
런타임 보안이 적용된 SUSE NeuVector 컨테이너 보안 플랫폼

오케스트레이션 및 컨테이너 관리 툴은 기본적인 RBACs 및 인프라 보안 기능을 제공하지만 보안 툴로 설계되지 않았습니다. 비즈니스 크리티컬한 배포를 위해서는 전문화된 쿠버네티스 보안 툴이 필요합니다. 특히, 세 가지 주요 보안 벡터(네트워크, 컨테이너 및 호스트)에 걸친 보안 문제를 해결할 수 있는 보안 솔루션이 필요합니다.

SUSE NeuVector는 다음과 같은 기능을 갖춘 고도로 통합된 자동화된 쿠버네티스용 보안 솔루션입니다.

- 빌드 단계와 레지스트리에서 이미지의 파이프라인 취약성 및 컴플라이언스 검사
- 취약하거나 허가되지 않은 이미지의 배포를 방지하는 승인 제어
- 네트워크, 컨테이너 및 호스트에 주소를 지정하는 다중 벡터 컨테이너 보안
- Layer 7 컨테이너 방화벽을 사용하여 서버간 트래픽과 서버 내외로 유출/유입되는 트래픽 보호
- 승인되지 않은 프로세스 및 파일 작업에 대한 컨테이너 보호 기능 제공
- 시스템 공격을 탐지하기 위한 호스트 보안
- 행동 학습을 통한 자동 정책 생성 및 자동 확장을 지원하는 적응형 시행
- 쿠버네티스 클러스터의 모든 컨테이너 또는 호스트에 대한 런타임 취약성 검사
- CIS 보안 벤치마크를 통한 컴플라이언스와 감사

SUSE NeuVector 솔루션은 쿠버네티스 또는 OpenShift, Rancher, Docker EE, IBM Cloud, SUSE CaaS, EKS 등과 같은 모든 오케스트레이션 시스템에서 배포 및 업데이트되는 컨테이너입니다.

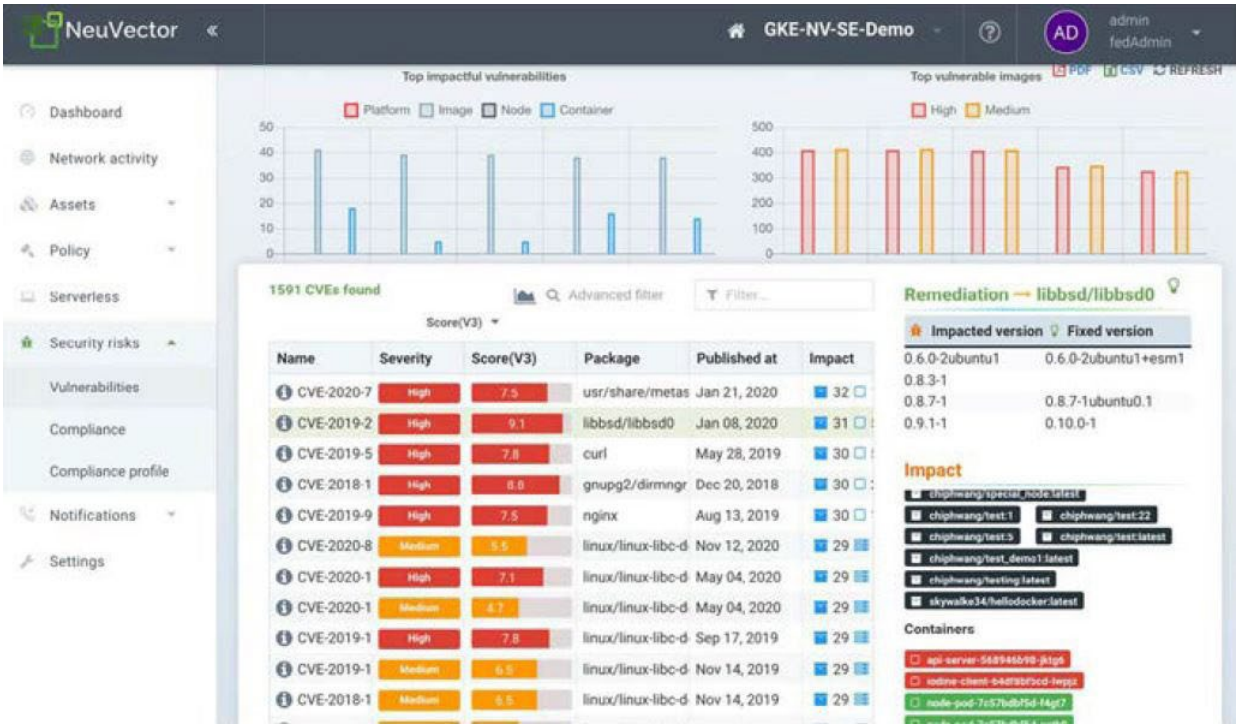


전 과정 취약성 및 컴플라이언스 관리

SUSE NeuVector는 CI/CD 파이프라인의 빌드 단계부터 원점 회귀 보안을 지원합니다. 컨테이너 이미지 빌드는 취약성 검사를 트리거하고 중요한 취약성이 있는 빌드를 중단시킬 수 있습니다. 개발자는 해당 빌드가 빌드 단계를 통과하고 승인된 레지스트리에 저장되기 전에 취약성을 수정하도록 요청 받을 수 있습니다. SUSE NeuVector는 Jenkins, CircleCI, Azure DevOps 및 Gitlab과 같은 인기 있는 파이프라인 툴을 모두 지원합니다. 사용 중인 다른 빌드 툴에도 rest API를 사용할 수 있습니다.

SUSE NeuVector는 승인된 레지스트리의 이미지를 지속적으로 검사하여 새로운 취약성을 찾습니다. 빌드 단계 또는 레지스트리에서 이미지 검사를 수행하는 동안 계층화 된 검색 결과가 표시될 뿐만 아니라 CIS 벤치마크, 탐지된 시크릿 및 파일 액세스 권한 위반에 대한 추가적인 컴플라이언스 검사가 실행됩니다.

취약성 및 컴플라이언스 탐색기는 결과를 분석하고, 컴플라이언스 보고서(PCI, HIPAA, GDPR, NIST 등)를 만들고, 취약성 수정에 대한 진행 상태를 보고하는 강력한 툴을 제공합니다.



또한 이미지 검사 결과를 승인 제어 정책과 연결하여 프로덕션 환경으로 취약한 이미지 또는 인증되지 않은 이미지가 배포되는 것을 방지할 수 있습니다.

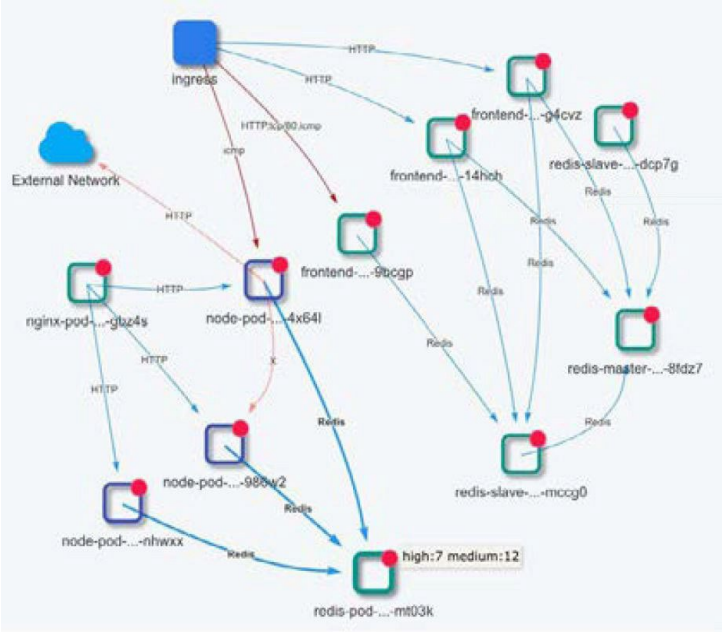
런타임 보안

SUSE NeuVector를 각 워커 노드에 배포한 후에는 컨테이너 네트워크 연결 및 서비스 종속성이 쉽게 시각화 됩니다. 쿠버네티스 배포를 격리하고 보호하기 위한 보안 정책은 자동으로 생성됩니다.

실시간으로 SUSE NeuVector 컨테이너는 네트워크 트래픽을 검사하고 컨테이너와 호스트에 의심스러운 활동이 있는지 모니터링하기 시작합니다. 다음은 쿠버네티스 배포 환경에서 SUSE NeuVector가 여러 공격 벡터에 대해 어떻게 방어하는 지에 대한 예입니다.

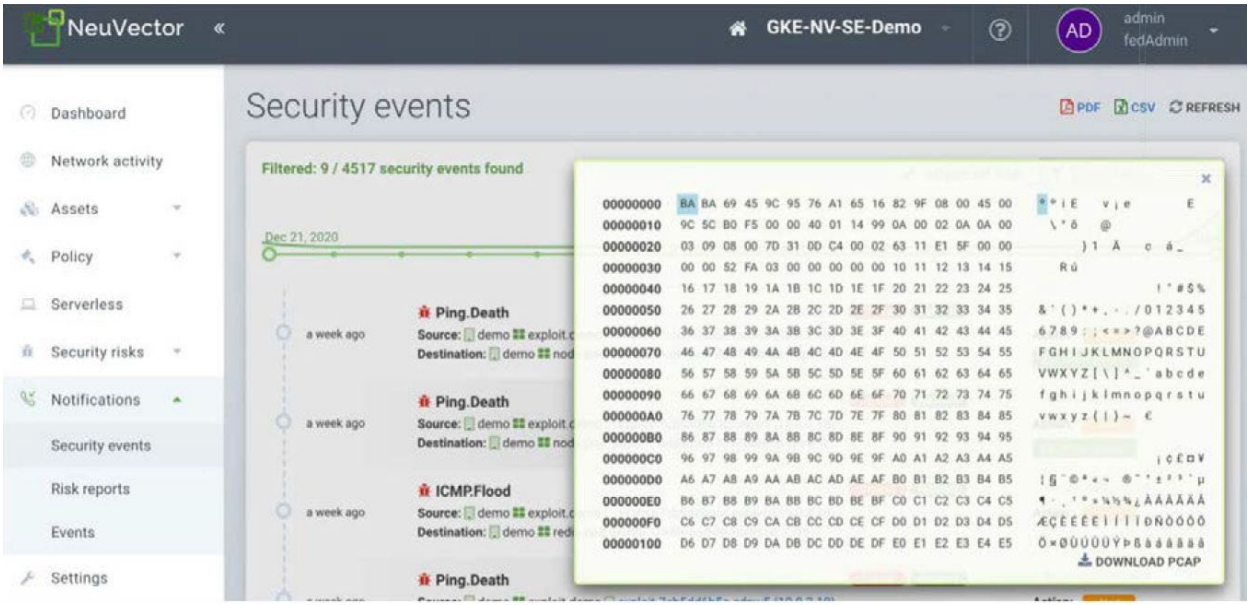
네트워크 격리, 분할 및 위협 탐지

실행 중인 포드와 네트워크 연결, 그리고 이들을 보호하기 위한 보안 정책이 자동으로 검색되고 시각화 됩니다. 각 응용 프로그램 스택은 자동으로 배포되는 화이트리스트 규칙을 통해 격리됩니다.



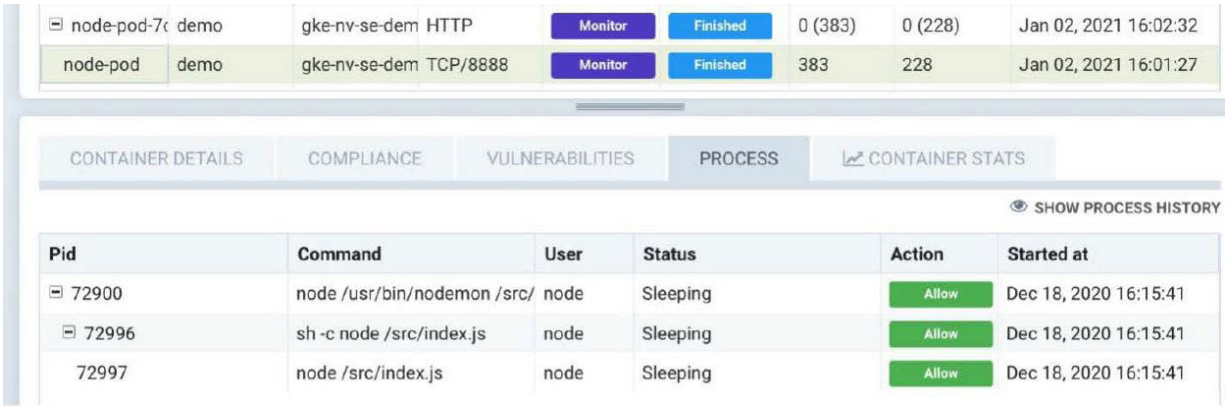
외부 또는 내부에서 발생한 컨테이너에 대한 공격을 탐지하고 차단할 수 있습니다. SUSE NeuVector Layer 7 방화벽은 모니터(네트워크 탭) 모드 또는 보호(인라인) 모드에서 실행할 수 있는데, 정상적인 트래픽에 대해 컨테이너를 활성 상태로 유지하면서 공격 또는 무단 연결을 차단할 수 있습니다. 모든 보안관련 사건은 네트워크 작업 콘솔에 요약됩니다.

패킷 캡처는 쿠버네티스 포드에 대해 자동화 및 단순화되어 포렌식, 로깅 및 애플리케이션 디버깅을 지원합니다.



컨테이너 손상 탐지

포트 검색 및 리버스 쉘과 같은 의심스러운 프로세스에 대한 기본 제공 탐지 기능을 통해 모든 컨테이너에서 비정상적인 활동을 탐지합니다. 또한 각 컨테이너에서 실행 중인 프로세스는 승인되지 않았거나 악의적인 프로세스를 탐지하는 데 도움이 되도록 기준선이 지정됩니다.



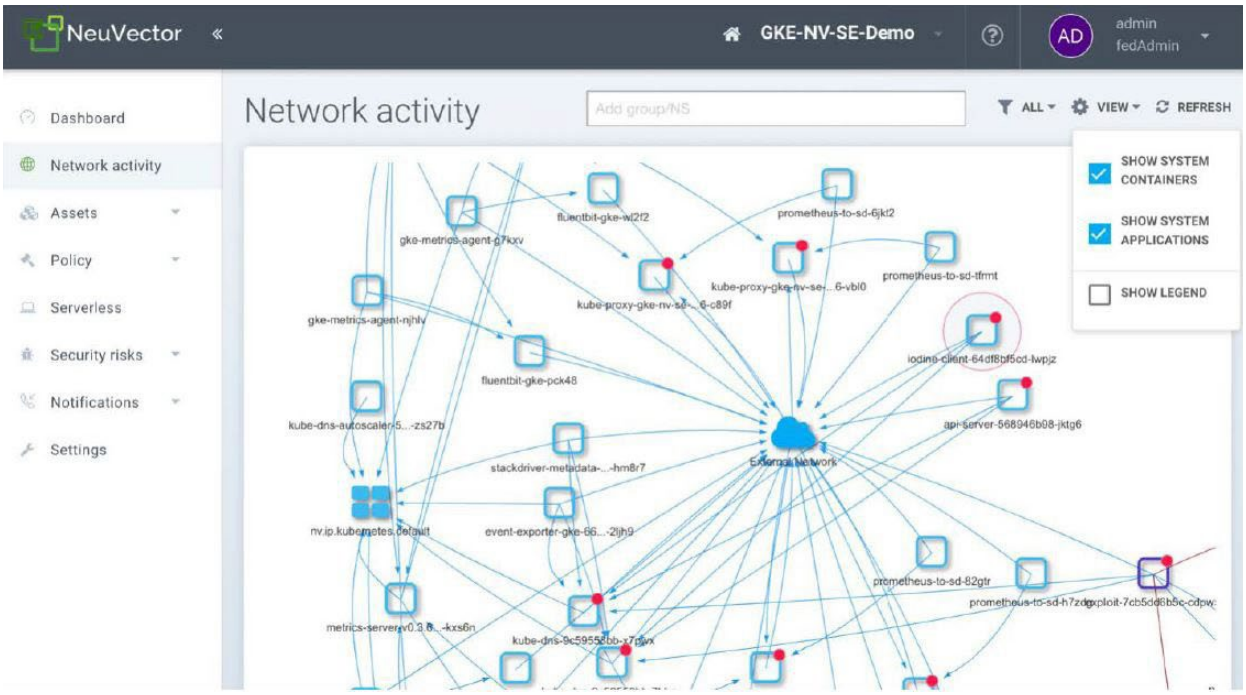
컨테이너 파일 시스템도 의심스러운 활동에 대해 모니터링됩니다. 예를 들어 패키지 또는 라이브러리가 설치 또는 업데이트되면 취약성 검사가 자동으로 트리거되고 경고가 발생됩니다.

호스트 손상 탐지

호스트 시스템은 권한 상승과 같은 공격에 대해 모니터링됩니다. 컨테이너에서 탐지된 의심스러운 프로세스도 호스트에서 실행되는 것으로 탐지됩니다. 예를 들어 포트 검사 또는 리버스 쉘 프로세스가 시작되면 SUSE NeuVector가 탐지하고 경고를 보냅니다. 또한 SUSE NeuVector는 호스트에서 실행되는 프로세스를 학습하고 화이트리스트에 포함할 수 있으며, 시작을 시도하는 인증되지 않은 호스트 프로세스를 차단할 수 있습니다.

시스템 컨테이너 모니터링

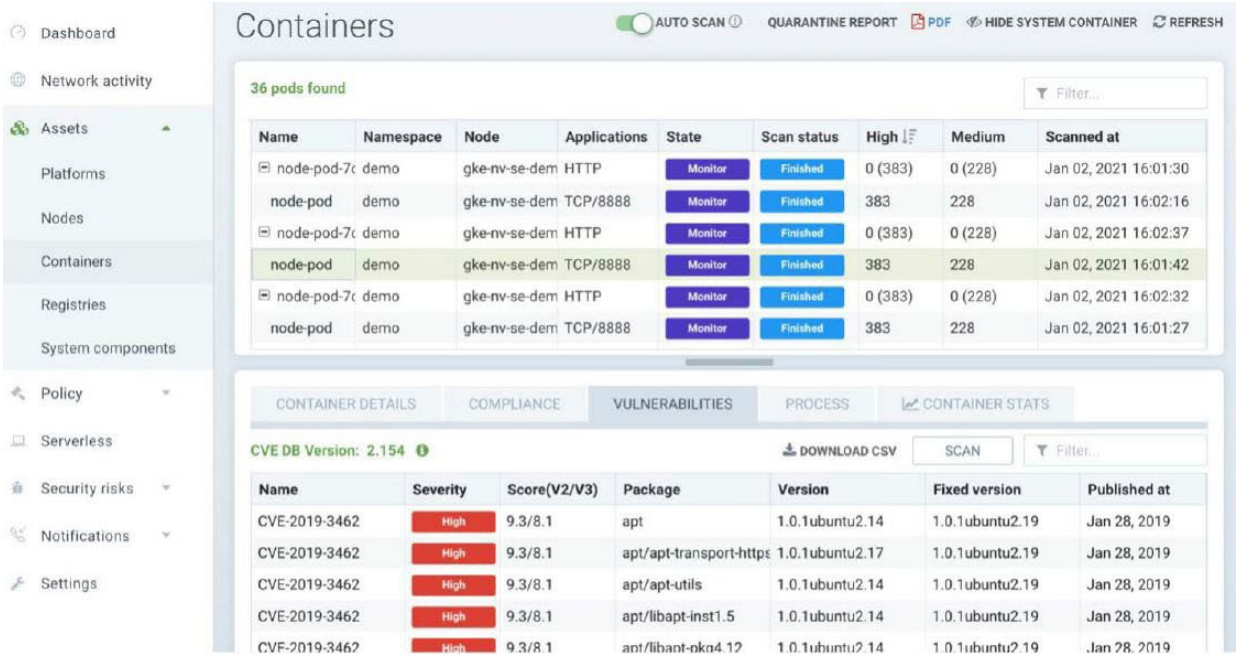
SUSE NeuVector는 또한 시스템 컨테이너와 각 컨테이너의 네트워크 활동을 모니터링합니다. 아래 다이어그램에서 쿠버네티스 및 OpenShift 컨테이너는 네트워크 연결과 함께 표시됩니다.



시스템 컨테이너로 오가는 의심스러운 활동은 쉽게 탐지될 수 있습니다.

런타임 검사 - 컴플라이언스 및 감사

SUSE NeuVector는 실행 중인 포드, 컨테이너 및 워커 노드에 대한 취약성을 자동으로 검사하고 모든 노드에서 쿠버네티스 CIS 벤치마크 테스트를 실행합니다. 시스템 컨테이너 및 오케스트레이션 플랫폼(예: 쿠버네티스 1.19)에서도 취약성을 검사합니다.



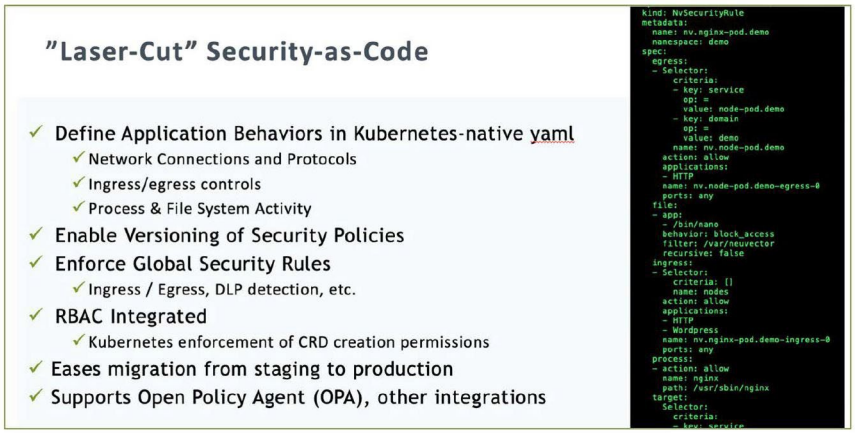
SUSE NeuVector는 빌드 및 출하 단계에서 이미지 레지스트리를 검사를 위해 혹은 CI/CD 자동 파이프라인 중에 취약성 검사를 실시할 수 있습니다. Jenkins와의 통합을 통해 이미지 빌드 프로세스 중에 검사가 가능합니다.

쿠버네티스 보안 자동화 – 가능할까요?

개발팀이 컨테이너와 쿠버네티스를 사용하여 애플리케이션 구축을 자동화하기 위해 서두르고 있는 상황에서 보안 자동화는 매우 중요합니다. 보안 팀이 애플리케이션, 인프라 또는 새로운 클라우드의 배포나 업데이트를 중단하거나 지연시킬 수 있는 시대는 지났습니다.

보안 자동화는 컨테이너를 실행을 위한 보안 인프라 및 플랫폼을 구성하는 것부터 시작하여 자동화된 런타임 보안까지 포함합니다. 반복 가능한 보안 구성을 가진 보안 인프라는 코드 개념 및 Terraform과 같은 툴로 사용함으로써 구현될 수 있습니다.

대부분의 런타임 보안이 행동 학습과 보안을 코드로 구현하기 위한 사용자 지정 리소스 정의(CRDs)와 쿠버네티스 통합의 조합으로 자동화될 수 있습니다. 초기 수동 설정 또는 사용자 지정이 필요할 수 있지만 프로덕션이 가동되고 쿠버네티스가 포드 관리를 시작하면 보안은 배포와 함께 자동화, 조정 및 확장될 것입니다.



SUSE NeuVector와 같은 새로운 보안 툴은 쿠버네티스 배포 모델에 맞춰 다중 벡터 런타임 보안을 제공할 수 있습니다. 쿠버네티스 컨테이너 방화벽을 컨테이너 검사 및 호스트 보안과 결합함으로써, 치명적인 공격을 실행하려는 킬 체인에서의 활동을 탐지하고 방지할 수 있습니다. SUSE NeuVector의 최신 클라우드 네이티브 아키텍처는 자체 컨테이너로 애플리케이션 컨테이너 바로 옆에 쉽게 배포되어 언제나 작동상태인 상시 보안을 제공합니다.

컨테이너 기반 애플리케이션의 선언적 특성과 쿠버네티스와 같은 툴을 위한 광범위한 통합 환경으로 인해 하이퍼 다이내믹 컨테이너 환경에서도 고급 보안 제어를 적용할 수 있습니다. 쿠버네티스와 통합하고 SUSE NeuVector와 같은 행동 학습 및 다중 벡터 보안 기능을 채택함으로써, 파이프라인 전반에 걸쳐 보안 자동화가 가능하게 되었으며, 보안 자동화는 이제 비즈니스 크리티컬 쿠버네티스 배포를 위해 꼭 필요한 요구사항이 되었습니다.

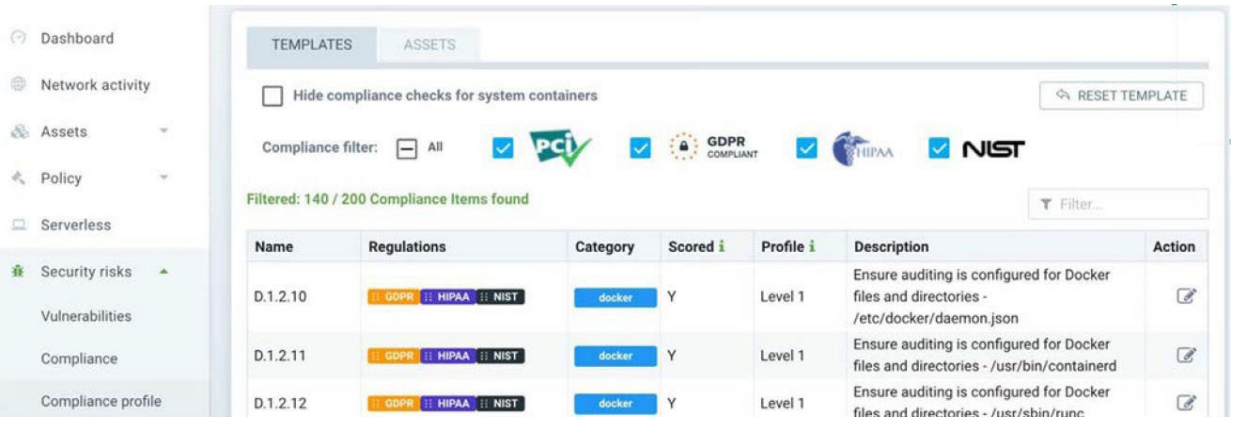
PCI, GDPR, SOC 2, HIPAA, NIST 컴플라이언스 실현

컨테이너 인프라에서는 업계 표준 컴플라이언스를 준수하는 것이 어려울 수 있습니다. 새로운 가상화 계층은 감사 및 컴플라이언스 컨설턴트에 의해 면밀히 확인되지 않았으며 컴플라이언스 요구 사항에 대한 일치된 의견도 검토되지 않았습니다.

SUSE NeuVector는 특정 보안 방식과 기능에 대해 PCI, GDPR, SOC 2, HIPAA, NIST 와 같은 표준에 맞게 컴플라이언스를 실현할 수 있도록 지원합니다. 이러한 표준들은 다음과 같은 보안 준수를 요구할 수 있습니다.

- **네트워크 분할 및 방화벽.** 컨테이너 및 쿠버네티스 네트워크 필터링 및 보호를 위해 설계된 Layer 7 컨테이너 방화벽을 통해 SUSE NeuVector는 이러한 요구 사항을 충족할 수 있는 독보적인 위치에 있습니다.
- **취약성 검사 및 업데이트 적용.** 전 단계에 걸친 취약성 관리를 통해 SUSE NeuVector는 빌드 단계부터 프로덕션까지 취약성 관리 정책을 적용할 수 있습니다.
- **구성 테스트 감사.** 시스템(호스트), Orchestrator 및 컨테이너의 구성을 검토하고 적절한 구성을 하며, 쿠버네티스 CIS 벤치마크와 같은 컴플라이언스 검사를 통해 잘못된 구성으로 인해 컴플라이언스 위반이 발생할 위험을 줄여 줍니다.
- **제한된 액세스 제어.** 최소 요구 사용자 액세스 권한을 평가하고 부여하여 RBAC 기반 공격 및 내부자 공격의 가능성을 제한 합니다.
- **암호화 및 중요한 데이터 보호.** SUSE NeuVector는 이동 중인 데이터에 대한 암호화 연결을 확인할 수 있으며 DLP 기술을 사용하여 사회보장 번호, 신용 카드 및 기타 PII와 같은 민감한 데이터 누출을 모니터링할 수도 있습니다.

SUSE NeuVector는 컨테이너 배포를 위해 PCI, GDPR, HIPAA 및 NIST 컴플라이언스 보고서에 대해 사전 구성된 사용자 지정 가능한 보고서를 제공합니다.



SUSE NeuVector는 사용자 정의 가능한 컴플라이언스 보고서, 전 단계에 걸친 취약성 관리, 방화벽 및 네트워크 분할 및 컴플라이언스 테스트를 결합하여 기업이 새로운 클라우드 네이티브 인프라 및 워크로드에 대한 컴플라이언스를 달성할 수 있도록 지원해 왔습니다. 이는 PCI 및 GDPR을 넘어 SOC(서비스 조직 제어) 2와 같은 표준까지 확장됩니다.

SOC 2는 서비스 공급자(애플리케이션)가 데이터를 안전하게 관리하여 조직의 이익과 고객의 개인 정보를 보호하는 감사 절차입니다.

보안에 민감한 비즈니스가 SaaS 공급업체를 고려할 때, SOC 2 컴플라이언스는 최소한의 요구 사항입니다. SOC 2 컴플라이언스는 클라우드에 클라이언트 정보를 저장하는 모든 관계된 기술 기반 서비스 조직에 필수적입니다. 이러한 비즈니스에는 SaaS 및 기타 클라우드 서비스를 제공하는 동시에 고객 정보를 클라우드에 저장하는 것을 포함합니다.

SUSE NeuVector는 SOC 2의 모든 주요 요구사항을 해결하여 조직이 SOC 2 표준 이상의 컴플라이언스를 달성할 수 있도록 해줍니다. 보안을 유지하고 컴플라이언스를 달성하려면 SUSE NeuVector에서 제공하는 추가적인 보호가 필요합니다. 금융 서비스 회사와 기타 규제가 심한 업종의 기업들은 고객 정보를 안전하게 유지하고 비즈니스를 언제든 감사하는데 필요한 쿠버네티스 데이터 손실 방지 기능을 위해 SUSE NeuVector를 채택하고 있습니다.

오픈 소스 Kubernetes 보안 툴

SUSE NeuVector 컨테이너 보안 플랫폼과 같은 상용 툴은 전 단계에 걸친 보호와 가시성을 제공하지만, 쿠버네티스 보안 기능을 추가하며 지속적으로 발전하는 오픈 소스 프로젝트들도 있습니다. 다음은 프로덕션에서 비즈니스 크리티컬하지 않은 프로젝트에 고려해 볼 수 있는 몇 가지 툴들입니다.

- **네트워크 정책.** 쿠버네티스 보안 정책은 자동화된 L3/L4(IP 주소/포트 기반) 분할을 제공합니다. Calico와 같은 네트워크 정책 시행을 지원하는 네트워크 플러그인이 필요합니다.
- **서비스 메시/ISTIO.** Istio는 라우팅, 인증, 권한 부여 및 암호화를 포함한 서비스 대 서비스 통신을 관리하기 위한 서비스 메시의 예입니다. Istio는 서비스 라우팅을 관리하기 위한 견고한 프레임워크를 제공하지만 공격, 위협 및 의심스러운 컨테이너 이벤트를 탐지하는 보안 툴로 설계되지는 않았습니다.
- **Grafeas.** Grafeas는 현대 소프트웨어 공급망을 감사하고 관리하기 위한 통일된 방법을 정의하는 툴을 제공합니다. 타사 툴과 통합하여 정책을 추적하고 적용할 수 있습니다. Grafeas는 CI/CD 파이프라인을 관리하는 데 유용할 수 있지만 런타임 보안

정책을 관리하기 위한 툴은 아닙니다.

- **Clair.** Clair는 이미지의 취약성 검사를 위한 간단한 툴로서 레지스트리 통합 및 워크플로우 지원이 부족합니다.
- **쿠버네티스 CIS 벤치마크.** CIS 쿠버네티스 보안 벤치마크는 100개 이상의 컴플라이언스 및 감사 검사를 제공합니다. 이러한 테스트에 대한 SUSE NeuVector의 구현을 여기에서 확인할 수 있습니다.
- **오픈 정책 에이전트(OPA).** OPA는 보안 정책을 관리하고 시행하기 위한 프레임워크를 제공합니다. 기업 전체에 걸쳐 서로 다른 보안 정책을 관리하기 위해 특수 쿼리 언어가 지원되며, 쿠버네티스 승인 제어를 통해 제한적으로 시행됩니다.

SUSE NeuVector는 이러한 오픈 소스 프로젝트와 호환되는 상용 컨테이너 보안 솔루션으로, 재무, 컴플라이언스 규제, 그리고 기타 비즈니스 크리티컬 컨테이너 배포를 보호하도록 설계된 고급 보안 기능을 제공합니다.

런타임 쿠버네티스 보안을 위한 간단한 요약 체크리스트

다음은 CI/CD 파이프라인과 런타임 중에 쿠버네티스 배포를 보호하기 위해 검토해야할 간단한 보안 체크 리스트입니다.

CI/CD 파이프라인

- 빌드 단계에서 이미지를 검사하여 수정 가능한 중요한 취약성이나 컴플라이언스 위반이 있는 이미지를 중단/거부합니다.
- 레지스트리에서 승인된 이미지를 지속적으로 검사하여 새 취약성이 발견되면 경고를 발생시킵니다.
- 내장된 시크릿, 루트, 파일 권한 및 기타 보안 문제뿐만 아니라 CIS 벤치마크에 의한 컴플라이언스 위반이 있는지 이미지를 검사합니다.
- 선언적 보안을 코드 프랙티스로 구현하여 개발자 및/또는 개발팀이 애플리케이션 워크로드에 대해 허용된 (화이트리스트) 애플리케이션 행동을 생성하거나 검토하게 합니다.

개발 및 보안 팀을 위한 프로덕션 이전 단계 체크리스트

- 네임스페이스를 사용
- Linux 기능을 제한
- SELinux를 사용
- Seccomp를 활용
- Cgroups 구성
- R/O 마운트 사용
- 최소 호스트 OS 사용
- 시스템 패치 업데이트
- CIS 벤치마크 테스트를 통한 보안 감사 및 컴플라이언스 검사 수행

운영 및 보안 팀의 런타임 체크리스트

- 승인 제어 정책을 사용하여 무단 배포 및 취약한 배포 방지
- 네트워크 분할 및 네임스페이스를 사용하여 애플리케이션/서비스별로 격리
- 응용 프로그램 공격이 있는지 네트워크 연결 검사
- 컨테이너에서 의심스러운 프로세스 또는 파일 시스템 활동 모니터링
- 호스트 권한 상승, 의심스러운 프로세스 또는 파일 시스템 활동으로부터 워커 노드 보호
- 보안 이벤트에 대한 패킷 캡처
- 손상된 컨테이너 검역이나 교정
- 컨테이너 및 호스트의 취약성 검사
- 보안 사고에 대한 실시간 경고, 기록과 대응
- CIS 벤치마크를 통한 보안 감사 및 컴플라이언스 검사 수행
-

쿠버네티스 시스템 보호

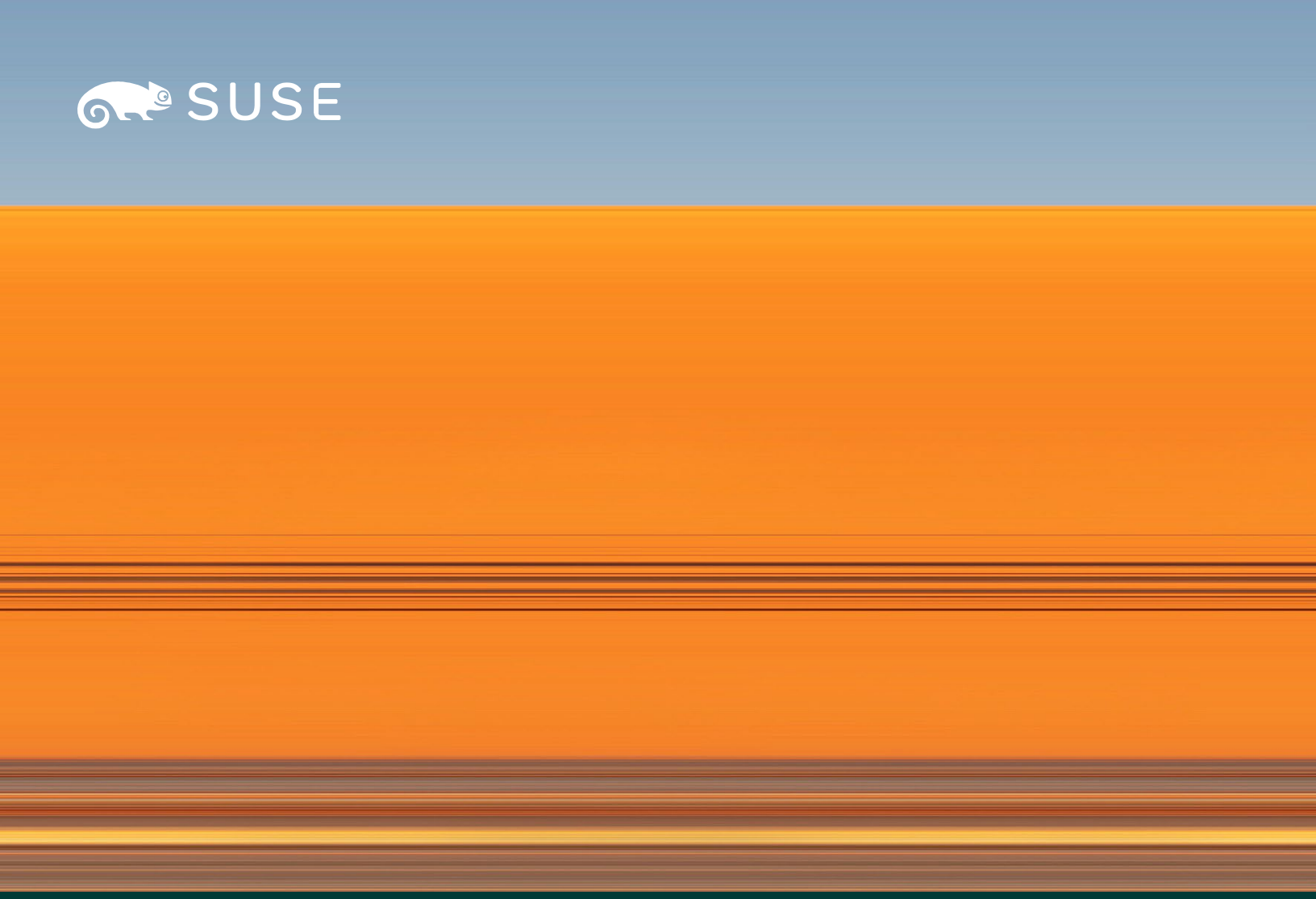
- 모든 RBACs 검토
- API 서버 보호

- Kubelet 권한 제한
- 외부 포트 보호
- 인증되지 않은 서비스의 화이트리스트 리소스 엑세스 제한
- 콘솔 액세스 제한
- 프로덕션 시스템 컨테이너 연결과 프로세스 모니터링
- 구성에 대한 감사를 위해 CIS 벤치마크 실행
- Orchestrator 버전을 업데이트하여 중요한 취약성 교정

다음 단계

더 많은 정보를 얻기를 원하십니까?

당사 블로그에 있는 추가 컨테이너 보안 기사를 보거나 SUSE NeuVector 쿠버네티스 보안 플랫폼의 데모를 예약하려면 NeuVector.com을 방문하십시오.



SUSE 코리아

서울특별시 강남구 테헤란로26길 14 대세빌딩 13F

www.suse.com/korea

더 많은 정보를 원하시면 다음으로 연락주십시오:

0030 8491 0147 (local toll-free)

sales-inquiries-APAC@suse.com

Innovate Everywhere