Sei procedure DevSecOps consigliate per gli sviluppatori

1 Riduci il rischio di dipendenza delle applicazioni

I componenti software utilizzati per creare le applicazioni potrebbero presentare vulnerabilità per cui è necessario trovare una soluzione. Gli strumenti di analisi SCA (Software Composition Analysis) possono essere impiegati per ridurre i rischi legati alla catena di distribuzione dei software, in particolare nel caso di componenti software open source. Scegli strumenti di analisi SCA che possono:

- Effettuare una scansione delle dipendenze delle applicazioni per verificare che siano prive di vulnerabilità.
- Contribuire ad automatizzare la conformità delle licenze software individuando i componenti e le relative licenze e segnalando quelle che potrebbero non essere compatibili.
- Verificare che le dipendenze delle applicazioni siano valide e che provengano da una community attiva, che rilascia ancora aggiornamenti.
- Costituire parte automatizzata del processo di creazione delle applicazioni e dell'ambiente di sviluppo. Questo aspetto offre agli sviluppatori la possibilità di risolvere i problemi prima dell'integrazione, riducendo così il numero di malfunzionamenti nella creazione delle applicazioni.

2 Unifica la gestione del codice e della configurazione

Il paradigma GitOps, ampiamente diffuso negli ambienti containerizzati e Kubernetes, prevede procedure che possono migliorare notevolmente il profilo di sicurezza, a partire dallo sviluppo:

- Adotta le procedure di sviluppo consigliate per la gestione del codice sorgente (SCM) fino alla configurazione. Utilizzare gli stessi controlli per l'avvio, l'unione e l'approvazione consente di monitorare le modifiche alla configurazione dell'infrastruttura in relazione a una persona e un momento specifici.
- Anziché affidarsi alle procedure Ops, gli sviluppatori dovrebbero iniziare a pianificare la configurazione fin dall'inizio del processo e definire l'ambiente di produzione previsto per l'applicazione. Utilizzare lo stesso tipo di ambiente e gli stessi controlli di sicurezza per le fasi di sviluppo, test e produzione semplifica la gestione della configurazione durante l'intero ciclo di vita.
- Utilizza una pipeline di creazione automatizzata per creare immagini dei container e artefatti binari per l'integrazione e la distribuzione continue (CI/CD). Non dovrebbero essere necessarie modifiche ad hoc per la distribuzione di queste immagini in produzione.
- Non archiviare dati sensibili in un sistema SCM. Utilizza gli strumenti adatti per effettuare una scansione delle immagini dei container e della configurazione per verificare che non contengano segreti.

3 Proteggi i segreti delle applicazioni

È importante gestire le identità e i segreti come le password, i token e le chiavi durante l'intero ciclo di vita delle applicazioni. Gli accessi ai sistemi SCM, ai registri dei container e ai repository binari devono essere controllati. È necessario garantire la sicurezza anche delle credenziali utilizzate dalle applicazioni per accedere a database e servizi, nonché quelle necessarie per le build automatizzate e i processi di test. I segreti possono essere divulgati per errore se archiviati in sistemi SCM o in file di configurazione. Per proteggere i segreti delle applicazioni è consigliabile:

- Definire l'infrastruttura per la gestione delle identità e per il controllo degli accessi all'inizio del ciclo di vita.
- Valutare l'impiego di un archivio protetto di segreti o un modulo di sicurezza hardware (HSM) per gestire e tutelare i segreti inattivi e in transito. Di solito gli archivi protetti di segreti sono soluzioni software, mentre gli HSM prevedono un hardware specializzato per aumentare il livello di protezione. Entrambi richiedono l'integrazione con l'infrastruttura di gestione dell'identità.

4 Usa immagini di base attendibili

Le immagini di base dei container sono distribuzioni Linux® altamente ridotte. È possibile preinstallare centinaia di pacchetti che possono contenere potenziali vulnerabilità. Per ridurre il rischio legato alle immagini dei container è possibile:

- Scegliere immagini fidate, dotate di aggiornamenti affidabili, regolari e sottoposti a test sicuri, nonché analizzare le sorgenti delle immagini e le opzioni di supporto disponibili.
- Usare strumenti per immagini per individuare le vulnerabilità note e scansionare le immagini per verificare che le configurazioni siano sicure e prive di segreti.
- Ridurre i vettori di attacco rimuovendo i file binari non necessari, tra cui gli strumenti del sistema operativo, che potrebbero essere utilizzati durante un exploit.

5 Occupati dei problemi di conformità e audit fin da subito

Per ridurre i ritardi in fase di produzione è importante comprendere i framework di conformità e i controlli tecnici richiesti nelle prime fasi dello sviluppo. È possibile inserire nella pipeline di creazione i controlli automatici per l'applicazione dei requisiti di conformità e sicurezza.

È bene iniziare a documentare in modo proattivo, perché la documentazione di politiche e procedure può costituire almeno il 50% di un audit. La documentazione delle politiche deve includere i controlli degli accessi e delle modifiche, i backup e la conservazione dei dati. I controlli di sicurezza, come i test della sicurezza delle applicazioni e l'analisi SCA, devono essere inclusi nella documentazione delle procedure.

6 Parti da una piattaforma e un ecosistema solidi

Le minacce alla sicurezza sono sempre più frequenti, perciò è di vitale importanza utilizzare una piattaforma che sia dotata di un ecosistema di sicurezza completo, con soluzioni integrate e supportate. Red Hat® OpenShift® è una piattaforma Kubernetes di livello enterprise dotata di numerose funzionalità per il supporto dello sviluppo e delle operazioni. Le potenti pipeline di sviluppo e distribuzione di Red Hat OpenShift costituiscono il punto ideale in cui implementare i controlli di sicurezza automatici. Questi possono essere inseriti in qualsiasi fase del processo, dalla creazione del codice sorgente alle immagini, fino al deployment in produzione.

Red Hat dispone di un ecosistema di partner di sicurezza che potenziano ed estendono le funzionalità di sicurezza di Red Hat OpenShift. Questi collaborano con Red Hat offrendo soluzioni supportate che si integrano con Red Hat OpenShift. È possibile scegliere tra un'ampia gamma di soluzioni per trovare quella più adatta ai propri requisiti di sicurezza e organizzazione.

Red Hat CodeReady Workspaces è un ambiente di sviluppo Kubernetes native eseguito su Red Hat OpenShift con cui accelerare lo sviluppo di applicazioni containerizzate. Red Hat Universal Base Images e Red Hat Runtimes costituiscono una base solida e di provenienza affidabile per le applicazioni.

Framework DevSecOps di Red Hat

Ottieni una panoramica del ciclo di vita della sicurezza e scopri in che modo le funzionalità di sicurezza dello sviluppo rientrano nel framework DevSecOps di Red Hat. Visita red.ht/DevSecOps.

Soluzioni per la sicurezza dello sviluppo

Guarda i webinar di Red Hat e dei suoi partner di sicurezza per scoprire come implementare la sicurezza nel ciclo di vita delle applicazioni.

Informazioni su Red Hat

Red Hat consente la standardizzazione in diversi ambienti e lo sviluppo di applicazioni cloud native, oltre a favorire l'automazione, la protezione e la gestione di ambienti complessi grazie a pluripremiati servizi di consulenza, formazione e supporto.

Italia
it.redhat.com
italy@redhat.com

Europa, Medio Oriente, e Africa (EMEA)
00800 7334 2835
it.redhat.com
europe@redhat.com

facebook.com/RedHatItaly
twitter.com/RedHatItaly
linkedin.com/company/red-hat

it.redhat.com
#F29682_0921