Integrazione della sicurezza nelle tre fasi di deployment dei container



L’adozione dei container ha avuto una crescita esponenziale negli ultimi anni e l’importanza della messa in sicurezza delle applicazioni create dai team DevOps e distribuite nei container è cresciuta di pari passo. Le misure di protezione configurate dai responsabili della sicurezza devono coprire l’intero ciclo di vita del container ed integrarsi nella pipeline DevOps in modo efficace e discreto. Riuscirci richiede conoscenza della tecnologia di containerizzazione Docker e l’adozione di strumenti e processi ideati su misura per questi ambienti.

Le sfide sulla sicurezza dei container

I container introducono problemi di sicurezza e compliance particolari, la maggior parte dei quali sono originati dalle caratteristiche che gli sviluppatori trovano così attraenti, come la possibilità di creare un pacchetto dell’applicazione e delle sue dipendenze in assenza di un sistema operativo ospite. Tra questi problemi annoveriamo l’uso di software non verificato proveniente da repository pubblici, che contiene spesso vulnerabilità irrisolte, e il deployment di container con configurazioni insicure o a valori predefiniti.

I container, inoltre, comunicano tra loro direttamente attraverso porte di rete esposte; sottraendosi ai controlli esercitati dall’host e restando quindi difficili da monitorare a causa della loro natura effimera.

Obiettivi di sicurezza per i container

Sono quattro gli obiettivi di sicurezza che è necessario perseguire per i container.

Innanzi tutto, occorre ottenere visibilità sull'intero ambiente del container identificando e tracciando tutti i deployment esistenti in locale e nel cloud.

In secondo luogo, è necessario gestire le vulnerabilità del container e contemporaneamente prevenire e rilevare le intrusioni. In terzo luogo, bisogna sfruttare le API REST per integrare i prodotti DevOps e gli strumenti di messa in sicurezza dei container.

Infine, va ripensata la funzionalità di risposta agli incidenti del programma di sicurezza del container, specialmente perché i container vulnerabili non vengono "rattoppati" ma sostituiti con nuove unità create con un'immagine aggiornata.

I team di sicurezza devono aggiornare il programma di risposta agli incidenti, dotarlo della capacità di raccogliere informazioni sul nuovo sistema operativo dei container e sulla piattaforma di orchestrazione, al fine di pianificare l'introduzione del modello "sostituzione completa" nel programma di risposta agli incidenti.

Protezione della pipeline del container

Devono essere messe in sicurezza tutte e tre le fasi del deployment del container, costruzione, implementazione e runtime, dove ognuna presenta particolarità e requisiti diversi. Approfondiamo ogni fase.

Costruzione

In questa prima fase, l'obiettivo principale consiste nel bloccare l'ingresso di immagini vulnerabili nei repository

dell’organizzazione. Questa misura è particolarmente importante nel caso in cui le immagini non vengano create da zero ma prelevate in parte o in toto da un repository pubblico, pratica assai comune tra gli sviluppatori di container.

Per bloccare l’ingresso di immagini pericolose negli archivi Docker, è fondamentale sfruttare le API REST o i plug-in nativi perché consentono l’esecuzione in automatico dei controlli di sicurezza direttamente negli strumenti CI/CD scelti dal team DevOps.

Occorre dotarsi della flessibilità di definire i parametri per bloccare le immagini non sicure, ad esempio la presenza di vulnerabilità con un punteggio di gravità di 3, 4 o 5, di rilevare il mancato rispetto dei requisiti di compliance e di individuare l’utilizzo aperto di secrets nelle immagini.

Grazie a queste integrazioni, i responsabili della sicurezza potranno concedere agli sviluppatori l’accesso agli strumenti di sicurezza, così che possano svolgere preventivamente alcune attività di sicurezza direttamente nel programma CI/CD senza delegarle al gruppo di sicurezza. Gli sviluppatori potranno accedere, ad esempio, ai risultati delle scansioni e operare interventi di remediation direttamente dall’interfaccia del loro programma CI/CD.

Implementazione

In questa fase è essenziale assicurarsi che le immagini già presenti nei repository siano prive di vulnerabilità. Fare l’inventario di registri e repository e, man mano che le immagini vengono aggiunte, avviare analisi di ricerca delle vulnerabilità. È buona norma, inoltre, pianificare scansioni quotidiane automatiche per cercare nuove eventuali vulnerabilità e controllare sistematicamente tutte le immagini che ogni giorno vengono aggiunte ai repository.

Accertare anche l’affidabilità e la rispettabilità delle fonti

da cui la propria organizzazione preleva le immagini e assicurarsi che le immagini siano aggiornate e prive di vulnerabilità conosciute. Affidarsi a servizi di certificazione della validità, come Docker Notary, per firmare le immagini e assicurarsi che il proprio ambiente ospiti solo immagini affidabili.

Runtime

Ora che le immagini dei container sono nel registro dell'azienda pronte per la produzione, è necessario disporre di visibilità costante e capacità di monitoraggio continue sugli ambienti runtime, oltre che della capacità di impedire le violazioni e di reagire.

I team di sicurezza devono essere in grado di identificare i container non autorizzati o vulnerabili, di individuarne la posizione e di valutarne il potenziale impatto in funzione del loro livello di impiego nell'ambiente.

Un accorgimento utile in questo scenario consiste nel contrassegnare i container già attivi nel sistema che evidenziano deviazioni rispetto al comportamento "immutabile" dell'immagine di partenza. Tali deviazioni potrebbero, infatti, indicare una falla.

Dal momento che i container sono uguali a un'immagine, è essenziale definire il comportamento predefinito dell'applicazione containerizzata per rilevare qualsiasi deviazione sospetta, cioè attività impreviste come chiamate al sistema, processi e comunicazioni.

I team di sicurezza devono determinare dove sono memorizzate tali immagini nell'ambiente runtime e individuare sia le versioni attive che quelle latenti.

Una volta individuati i container compromessi, il programma di sicurezza dovrà consentire all'utente di applicare contromisure, come il blocco dell'esecuzione del container o

la messa in quarantena, e di analizzare in modo approfondito i dettagli delle anomalie per determinare l'origine del problema. Generalmente, questa attività non interferisce con il normale funzionamento del sistema perché sarà possibile sostituire l'unità compromessa attivando un nuovo container dalla stessa immagine di partenza.

Una strategia migliore sarebbe usare uno strumento che convalidi automaticamente la congruenza dell'immagine alle policy di sicurezza e blocchi la generazione di nuovi container usando immagini non autorizzate. Qui entrano in gioco i servizi di orchestrazione, come Kubernetes, che impediscono ai container compromessi di superare i controller di ammissione ed entrare nell'ambiente.

Per creare ambienti operativi basati sul modello "di accesso minimo", occorre inoltre attenersi alle pratiche di sicurezza prescritte per gli ambienti di orchestrazione e, se esiste un canale di accesso all'host sottostante, rafforzarlo mettendo in atto un controllo delle vulnerabilità e definendo requisiti di compliance.

Sintesi

Ci auguriamo che questo articolo abbia gettato un po' di luce sui particolari problemi di sicurezza che riguardano i container e come affrontarli nel corso del loro intero ciclo di vita: bloccare l'ingresso di immagini vulnerabili nei repository durante la fase di costruzione; mantenere sicure le immagini inviate nei registri durante la fase di implementazione ed eseguire analisi delle immagini pronte per la produzione nella fase di runtime per individuare e gestire i container compromessi.

Risorse aggiuntive

- Per maggiori informazioni potete guardare su richiesta il nostro webcast

“Building Security into 3 phases of Container Deployment”

- Seguite ci su Twitter e Linkedin
- Per maggiori informazioni su Qualys: www.qualys.com

A cura di: **Hari Srinivasan**

L’articolo Integrazione della sicurezza nelle tre fasi di deployment dei container proviene da ICT Security Magazine.

https://www.ictsecuritymagazine.com/notizie/integrazione-della-sicurezza-nelle-tre-fasi-di-deployment-dei-container/