Trivy vs Snyk

Confronto tra due scanner di vulnerabilità per container

Giovanni Bellini

Matricola: 600035

ICT Risk Assessment
Corso di laurea magistrale in Informatica
Università di Pisa

Febbraio 2024

Indice

Capitolo 1

Introduzione

Nell'ambito della sicurezza informatica, la gestione delle vulnerabilità nei software è diventata un pilastro fondamentale per garantire l'integrità e la protezione dei sistemi informativi. Con l'avvento delle tecnologie Cloud e la diffusione di metodologie di sviluppo Agile, gli strumenti di scansione delle vulnerabilità sono diventati strumenti indispensabili per gli sviluppatori e i professionisti della sicurezza. Questo report si propone di esplorare e confrontare due dei più rilevanti strumenti nel panorama della sicurezza informatica: Trivy e Snyk. Entrambi gli strumenti hanno guadagnato notorietà per la loro capacità di fornire analisi dettagliate e soluzioni alle vulnerabilità di sicurezza in applicazioni e container, ma presentano approcci, caratteristiche e punti di forza distinti.

Trivy

Trivy[1], sviluppato da Aqua Security, è uno scanner di vulnerabilità open source semplice ma completo, che si distingue per la sua facilità d'uso e la capacità di integrarsi in vari ambienti di sviluppo e pipeline CI/CD. Le sue funzionalità includono la scansione di repository Git, filesystem, VM e configurazioni Infrastructure as Code (IaC), che rendono quindi Trivy uno strumento versatile sia per sviluppatori che per i team di sicurezza. Trivy è inoltre altamente personalizzabile, con la possibilità, ad esempio, di includere vulnerabilità ancora non pubblicate nell'analisi dei container.

Snyk

Snyk[2] si posiziona come una soluzione SaaS «chiavi in mano» per tutto ciò che concerne le vulnerabilità nello sviluppo software. Diviso in quattro componenti principali, rende possibile l'analisi delle vulnerabilità sin dai primi momenti di scrittura del codice. Inoltre, punta a semplificare la gestione delle vulnerabilità nelle dipendenze, attuando ove possibile risoluzioni automatiche. Utilizzabile sia da interfaccia Web che da linea di comando, Snyk mira a dotare anche i team più piccoli degli strumenti e dei consigli per affrontare le vulnerabilità all'interno del loro codice, delle dipendenze open source, dei container e delle configurazioni IaC.

[1] `https://github.com/aquasecurity/trivy`
[2] `https://snyk.io`

1.1 Processi di scansione

1.1.1 Trivy

I tool effettuano il processo di scansione delle vulnerabilità in container Docker in modi diversi: Una volta installato, **Trivy** utilizza il comando `trivy image` per eseguire la scansione di un'immagine Docker[1]. Questo comando esegue automaticamente tre funzioni:

1. **Download dell'immagine e del database delle vulnerabilità**: Trivy scarica automaticamente sia l'immagine da scansionare, sia l'ultima versione del database delle vulnerabilità dalla repository ufficiale. Un nuovo database viene infatti rilasciato ogni 6 ore da Aqua Security tramite GitHub Container Registry.

2. **Scansione dell'immagine**: Trivy esegue la scansione dell'immagine Docker specificata, identificando e classificando le vulnerabilità presenti nei pacchetti e nelle librerie. Nel processo di scansione, Trivy analizza i file presenti nell'immagine, identificando le versioni dei pacchetti e confrontandole con il database di vulnerabilità. Nel processo, vengono scansionate quattro categorie di problemi di sicurezza:

   - **Vulnerabilità**: problemi di sicurezza noti e documentati, che possono essere sfruttati da attaccanti per compromettere l'integrità e la disponibilità del sistema.
   - **Configurazioni errate**: errori di configurazione e di implementazione che possono esporre il sistema a rischi di sicurezza.
   - **Segreti e chiavi di accesso**: presenza di segreti e chiavi di accesso non crittografati all'interno dell'immagine, che possono essere sfruttati da attaccanti per ottenere accesso non autorizzato al sistema.
   - **Licenze:** presenza di licenze non conformi o non autorizzate all'interno dell'immagine, che possono esporre il sistema a rischi legali e di sicurezza.

   Di default, Trivy esegue la scansione di tutte e quattro le categorie, ma è possibile specificare una categoria specifica da scansionare utilizzando l'opzione `-scanners <vuln_type>`.

3. **Generazione del report**: Trivy genera un report dettagliato delle vulnerabilità rilevate, classificandole in base al loro grado di gravità e fornendo informazioni dettagliate sulle azioni consigliate per mitigarle. La gravità di una vulnerabilità è assegnata tramite due fattori principali:

   - Principalmente, si fa affidamento al livello di gravità riportato dal vendor per tale vulnerabilità.
   - In caso il vendor non riporti la categoria della vulnerabilità, Trivy fa affidamento alla gravità assegnata dal database NVD.
   - In caso anche il database NVD non riporti la categoria della vulnerabilità, Trivy la riporta come di categoria UNKNOWN.

1.1.2 Snyk

Snyk opera invece tramite un approccio differente[2]. Invece di utilizzare un unico comando per performare tutte le azioni di scansione, Snyk è suddiviso in quattro componenti principali, ognuna dedicata ad un momento differente nello sviluppo software:

- **Snyk Code:** componente relativo all'analisi statica. Connettendo il proprio sistema di gestione del codice sorgente (es. Git) a Snyk Code tramite l'apposita integrazione, il proprio progetto viene analizzato automaticamente, per identificare i possibili problemi di sicurezza presenti nel codice. Gli eventuali problemi vengono notificati all'utente tramite l'interfaccia web di Snyk, con un'evidenziazione delle linee di codice interessate e delle azioni consigliate per mitigare i problemi, come riportato in Figura 1.1.

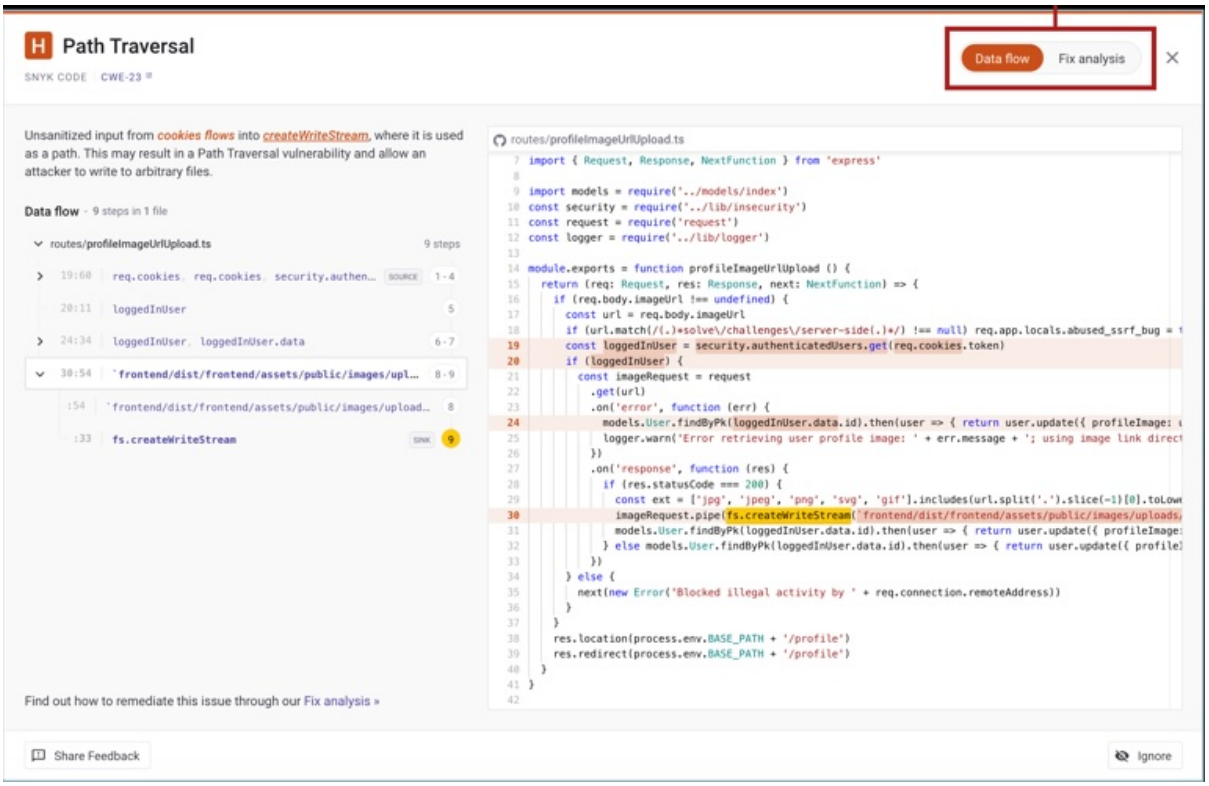


Figura 1.1: Risultato della scansione del codice sorgente con Snyk Code.

- **Snyk Container:** il vero e proprio strumento per l'analisi delle vulnerabilità nei container. Questo processo, avviabile tramite il comando `snyk test`, è suddiviso in tre fasi principali:

  1. **Download dell'immagine**: l'immagine viene automaticamente scaricata in locale per permetterne l'analisi.
  2. **Ottenimento della lista del software installato:** Snyk cerca il software installato all'interno dell'immagine. La ricerca viene effettuata in base a tre criteri:
     - Software installato tramite package manager (es. apt, yum, apk)
     - Software comunemente installato, e residente in posizioni predefinite
     - Applicazioni basate sulla presenza di un file manifest (es. package.json, requirements.txt)
  3. **Invio delle vulnerabilità a Snyk:** La lista del software trovato viene inviata tramite API a Snyk, la quale confronta i dati con il suo database di vulnerabilità. Un report contenente le vulnerabilità trovate viene quindi restituito all'utente.

  Snyk assegna inoltre ad ogni vulnerabilità un cosiddetto **priority score**, un punteggio da 0 a 1000 che indica la priorità con la quale la vulnerabilità dovrebbe essere risolta. Questo punteggio è calcolato in base a diversi fattori, tra cui la gravità della vulnerabilità,

la popolarità del pacchetto, la presenza di exploit pubblici e il numero di occorrenze della vulnerabilità all'interno del progetto. In Figura 1.2 è riportata una vulnerabilità rilevata tramite analisi con Snyk Container. È possibile notare, sulla destra, anche il Priority Score.

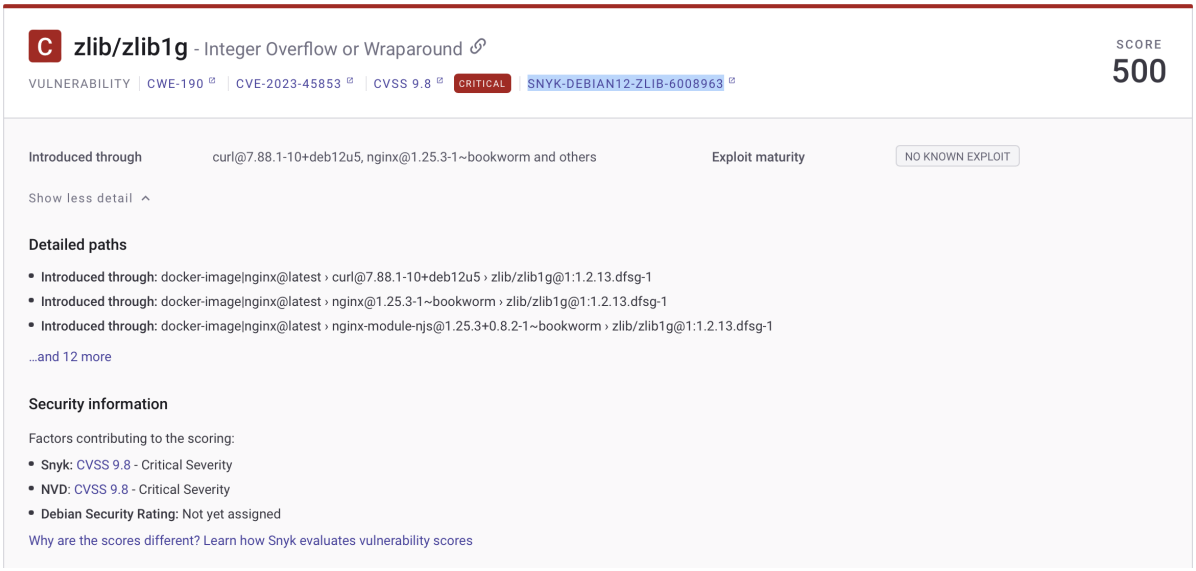


Figura 1.2: Vulnerabilità rilevata tramite l'analisi con Snyk Container.

- **Snyk Open Source:** Uno strumento di analisi delle vulnerabilità per le dipendenze open source presenti nel progetto, che identifica e classifica le vulnerabilità presenti nelle librerie e nei pacchetti utilizzati all'interno del codice sorgente. Snyk Open Source è in grado di eseguire la scansione delle dipendenze open source, identificando le vulnerabilità e fornendo informazioni dettagliate sulle azioni consigliate per mitigarle.
- **Snyk Infrastructure as Code:** Snyk IaC è uno strumento di analisi delle vulnerabilità per le configurazioni Infrastructure as Code (IaC), che identifica e classifica possibili problemi presenti nei file di configurazione di Terraform, CloudFormation e altri strumenti di automazione dell'infrastruttura. È inoltre possibile impostare Snyk per confrontare lo stato specificato su Terraform con lo stato effettivamente in uso su un cloud provider, per individuare qualsiasi possibile variazione non desiderata.

Capitolo 2

Confronto

2.1 Scansione delle vulnerabilità

Il database di entrambi gli strumenti è un aspetto cruciale per la qualità e l'accuratezza delle analisi di vulnerabilità. Trivy e Snyk utilizzano database di vulnerabilità differenti, che influenzano la loro capacità di identificare e classificare le vulnerabilità.

2.1.1 Vulnerabilità nella Supply Chain

La struttura "a strati" dei container Docker permette un'enorme flessibilità dal punto di vista dello sviluppo e della distribuzione delle applicazioni, ma introduce anche una serie di nuove sfide per la sicurezza. Le immagini Docker sono costruite a partire da altre immagini che possono contenere a loro volta una vasta gamma di pacchetti e librerie, ognuno eventualmente vulnerabile ad una serie di minacce.
Sulla base di ciò, possono quindi nascere due nuove classificazioni di vulnerabilità:

- **Vulnerabilità ereditate**: vulnerabilità presenti nella base image utilizzata per costruire l'immagine Docker.
- **Vulnerabilità installate**: vulnerabilità presenti nei pacchetti e nelle librerie installate dall'utente nell'immagine Docker in uso.

È quindi necessario categorizzare le vulnerabilità in base alla loro origine, perché il modo in cui esse devono essere affrontate è differente.

- Le vulnerabilità ereditate richiedono una modifica della base image. Per risolverle, potrebbe essere necessario aggiornare la base image, o sostituirla con una versione alternativa. Quest'ultimo processo può essere complesso e dispendioso, in quanto può richiedere la modifica della pipeline di sviluppo e la verifica della compatibilità tra l'applicazione esistente e la nuova base image.
- Le vulnerabilità installate possono essere corrette più facilmente, tipicamente aggiornando i pacchetti e le librerie installate all'interno dell'immagine. Questo processo è generalmente più semplice e meno dispendioso rispetto alla correzione delle vulnerabilità ereditate.

Il report annuale del 2023 dell'azienda di cybersicurezza Sysdig[3] riporta che l'87% delle immagini Docker in utilizzo contengono vulnerabilità di livello critico o alto. Il restante 13% contengono invece vulnerabilità di livello medio o basso, o nessuna vulnerabilità rilevata.

Questo dimostra quindi che la presenza di vulnerabilità nelle immagini Docker è un problema diffuso e attuale, e che la loro correzione è un'attività di sicurezza fondamentale.
In questa prima parte di testing, si è voluto quindi valutare la capacità di Trivy e Snyk di identificare e classificare le vulnerabilità presenti in immagini altamente utilizzate. Per fare ciò, è stato utilizzato un campione di immagini Docker, e si è confrontato il risultato delle scansioni di Trivy e Snyk. I risultati di questi test sono stati valutati in base alla precisione e alla completezza delle informazioni fornite da Trivy e Snyk.
Le immagini Docker sottoposte a scansione sono state selezionate in base alla loro popolarità e alla loro rilevanza nel panorama delle applicazioni e dei servizi cloud. In particolare, si è scelto di testare le seguenti immagini:

- **nginx**: uno dei principali server web e reverse proxy.
- **mongo**: database NoSQL flessibile e scalabile basato su MongoDB.
- **wordpress**: piattaforma di blogging e CMS.
- **alpine**: una delle immagini di container più leggere e minimali, basata su Alpine Linux È ampiamente usata come base per altre immagini di container.
- **node**: ambiente di esecuzione per JavaScript basato su Chrome V8.

Inoltre, si è voluto testare due tipi di versioni per ogni immagine: la versione più recente, e una versione più datata. Nella tabella 2.1 sono riportati i risultati ottenuti da entrambi i tool per ciascuna immagine testata.

Immagine	Vers.	Snyk				Trivy			
		Crit.	High	Mid	Low	Crit.	High	Mid	Low
nginx	latest	1	6	3	78	2	16	34	83
	v1.23.0	10	39	73	111	16	80	139	115
mongo	latest	0	0	1	12	0	0	1	21
	v4.4.3	0	7	88	69	0	13	195	105
wordpress	latest	1	1	3	150	3	50	143	326
	v6.0.0	23	66	142	228	49	340	509	547
alpine	latest	0	0	0	0	0	0	0	0
	v3.11	1	0	0	0	1	0	0	0
node	latest	1	4	3	160	5	73	246	481
	v16.0.0	48	183	275	390	140	970	1353	1417

Tabella 2.1: Risultati delle scansioni di Snyk e Trivy per ciascuna immagine testata.

2.1.2 Risultati delle scansioni

Dai risultati ottenuti, si evincono le seguenti considerazioni:

Numero di vulnerabilità rilevate

Trivy generalmente avvisa di un numero maggiore di vulnerabilità rispetto a Snyk. Questo è particolarmente evidente per le versioni più datate delle immagini, dove Trivy rileva un numero significativamente maggiore di vulnerabilità. Da solo, il mero numero di vulnerabilità rilevate non è un indicatore della qualità o dell'accuratezza delle scansioni, per via della presenza di eventuali falsi positivi o di vulnerabilità non rilevate. Dopo un'ispezione manuale a campione,

è stato però visto che la quasi totalità delle vulnerabilità era effettivamente tale, seppure la loro gravità fosse praticamente trascurabile. I rari casi si sono sempre limitati ad estremamente sporadici.

Differenza nella classificazione delle stesse vulnerabilità

Gli strumenti hanno riportato, in molti casi, una differenza nelle classificazioni delle stesse vulnerabilità. Questo è dovuto alla differenza di valutazione delle vulnerabilità da parte dei database utilizzati. In particolare, Snyk tende a classificare le vulnerabilità in modo più conservativo, assegnando un numero inferiore di vulnerabilità di livello critico o alto rispetto a Trivy. Ad esempio, per l'immagine `nginx:latest`, entrambi i tool hanno rilevato correttamente la vulnerabilità CVE-2023-6879, relativa ad un buffer overflow presente nella libreria `AOMedia`, ma mentre Trivy ha classificato la vulnerabilità come di livello critico, Snyk ha classificato la stessa come di livello basso. Questa differenza è dovuta alle sorgenti di informazioni di valutazione utilizzate dai due tool:

- Trivy utilizza principalmente la gravità assegnata dal vendor. In caso essa non sia disponibile, si induce la gravità dal punteggio CVSS. In caso anch'esso non sia disponibile, viene usata la classificazione NVD.
- Snyk invece assegna la categoria in base a tre elementi di valutazione:
  - L'analisi interna condotta dal team di Snyk.
  - Una valutazione della gravità fornita dal team di sicurezza del manutentore della distribuzione Linux.
  - La gravità della vulnerabilità secondo il database NVD.

Consultando infatti tale vulnerabilità nel sito web `security-tracker.debian.org`, è possibile osservare che la vulnerabilità CVE-2023-6879 è stata classificata come di livello basso dal team di sicurezza di Debian. Questo ha quindi influenzato in modo predominante la classificazione di Snyk.

Confermata di presenza di vulnerabilità ereditate dalla Base Image

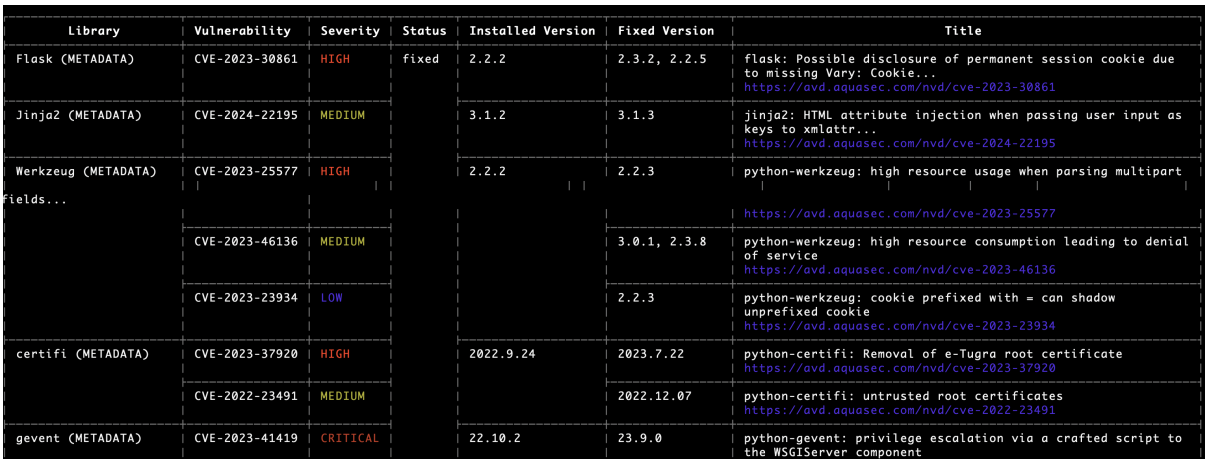
È stata inoltre confermata la presenza di vulnerabilità "ereditate" dalla Base Image: ad esempio, l'unica vulnerabilità che Snyk rileva come critica nell'ultima versione delle immagini `nginx, wordpress, node` è la stessa vulnerabilità CVE-2023-45853. Questa vulnerabilità è attualmente presente nel sistema Debian e relativa alla libreria `zlib`, ed è stata quindi ereditata dalle immagini Docker che utilizzano Debian come base image. Trivy riporta anch'esso che la vulnerabilità è critica, ma anche che essa ha uno stato `will_not_fix`. Ciò indica che tale vulnerabilità è conosciuta, ma al momento non ci sono piani per una correzione. Snyk non riporta invece tale informazione.

2.1.3 Formati di output

Entrambi gli strumenti offrono la possibilità di generare report in vari formati per permetterne, ad esempio, l'integrazione diretta con altri strumenti.

Trivy

Il formato di output predefinito di Trivy è il formato tabellare (Figura 2.1), che fornisce un report strutturato e facilmente leggibile delle vulnerabilità rilevate.



Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
Flask (METADATA)	CVE-2023-30861	HIGH	fixed	2.2.2	2.3.2, 2.2.5	flask: Possible disclosure of permanent session cookie due to missing Vary: Cookie... https://avd.aquasec.com/nvd/cve-2023-30861
Jinja2 (METADATA)	CVE-2024-22195	MEDIUM		3.1.2	3.1.3	jinja2: HTML attribute injection when passing user input as keys to xmlattr... https://avd.aquasec.com/nvd/cve-2024-22195
Werkzeug (METADATA)	CVE-2023-25577	HIGH		2.2.2	2.2.3	python-werkzeug: high resource usage when parsing multipart fields... https://avd.aquasec.com/nvd/cve-2023-25577
	CVE-2023-46136	MEDIUM			3.0.1, 2.3.8	python-werkzeug: high resource consumption leading to denial of service https://avd.aquasec.com/nvd/cve-2023-46136
	CVE-2023-23934	LOW			2.2.3	python-werkzeug: cookie prefixed with = can shadow unprefixed cookie https://avd.aquasec.com/nvd/cve-2023-23934
certifi (METADATA)	CVE-2023-37920	HIGH		2022.9.24	2023.7.22	python-certifi: Removal of e-Tugra root certificate https://avd.aquasec.com/nvd/cve-2023-37920
	CVE-2022-23491	MEDIUM			2022.12.07	python-certifi: untrusted root certificates https://avd.aquasec.com/nvd/cve-2022-23491
gevent (METADATA)	CVE-2023-41419	CRITICAL		22.10.2	23.9.0	python-gevent: privilege escalation via a crafted script to the WSGIServer component

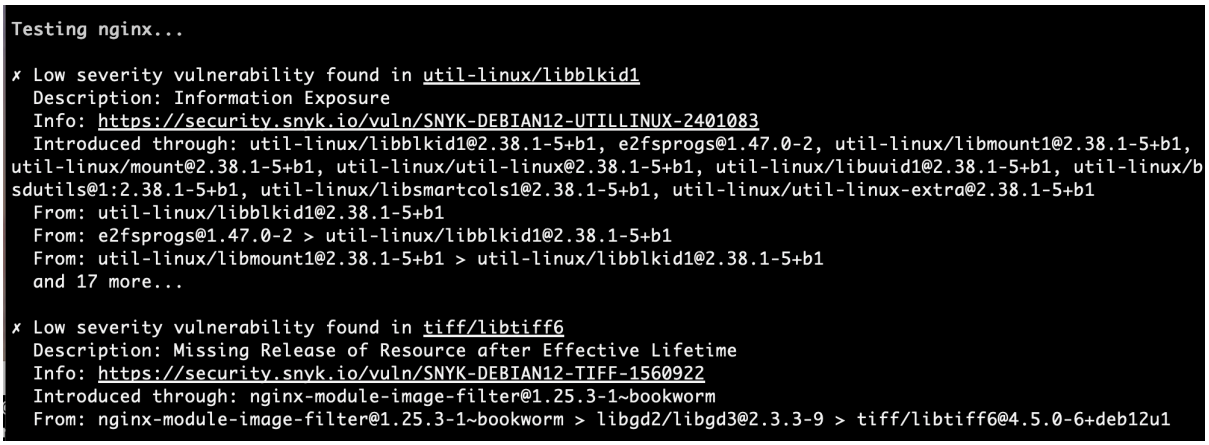
Figura 2.1: Formato di output di default di Trivy.

Inoltre, Trivy permette di generare report nei seguenti formati:

- **JSON**
- **JUnit XML**: un formato di output standard per i risultati dei test.
- **Sarif**: un formato di output standard per gli strumenti di analisi statica del codice.
- **Personalizzato**: se i formati predefiniti non soddisfano le esigenze, è possibile specificare un formato di output completamente personalizzato.

Snyk

Nella configurazione di default, Snyk restituisce un report in formato "lista" (Figura 2.2), con l'elenco di tutte le vulnerabilità rilevate.



```
Testing nginx...

✗ Low severity vulnerability found in util-linux/libblkid1
  Description: Information Exposure
  Info: https://security.snyk.io/vuln/SNYK-DEBIAN12-UTILLINUX-2401083
  Introduced through: util-linux/libblkid1@2.38.1-5+b1, e2fsprogs@1.47.0-2, util-linux/libmount1@2.38.1-5+b1,
util-linux/mount@2.38.1-5+b1, util-linux/util-linux@2.38.1-5+b1, util-linux/libuuid1@2.38.1-5+b1, util-linux/b
sdutils@1:2.38.1-5+b1, util-linux/libsmartcols1@2.38.1-5+b1, util-linux/util-linux-extra@2.38.1-5+b1
  From: util-linux/libblkid1@2.38.1-5+b1
  From: e2fsprogs@1.47.0-2 > util-linux/libblkid1@2.38.1-5+b1
  From: util-linux/libmount1@2.38.1-5+b1 > util-linux/libblkid1@2.38.1-5+b1
  and 17 more...

✗ Low severity vulnerability found in tiff/libtiff6
  Description: Missing Release of Resource after Effective Lifetime
  Info: https://security.snyk.io/vuln/SNYK-DEBIAN12-TIFF-1560922
  Introduced through: nginx-module-image-filter@1.25.3-1~bookworm
  From: nginx-module-image-filter@1.25.3-1~bookworm > libgd2/libgd3@2.3.3-9 > tiff/libtiff6@4.5.0-6+deb12u1
```

Figura 2.2: Formato di output di default di Snyk.

Inoltre, vengono mostrati anche:

- Il numero totale di vulnerabilità rilevate.

- **Consigli per la base image**: Snyk propone una base image successiva o alternativa, per notificare l'utente che aggiornando la base image alcune delle vulnerabilità rilevate siano state risolte (Figura 2.3).

```
Tested 142 dependencies for known issues, found 233 issues.

Base Image    Vulnerabilities  Severity
nginx:1.23.0  233              10 critical, 39 high, 73 medium, 111 low

Recommendations for base image upgrade:

Minor upgrades
Base Image    Vulnerabilities  Severity
nginx:1.25.3  80               1 critical, 1 high, 1 medium, 77 low

Alternative image types
Base Image                  Vulnerabilities  Severity
nginx:1.25.4-bookworm-perl  80               1 critical, 1 high, 1 medium, 77 low
nginx:stable                115              2 critical, 4 high, 4 medium, 105 low
nginx:1.24.0-perl           115              2 critical, 4 high, 4 medium, 105 low
nginx:1.25.0-bullseye-perl  144              3 critical, 13 high, 20 medium, 108 low
```

Figura 2.3: Snyk: Consigli per una base image alternativa.

Snyk permette inoltre di generare report nei seguenti formati:

- JSON
- SARIF

Differentemente da Trivy, Snyk non permette di generare report in formati tabellari o personalizzati.

2.2 Scansione di configurazioni IaC

Un'altra funzionalità offerta da entrambi i tool è relativa alla scansione di configurazioni IaC, al fine di rilevare problemi di sicurezza o cattive pratiche di configurazione. Per testare tale funzionalità, è stato redatto un campione di file di configurazione di Terraform, appropriatamente configurato con una vulnerabilità di esempio nota e ben documentata. Il codice in questione è il seguente:

```
1   provider "aws" {
2      region = "us-east-1"
3    }
4
5    resource "aws_s3_bucket" "bucket_insecure" {
6      bucket = "my-insecure-bucket"
7      acl    = "public-read"
8
9      tags = {
10       Name        = "Insecure Bucket"
11       Environment = "Test"
12     }
13   }
```

La vulnerabilità in questione è rappresentata dalla configurazione errata delle politiche di accesso su un bucket S3 di AWS, il quale è stato impostato per permettere l'accesso in lettura al pubblico di tutto il bucket (public-read). Tale configurazione espone i dati contenuti nel bucket a potenziali accessi non autorizzati, rappresentando un rischio significativo per la sicurezza dei dati. Eseguendo la scansione del codice soprastante con entrambi gli strumenti, sono state rilevate le seguenti vulnerabilità:

Strumento	Critical	High	Medium	Low
Snyk	0	0	1	3
Trivy	0	7	1	2

Tra le vulnerabilità rilevate, in entrambi i casi è stata rilevata la vulnerabilità desiderata. Su Snyk, essa è stata classificata come l'unica di livello medio. Trivy, invece, ha classificato la vulnerabilità come di livello alto. Questa differenza è dovuta alla diversa classificazione delle vulnerabilità da parte dei database utilizzati, come già discusso in precedenza. Tali vulnerabilità sono descritte nelle Figure 2.4 e 2.5. Inoltre, le altre vulnerabilità di livello alto rilevate da Trivy sono state:

- **Quattro vulnerabilità** molto simili tra loro, relative alla mancanza di ACLs sul bucket S3. (es. No public access block so not blocking public acls)
- **Bucket does not have encryption enabled:** Vulnerabilità rilevata in quanto nel bucket S3 non è stata configurata la cifratura dei dati.
- **Bucket does not encrypt data with a customer managed key:** Vulnerabilità rilevata in quanto il bucket S3 non ha una configurazione per la cifratura dei dati con una chiave gestita unicamente dal cliente.

Figura 2.4: Trivy: corretto riconoscimento della vulnerabilità public-read.

Questi errori non sono stati rilevati da Snyk, che però ha rilevato due possibili vulnerabilità di livello basso non rilevate invece da Trivy:

- **S3 bucket MFA delete control disabled:** Vulnerabilità rilevata in quanto nel bucket S3 non è stato configurato correttamente il controllo Multi-Factor Authentication per autorizzare la cancellazione dei dati.
- **S3 server access logging is disabled:** Vulnerabilità rilevata in quanto nel bucket S3 non è stato configurato il logging degli accessi al server.

```
Medium Severity Issues: 1

  [Medium] S3 Bucket is publicly readable
  Info:    That this S3 bucket is publicly readable without any authentication
           or authorization. . That you may be leaking sensitive information to
           members of the public without realizing.
  Rule:    https://security.snyk.io/rules/cloud/SNYK-CC-TF-18
  Path:    input > resource > aws_s3_bucket[bucket_insecure] > acl
  File:    main.tf
  Resolve: Set `acl` attribute to `private`, or remove the attribute

------------------------------------------------------
```

Figura 2.5: Snyk: corretto riconoscimento della vulnerabilità public-read.

2.3 Funzionalità uniche per ogni prodotto

Durante il testing, sono state rilevate inoltre delle funzionalità presenti in uno dei due strumenti, ma non nell'altro. Le funzionalità sono descritte di seguito. Essendo uniche per ogni prodotto, non è stato possibile confrontarle direttamente.

2.3.1 Snyk: Monitoraggio Continuo

Snyk offre un monitoraggio continuo delle applicazioni e delle dipendenze, inviando notifiche in tempo reale in caso di nuove vulnerabilità che influenzano il codice già in uso. Questo assicura che i team possano reagire rapidamente a nuove minacce. Durante il testing, è stato possibile osservare il funzionamento di questa funzione, avendo ricevuto una e-mail contenente nuove vulnerabilità rese pubbliche solamente pochi giorni prima, come riportato in Figura 2.6.

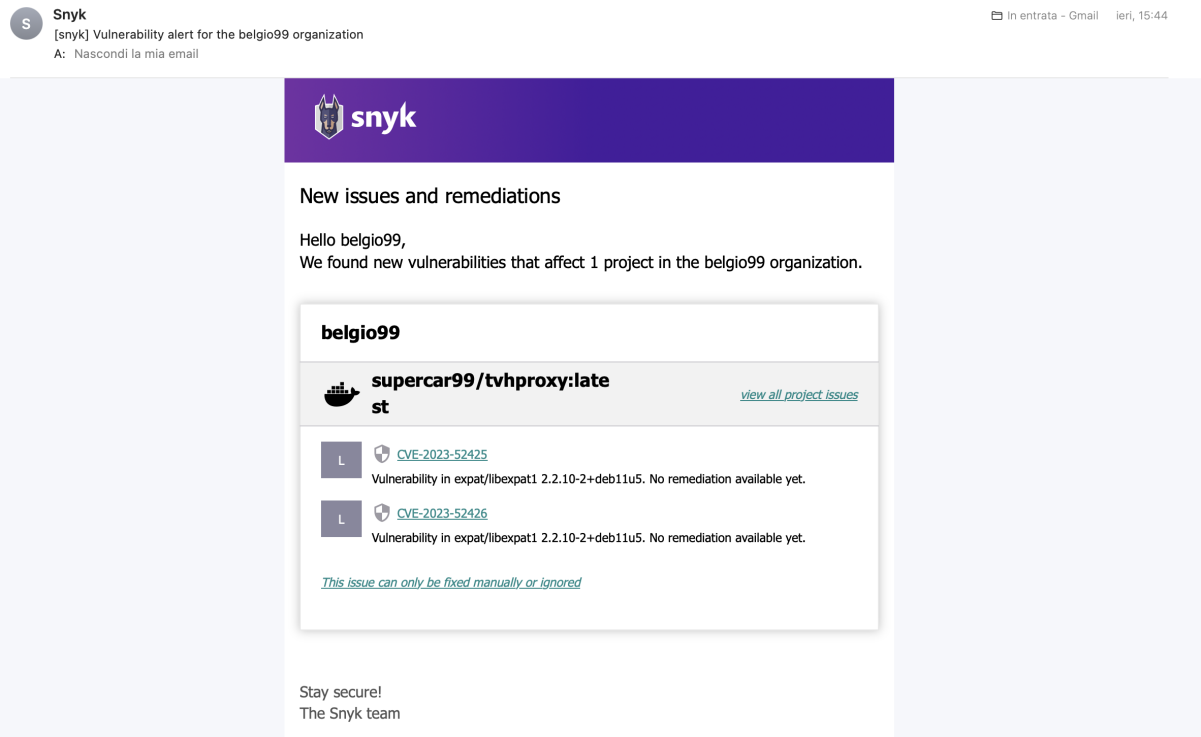


Figura 2.6: E-mail ricevuta da Snyk contenente notifica di rilevazione di nuove vulnerabilità.

Dopo la ricezione della e-mail, è stata subito eseguita una scansione con Trivy, che ha confermato la presenza delle vulnerabilità segnalate da Snyk. Questo ha permesso quindi automaticamente di verificare anche la tempestività dell'aggiornamento dei database di vulnerabilità di entrambi gli strumenti.

2.3.2 Trivy: Funzionamento Offline

Essendo un tool completamente open source, Trivy ha la capacità di poter essere utilizzato in ambienti completamente isolati, senza la necessità di connessione a Internet. Questo è possibile grazie alla possibilità di scaricare il database di vulnerabilità in locale, e di eseguire le scansioni utilizzando il database locale. Questa funzionalità è particolarmente utile in ambienti ad alta sicurezza, dove la connessione a Internet è limitata o non disponibile. Snyk, invece, richiede sempre una connessione a Internet per poter funzionare correttamente, necessitando di contattare la propria API per eseguire la vera e propria scansione.

2.3.3 Snyk: Creazione Automatica di Pull Request

Snyk offre la possibilità di creare automaticamente pull request atte a correggere le vulnerabilità rilevate. Questa funzionalità è particolarmente utile per i team di sviluppo, in quanto permette di automatizzare il processo di correzione delle vulnerabilità, riducendo il tempo e lo sforzo necessario per applicare le correzioni. In Figura 2.7 è possibile vedere un esempio di tale funzione.

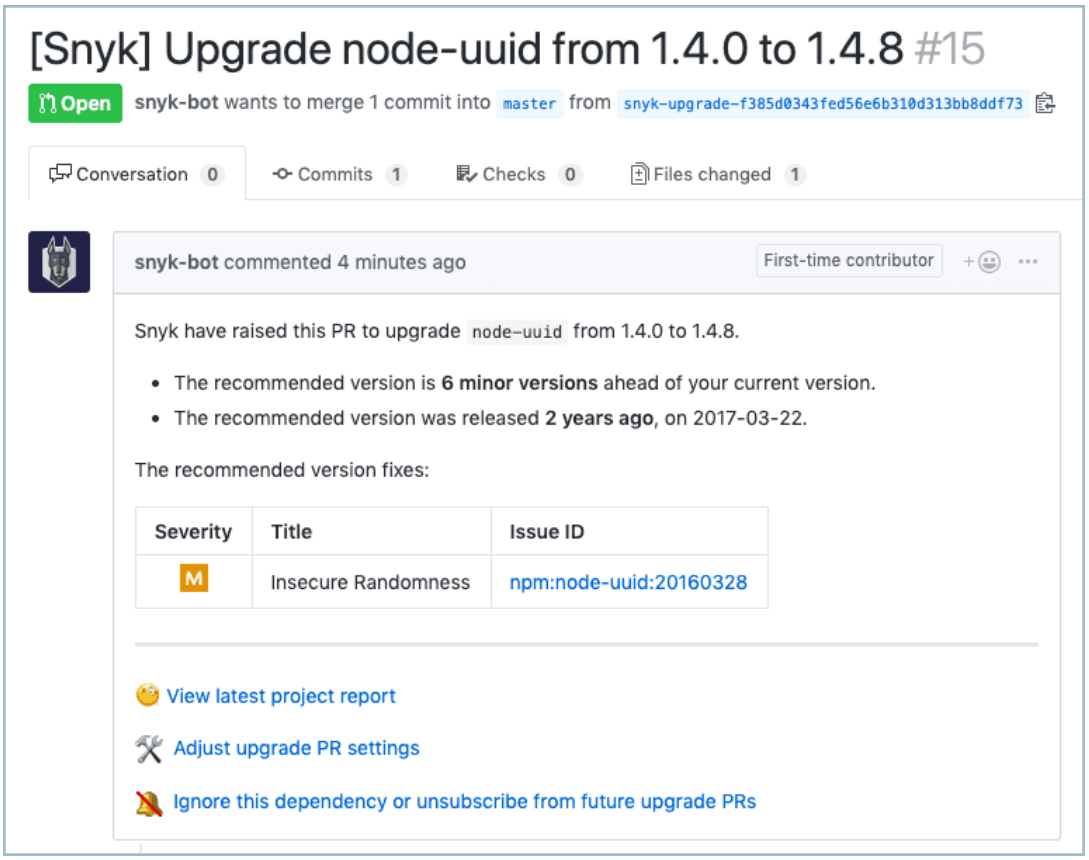


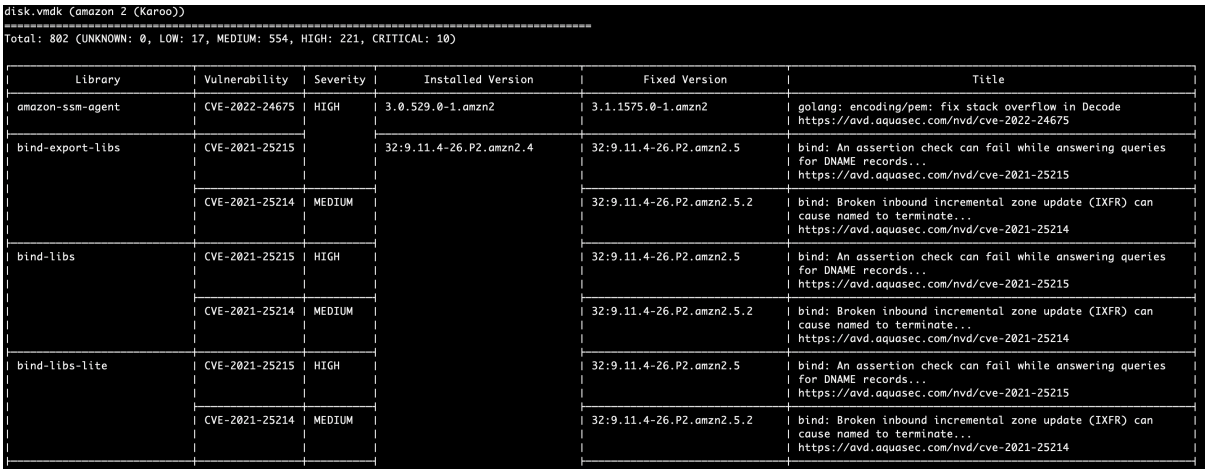
Figura 2.7: Esempio di pull request eseguita da Snyk.

2.3.4 Trivy: Scansione di Immagini VM

Un'ultima funzione unica di Trivy è quella di poter eseguire la scansione di immagini di macchine virtuali, oltre che di immagini di container. Tramite il comando `trivy vm`, è possibile scansionare:

- **File di macchine virtuali locali**: è supportata la scansione di file `.vmdk`
- **Cluster AWS EC2**: è supportata sia la scansione di AMI (Amazon Machine Image), sia eventuali immagini memorizzate come snapshot EBS (Elastic Block Storage).

In Figura 2.8, è riportato un esempio di scansione di un'immagine VM con Trivy, dove vengono rilevate varie vulnerabilità di criticità alta e media. Viene inoltre riportata in quale versione dell'immagine VM sono state corrette.



```
disk.vmdk (amazon 2 (Karoo))
============================
Total: 802 (UNKNOWN: 0, LOW: 17, MEDIUM: 554, HIGH: 221, CRITICAL: 10)

┌──────────────────────┬─────────────────┬──────────┬──────────────────────┬──────────────────────┬──────────────────────────────────────────────────────────────────────────────┐
│       Library        │  Vulnerability  │ Severity │  Installed Version   │    Fixed Version     │                                    Title                                     │
├──────────────────────┼─────────────────┼──────────┼──────────────────────┼──────────────────────┼──────────────────────────────────────────────────────────────────────────────┤
│ amazon-ssm-agent     │ CVE-2022-24675  │ HIGH     │ 3.0.529.0-1.amzn2    │ 3.1.1575.0-1.amzn2   │ golang: encoding/pem: fix stack overflow in Decode                           │
│                      │                 │          │                      │                      │ https://avd.aquasec.com/nvd/cve-2022-24675                                   │
├──────────────────────┼─────────────────┼──────────┼──────────────────────┼──────────────────────┼──────────────────────────────────────────────────────────────────────────────┤
│ bind-export-libs     │ CVE-2021-25215  │          │ 32:9.11.4-26.P2.amzn2.4 │ 32:9.11.4-26.P2.amzn2.5 │ bind: An assertion check can fail while answering queries                    │
│                      │                 │          │                      │                      │ for DNAME records...                                                         │
│                      │                 │          │                      │                      │ https://avd.aquasec.com/nvd/cve-2021-25215                                   │
│                      ├─────────────────┼──────────┤                      ├──────────────────────┼──────────────────────────────────────────────────────────────────────────────┤
│                      │ CVE-2021-25214  │ MEDIUM   │                      │ 32:9.11.4-26.P2.amzn2.5.2 │ bind: Broken inbound incremental zone update (IXFR) can                   │
│                      │                 │          │                      │                      │ cause named to terminate...                                                  │
│                      │                 │          │                      │                      │ https://avd.aquasec.com/nvd/cve-2021-25214                                   │
├──────────────────────┼─────────────────┼──────────┤                      ├──────────────────────┼──────────────────────────────────────────────────────────────────────────────┤
│ bind-libs            │ CVE-2021-25215  │ HIGH     │                      │ 32:9.11.4-26.P2.amzn2.5 │ bind: An assertion check can fail while answering queries                    │
│                      │                 │          │                      │                      │ for DNAME records...                                                         │
│                      │                 │          │                      │                      │ https://avd.aquasec.com/nvd/cve-2021-25215                                   │
│                      ├─────────────────┼──────────┤                      ├──────────────────────┼──────────────────────────────────────────────────────────────────────────────┤
│                      │ CVE-2021-25214  │ MEDIUM   │                      │ 32:9.11.4-26.P2.amzn2.5.2 │ bind: Broken inbound incremental zone update (IXFR) can                   │
│                      │                 │          │                      │                      │ cause named to terminate...                                                  │
│                      │                 │          │                      │                      │ https://avd.aquasec.com/nvd/cve-2021-25214                                   │
├──────────────────────┼─────────────────┼──────────┤                      ├──────────────────────┼──────────────────────────────────────────────────────────────────────────────┤
│ bind-libs-lite       │ CVE-2021-25215  │ HIGH     │                      │ 32:9.11.4-26.P2.amzn2.5 │ bind: An assertion check can fail while answering queries                    │
│                      │                 │          │                      │                      │ for DNAME records...                                                         │
│                      │                 │          │                      │                      │ https://avd.aquasec.com/nvd/cve-2021-25215                                   │
│                      ├─────────────────┼──────────┤                      ├──────────────────────┼──────────────────────────────────────────────────────────────────────────────┤
│                      │ CVE-2021-25214  │ MEDIUM   │                      │ 32:9.11.4-26.P2.amzn2.5.2 │ bind: Broken inbound incremental zone update (IXFR) can                   │
│                      │                 │          │                      │                      │ cause named to terminate...                                                  │
│                      │                 │          │                      │                      │ https://avd.aquasec.com/nvd/cve-2021-25214                                   │
└──────────────────────┴─────────────────┴──────────┴──────────────────────┴──────────────────────┴──────────────────────────────────────────────────────────────────────────────┘
```

Figura 2.8: Esempio di scansione di un'immagine VM con Trivy.

Tale funzionalità rimane, al momento della stesura del presente report, come funzionalità sperimentale e ancora in sviluppo.

2.4 Conclusioni

Nonostante entrambi gli strumenti si confermino come ottimi nel rilevare problematiche all'interno di container Docker, il confronto svolto ha evidenziato punti di forza e aree di miglioramento per entrambi. La scelta finale tra i due strumenti andrebbe infatti eseguita considerando l'ambiente di utilizzo, la grandezza del team di sviluppo ed eventualmente anche alla disponibilità di personale assegnato unicamente alla cybersicurezza. Snyk è più capace nel far nascondere al team i processi manuali, come quelli di aggiornamento delle dipendenze o dell'esecuzione di scansioni manuali ricorrenti. Inoltre, l'approccio più conservativo nella classificazione delle vulnerabilità può potenzialmente ridurre il rischio di allarmi non strettamente necessari. Dall'altra parte, Trivy si contraddistingue per un'altissima flessibilità ed, essendo totalmente open source, nessun lock-in con un fornitore esterno. Inoltre, con la capacità di essere più sensibile per un numero maggiore di vulnerabilità, si dimostra particolarmente efficace per gli ambienti ad alta sicurezza che richiedono una scansione approfondita e dettagliata, anche a costo di un maggiore lavoro richiesto per identificare le vulnerabilità effettivamente rilevanti. La sua funzionalità di scansione di immagini VM aggiunge un ulteriore strato di utilità, rendendolo adatto per ambienti che utilizzano una varietà di soluzioni di virtualizzazione. In conclusione, entrambi gli strumenti offrono vantaggi significativi e si completano a vicenda in diversi aspetti; pertanto, potrebbe essere persino vantaggioso considerare un approccio ibrido che sfrutti i punti di forza di entrambi per garantire la massima protezione e efficienza nel mantenimento della sicurezza dei container Docker.

Bibliografia

[1] Aqua Security. *Trivy docs*. URL: https://github.com/aquasecurity/trivy.

[2] Snyk. *Snyk docs*. URL: https://snyk.io/docs/.

[3] Sysdig. *Sysdig 2023 Cloud-Native Security and Usage Report*. URL: https://sysdig.com/blog/2023-cloud-native-security-usage-report/.