SPARKFABRIK

SPARKFABRIK **WHITE PAPERS**

GUIDA ALLA **CLOUD NATIVE SECURITY**

LE SFIDE ALLA SICUREZZA, GLI ATTACCHI CLOUD NATIVE E LE MIGLIORI STRATEGIE DI DIFESA

INTRODUZIONE

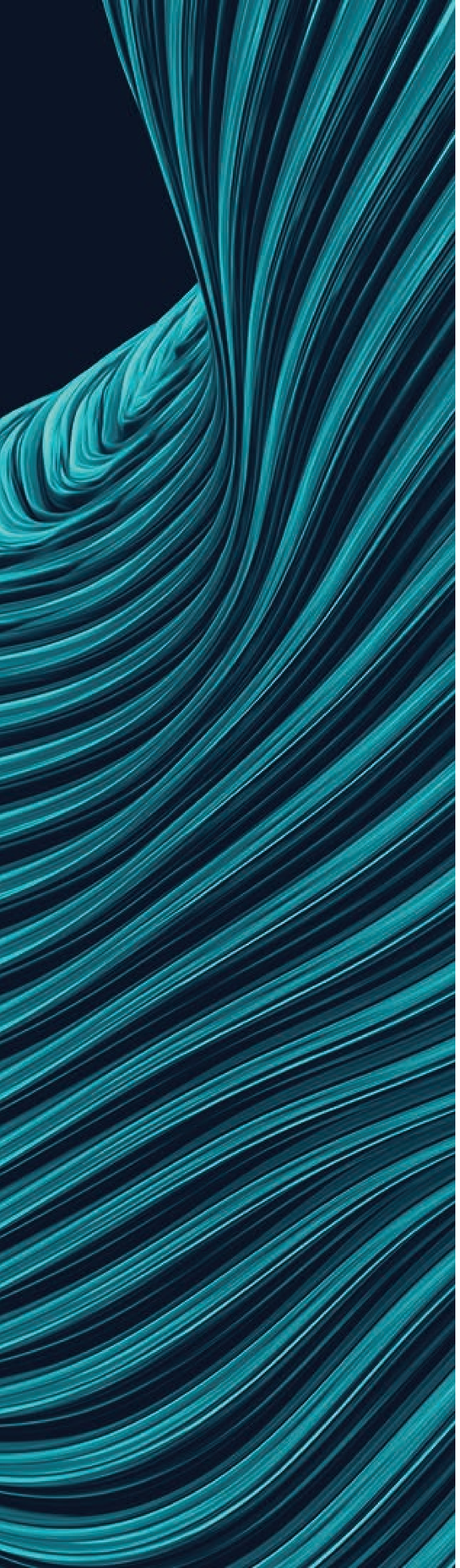
Le **tecnologie Cloud Native** permettono di ottimizzare le applicazioni esistenti, accorciare i tempi di sviluppo e ottenere ambienti integrati. È, insomma, la **risposta a un mercato moderno e mutevole**. A **partire dalla definizione ufficiale** della Cloud Native Computing Foundation, infatti, possiamo sinteticamente dire che il Cloud Native sia una metodologia per approcciare il mercato. Il suo scopo è rendere un'attività aziendale capace di reagire in termini rapidi al cambiamento delle richieste di mercato.

È questo il motivo per cui sempre più organizzazioni adottano un approccio Cloud Native. Progettare i propri servizi con applicazioni in ambienti cloud (con tutte le complessità di approccio che si possono immaginare: cloud privato, pubblico, cloud ibrido e multicloud) porta vantaggi tangibili e accelera l'innovazione dei business. Tuttavia, le sfide non mancano. E tra queste, **le minacce alla sicurezza si fanno sempre più complesse**.

Quali differenti implicazioni riguardanti la sicurezza caratterizzano un ambiente Cloud Native rispetto ad uno tradizionale?

In questo white paper dedicato a CIO e CTO risponderemo a questa e altre domande. Elencheremo i rischi di sicurezza da considerare e le principali minacce. Infine scopriremo quali sono le strategie per difendersi dagli attacchi, sempre più frequenti e sofisticati.

INDICE



CHE COS'È LA CLOUD NATIVE SECURITY

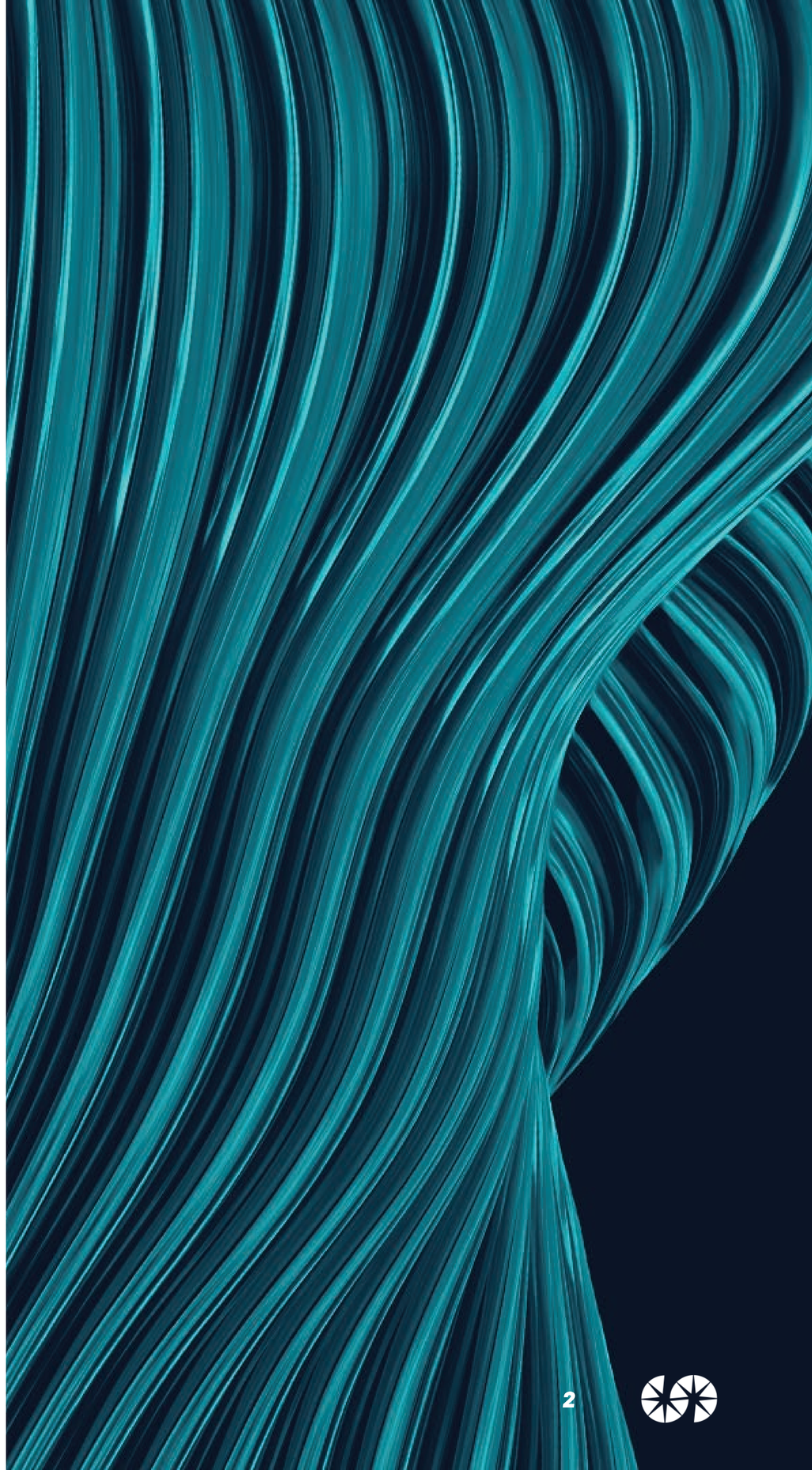
Per parlare di Cloud Native Security partiamo dal concetto di Cloud Native. La **Cloud Native Computing Foundation (CNCF)** definisce il Cloud Native come «*quell'insieme di tecnologie che consente alle organizzazioni di creare ed eseguire applicazioni scalabili in ambienti moderni e dinamici come cloud pubblici, privati e ibridi*».

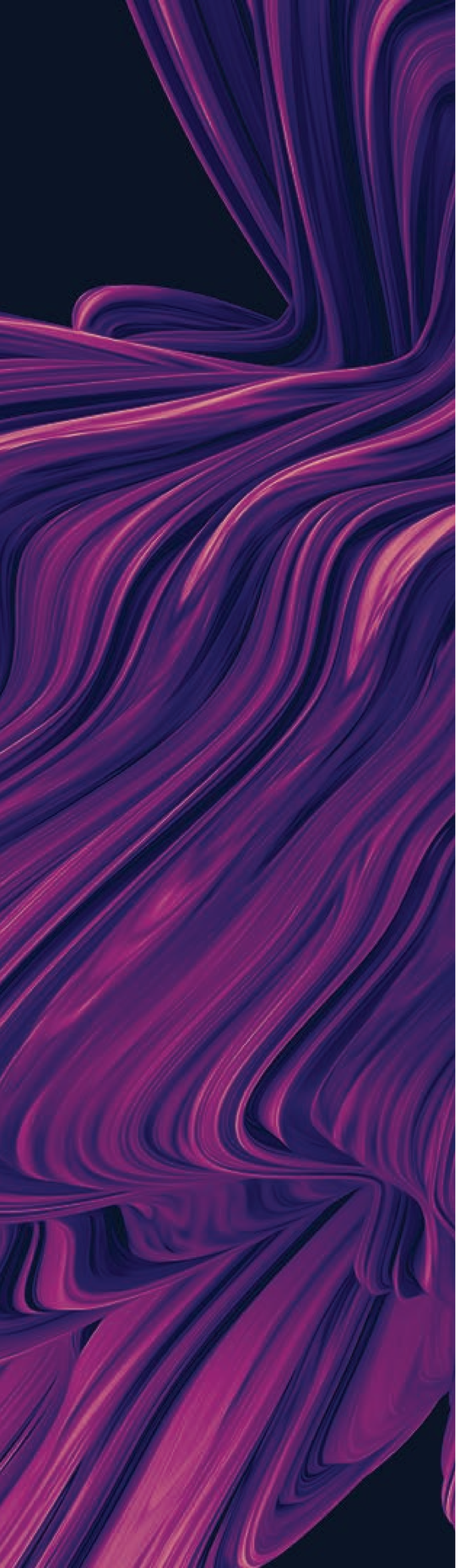
Quando si parla di **Cloud Native Security** ci si riferisce sia alla sicurezza dell'infrastruttura cloud sia alla sicurezza continua delle applicazioni che girano sopra di essa (Continuous Application Security).

La sicurezza deve essere integrata su più livelli, dal sistema operativo ai container che ospitano l'applicativo. Per proteggere un'applicazione bisogna entrarci: comprenderne i flussi di dati e le transazioni al fine di identificare i punti critici e prendere provvedimenti adeguati in fase di sviluppo e manutenzione della stessa.

Detto in un altro modo, le applicazioni con una struttura nuova e distinta come quelle Cloud Native richiedono, ad esempio, un approccio alla sicurezza molto diverso da quelle Cloud "tradizionali".

Infatti, a seconda della tecnologia utilizzata per eseguire i vari microservizi - siano essi fatti girare su macchine virtuali, container, o funzioni serverless - i problemi che si possono incontrare e le minacce da cui è necessario difendersi sono differenti. Dovrà quindi variare l'approccio per adattarsi ad ogni situazione specifica.





LE SFIDE DELLA CLOUD NATIVE SECURITY

Le minacce alla **sicurezza dei dati** possono provenire da molte aree, sia interne sia esterne. Ci sono delle criticità più generali che riguardano il cloud tradizionale e che valgono anche in ambienti Cloud Native, e poi ci sono quelle più specificamente rivolte ad architetture Cloud Native.

Ad esempio, una tipologia di attacchi del secondo tipo che vedremo più avanti è quella che mira a infettare le immagini sorgenti da cui vengono generati i container. Inoltre, sono sempre molto utilizzate (anche perché costantemente aggiornate) le tattiche per appropriarsi di token o credenziali e compiere attacchi sfruttando tecniche di privilege escalation.

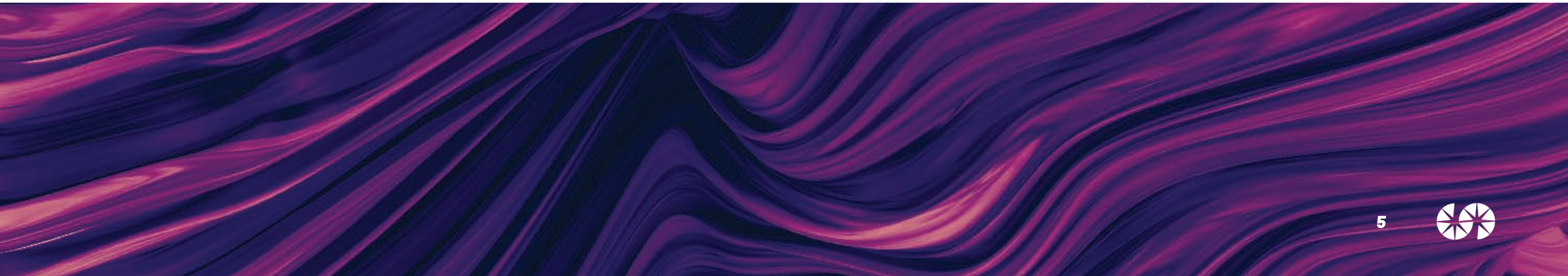
Identifichiamo di seguito 5 principali sfide alla sicurezza.

LA MANCANZA DI VISIBILITÀ

Man mano che la complessità degli applicativi rilasciati nel cloud aumenta, diventa cruciale tenere traccia dell'intero ciclo di vita del crescente numero di risorse. È necessaria anche la piena conoscenza di come funziona il proprio applicativo: questo comprende molto di più che non il semplice **saper identificare flussi anomali di richieste**.

Un ambiente di test, ad esempio, avrà un flusso minore di uno di produzione, e quest'ultimo potenzialmente maggiore durante particolari giorni dell'anno, e così via. La complessità crescente genera situazioni in cui possono crearsi comportamenti imprevisti, che sfuggono ai tradizionali sistemi di monitoraggio.

Occorrono altri metodi capaci di rilevare e aiutare a risolvere problemi mai visti prima, con l'obiettivo di mantenere i sistemi efficienti e affidabili. Qui entrano in gioco, oltre alle soluzioni di monitoraggio, anche quelle di Observability che vedremo più avanti.



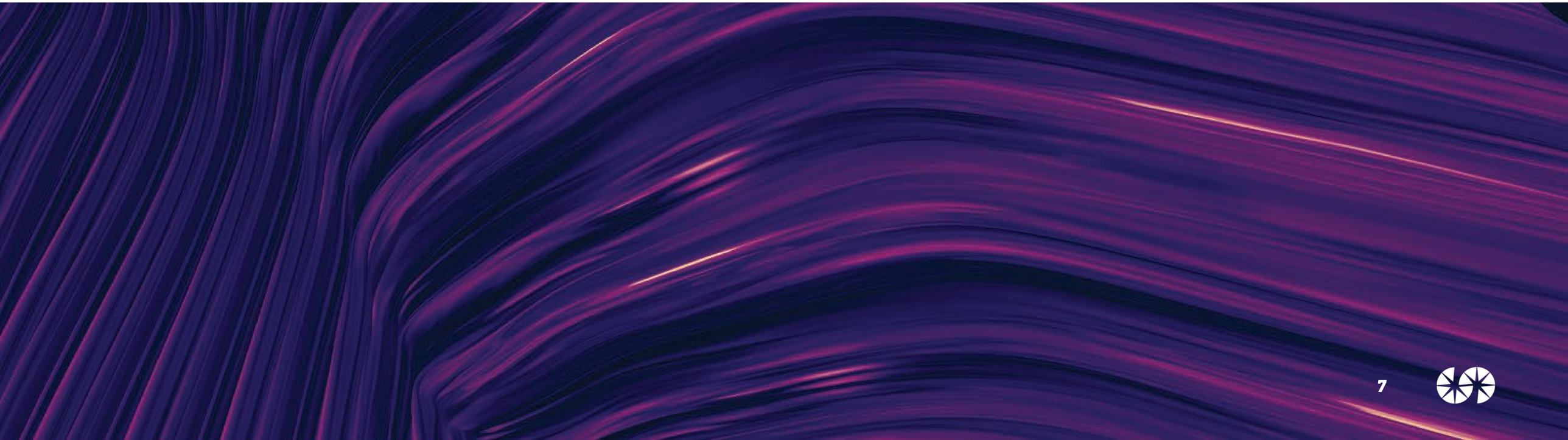
LA DIFFICILE APPLICAZIONE DI POLITICHE DI SICUREZZA

Negli odierni ambienti Cloud Native spesso si trova una varietà enorme di strumenti che rende difficile centralizzare le policy di sicurezza. L'utilizzo di best practice nello sviluppo, come il **Least Privilege Principle** - ovvero fornire solo gli stretti privilegi necessari a un applicativo o utente per eseguire i propri compiti - è solo uno dei modi di prevenire attacchi indesiderati.

Una grande sfida consiste proprio nel cercare di **unificare e automatizzare la creazione di regole, policy, avvisi e tecniche di riconciliazione** per facilitare i processi di hardening delle infrastrutture.

GLI ERRORI DI CONFIGURAZIONE

Spesso la causa di un attacco è nell'errata configurazione di uno o più componenti del sistema compromesso. **Secondo il Cloud Security Report 2020**, l**a maggiore minaccia nel cloud è l'errata configurazione**. In cima all'elenco, il 30% degli intervistati ha segnalato come problema «*password predefinita o assente per l'accesso a console di gestione*». Un dato a dir poco allarmante.



“

La maggiore minacca nel cloud è l’errata configurazione: il **30% degli intervistati** ha segnalato come problema “password predefinita o assente per l’accesso a console di gestione”.

”

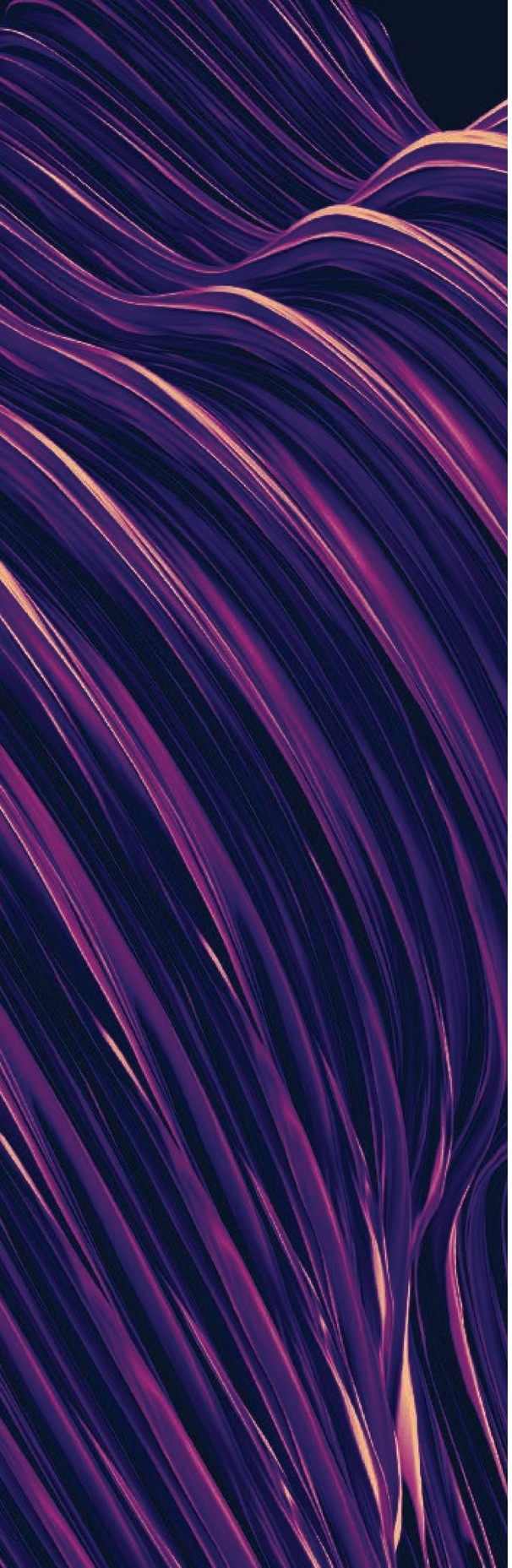
Fonte: Cloud Security Report 2020

I PROCESSI DI SECURITY LENTI

Un'altra sfida di sicurezza che le aziende che adottano un approccio Cloud Native devono affrontare riguarda l'**evoluzione delle applicazioni sottostanti**. Bisogna tenere il passo ed implementare policy sempre più sicure, senza però commettere l'errore di anteporre la sicurezza a efficienza e velocità.

La migliore strategia per trovare il giusto equilibrio tra sicurezza ed efficienza consiste nell'implementare e **automatizzare la sicurezza nelle prime fasi di creazione delle pipeline di rilascio del software**. In questo modo si preservano velocità e scalabilità proprie del mondo Cloud Native.

In questo nuovo contesto, la protezione dello stack tecnologico diventa ibrida per definizione: proteggere singoli host, VM o container non basta più.



MINACCE DI VARIA NATURA

Tra le sfide della Cloud Native Security non può mancare la protezione dei dati. Si può risalire ai dati sensibili usati per poi accedere a sistemi privati tramite tecniche di phishing, brute force o acquistando credenziali sul dark web.

Molti cloud provider mettono a disposizione diversi strumenti per analizzare il traffico dati, e **mettere in atto fin da subito le regole previste da veri e propri framework**. Esistono molti strumenti diversi che operano su differenti livelli e tecnologie. Un esempio di alto livello è il **MITRE ATT&CK**. Si tratta di una knowledge base accessibile a livello globale che raccoglie le tattiche avversarie che gli utenti malintenzionati utilizzano, come quella del lateral movement. In questo modo i comportamenti avversari possono essere studiati per costruire difese ai potenziali attacchi.

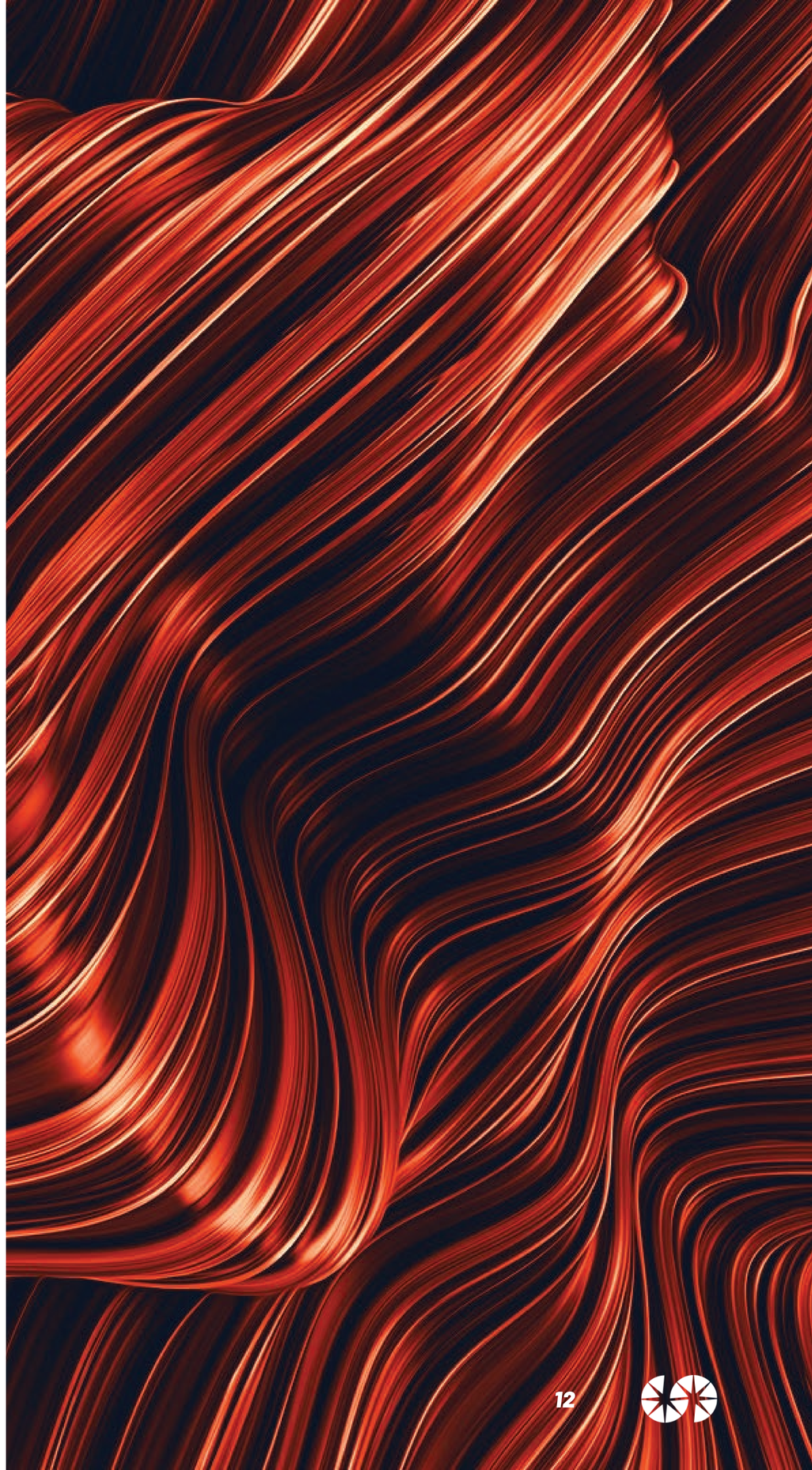
Tutti **i fornitori di cloud mettono a disposizione degli strumenti di monitoraggio** e analisi del traffico dati, così come terze parti specializzate in servizi SaaS, come ad esempio **CA Unified Infrastructure Management** o **vRealize Hyperic** di VMware. I super-vendor (Amazon AWS, Google e Microsoft) hanno realizzato molti strumenti di analisi dei loro cloud: da **Amazon Cloud Watch** di Amazon AWS, a **Monitoraggio di Azure** per Microsoft, sino a **Cloud Monitoring** di Google.

L'IMPORTANZA DELLA RESPONSABILITÀ CONDIVISA

Nell'IT tradizionale tutte le responsabilità spettano all'azienda, dal controllo degli accessi alla sicurezza fisica della struttura. Il cloud computing scarica molte di queste attività sul cloud provider, ma **l'azienda mantiene la responsabilità della protezione dei dati che inserisce nel cloud** secondo il modello di responsabilità condivisa (*Cloud Shared Responsibility Model*).

È fondamentale capire dove si trovano le responsabilità, che variano a seconda dei servizi che si consumano. Molte organizzazioni non ci riescono, quasi sempre per gli stessi motivi: patch critiche mancanti, account compromessi, erronea esposizione al pubblico di dati sensibili e accettazione di traffico da qualsiasi fonte. **Gartner**, la società di analisi di mercato, prevede che entro la fine del 2025 il **99% dei problemi nel cloud sarà colpa del cliente**, non del cloud provider.

La responsabilità deve quindi essere condivisa in modo consapevole. E non solo tra azienda e vendor, ma anche internamente alle organizzazioni. Se prima i team di Operations/Security avevano un ruolo chiave sulla sicurezza, ora l'approccio DevOps si arricchisce di un nuovo fondamentale tassello: il concetto di DevSecOps, di cui parleremo in seguito.

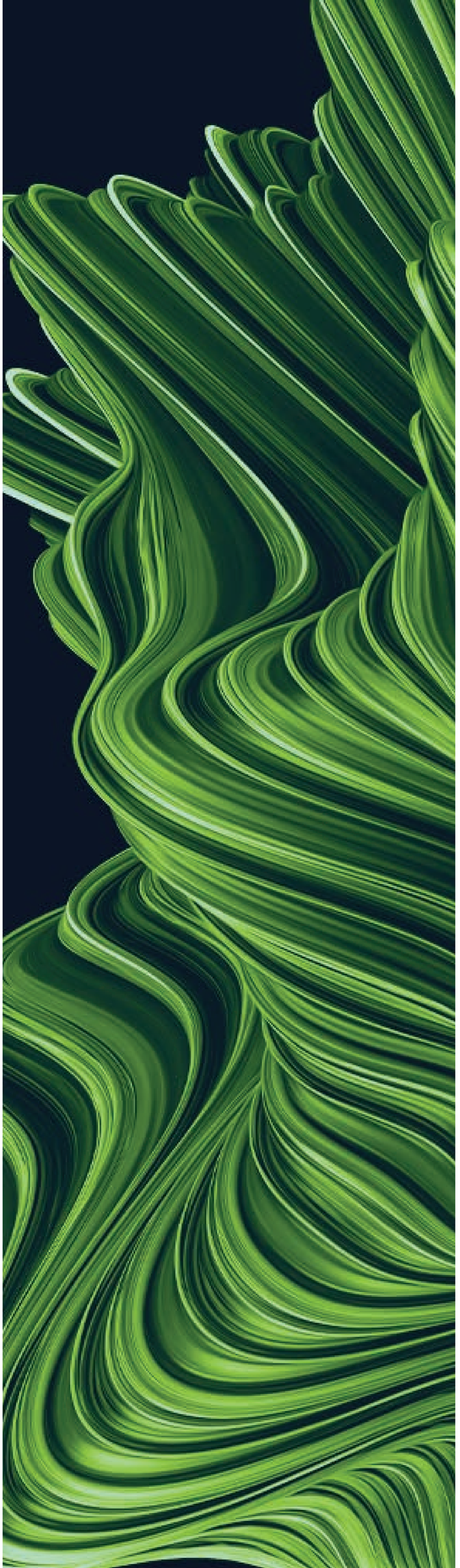


“

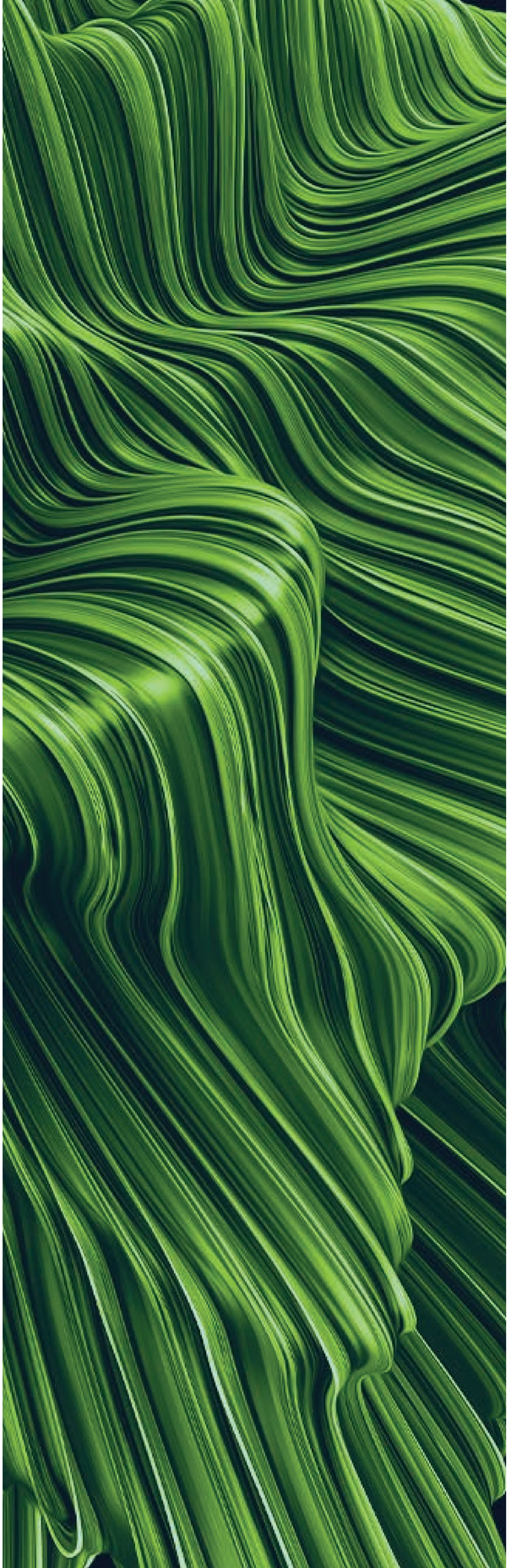
Entro la fine del 2025 il **99% dei problemi** di sicurezza nel cloud non sarà imputabile al cloud provider.

”

Fonte: Gartner



COME CAMBIANO LE MINACCE ALLA SICUREZZA NEL CLOUD NATIVE



Man mano che gli aggressori cambiano il loro approccio e si spostano verso attacchi Gen VI che sfruttano debolezze delle infrastrutture rilasciate sul public cloud, i professionisti della sicurezza e gli sviluppatori devono adeguare le proprie applicazioni di conseguenza.

Come dicevamo, spesso le debolezze delle infrastrutture sono dovute a disattenzioni. Ma va anche detto che gli attacchi sono sempre più sofisticati: esistono nuovi modi per inserirsi nel ciclo di rilascio nel mondo Cloud Native.
Nei prossimi paragrafi vedremo come difendersi da questi attacchi. Prima però mappiamo le **principali minacce alla sicurezza in ambito Cloud Native**.

Frequentemente l'obiettivo degli attacchi è quello di **infettare immagini sorgenti da cui vengono generati i container**, sfruttare dipendenze e componenti open source utilizzate più di frequente dagli sviluppatori. Infatti, gli sviluppatori spesso usano immagini di terze parti o pubbliche come punto di partenza per costruire i propri container. In questo modo i tempi di sviluppo si accorciano. Ma i rischi aumentano: queste immagini potrebbero essere corrotte o contenere codice dannoso.

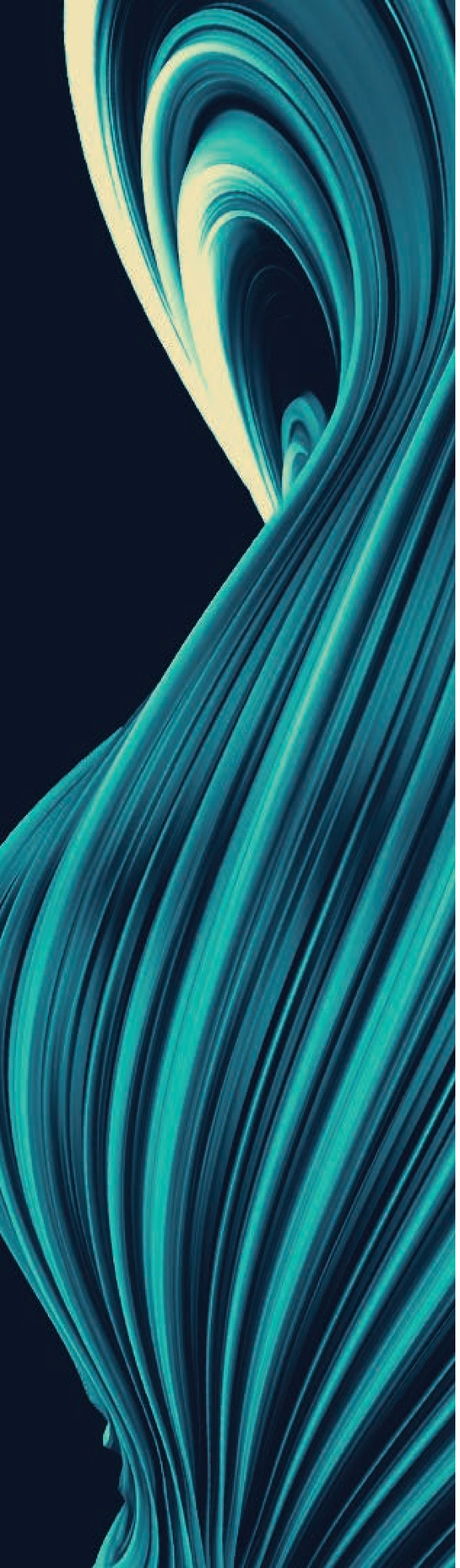
Ecco alcuni degli esempi di attacchi basati su immagini di container tratti dal **rapporto Aqua**:

Legittimare il nome di immagini docker: Sfruttando nomi che riconducono a immagini legittime, gli aggressori riescono più facilmente a far credere allo sviluppatore di aver utilizzato un'immagine affidabile, quando invece questa è stata preventivamente compromessa.

Build dell'immagine docker sugli host: In questo scenario l'autore dell'attacco sfrutta una Docker API port non configurata correttamente per creare ed eseguire una Container Image contenente codice malevolo. A differenza degli attacchi standard dove un'immagine dannosa viene scaricata da un docker registry pubblico, in questo scenario viene costruita localmente sull'host. Di solito si tratta di un cryptominer ma può essere anche uno script per lanciare un attacco denial-of-service di rete contro altri host o, peggio, un applicativo pensato per uscire dal container ed espandersi nella rete dell'host.

Exploiting dei processi di build e release: spesso vengono attaccati ambienti di sviluppo SaaS per eseguire criptomining. Attacchi di questa natura si sono concentrati su diversi ambienti di sviluppo tra cui Docker Hub, GitHub, Travis CI e Circle CI, abusando dei loro processi di build automatizzati.

Questi sono solo alcuni dei principali attacchi. Sul sito del MITREATT&CK si trova **una lista esaustiva di tecniche usate dagli attaccanti**.



IL CLOUD NATIVE THREAT REPORT DI AQUA

Secondo l'ultimo **Cloud Native Threat Report di Aqua**, azienda specializzata in Cloud Security, le minacce agli ambienti container-based sono aumentate drasticamente diventando più pericolose per via dei nuovi metodi utilizzati.

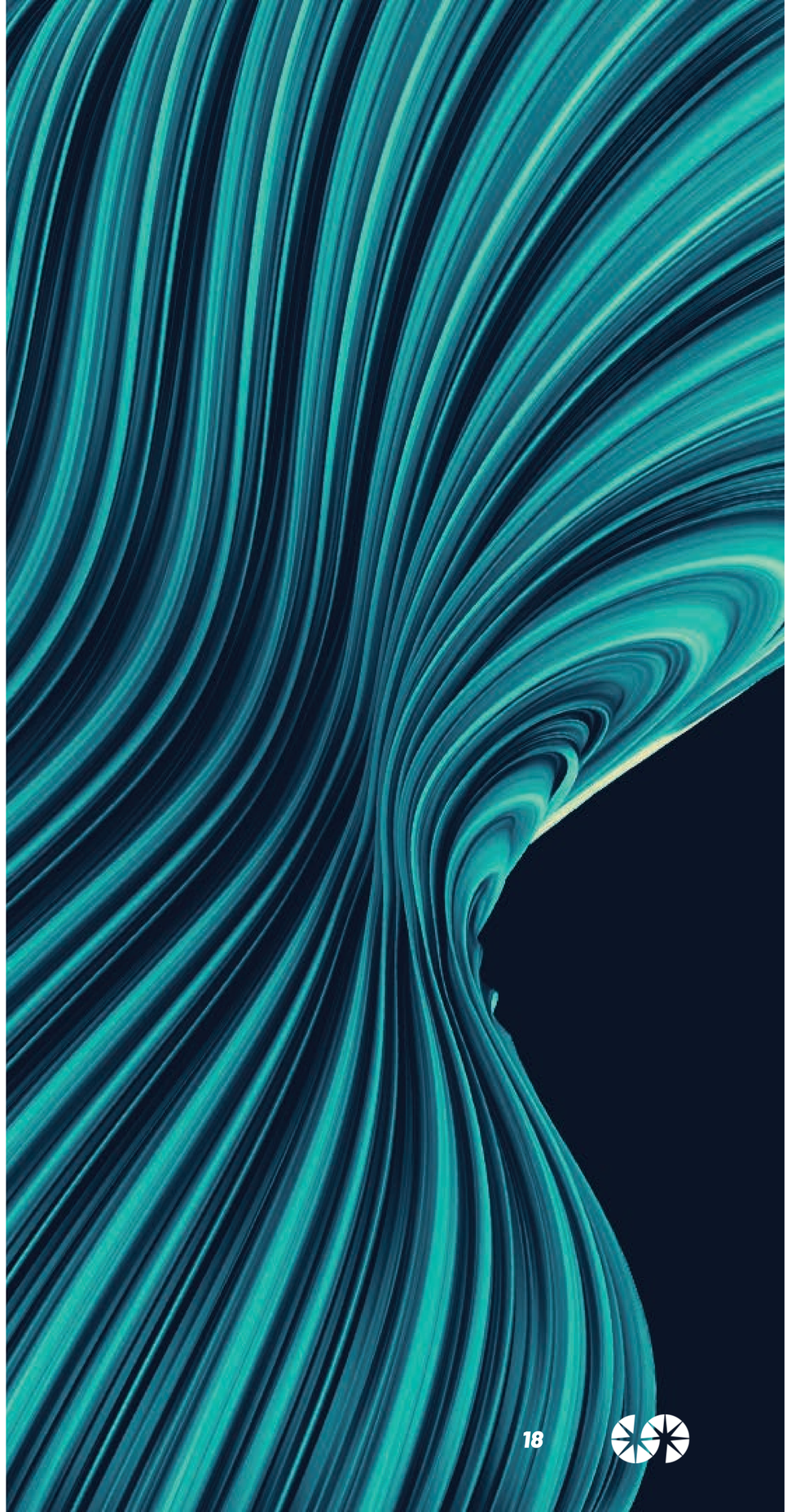
Il report fornisce un'analisi dettagliata degli attacchi di alto profilo scoperti, sottolineando i seguenti aspetti chiave:

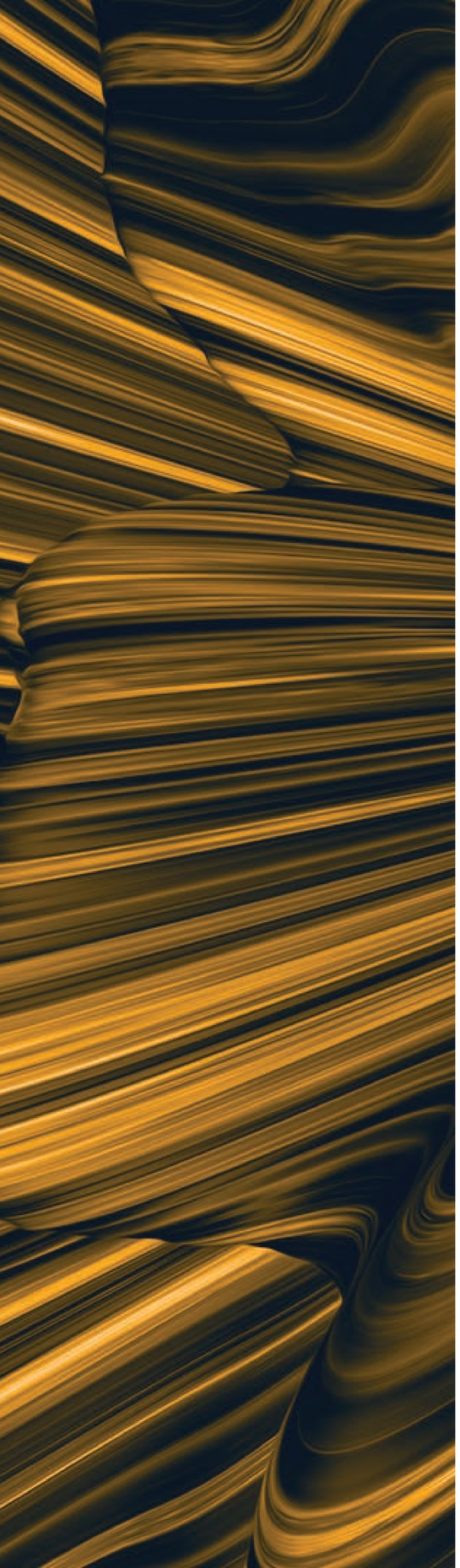
Gli attacchi sono più sofisticati - Gli aggressori hanno ampliato il loro uso di tecniche di elusione e offuscamento per evitare di essere scoperti.

Le botnet trovano e infettano rapidamente nuovi host man mano che diventano vulnerabili - Il 50% delle nuove API Docker mal configurate viene attaccato da botnet entro 56 minuti dalla configurazione.

Il mining di criptovalute è ancora l'obiettivo più comune - Oltre il 90% degli attacchi mira al dirottamento di risorse verso task di questa natura.

Maggiore utilizzo di backdoor - Il 40% degli attacchi ha coinvolto la creazione di backdoor sugli host remoti.



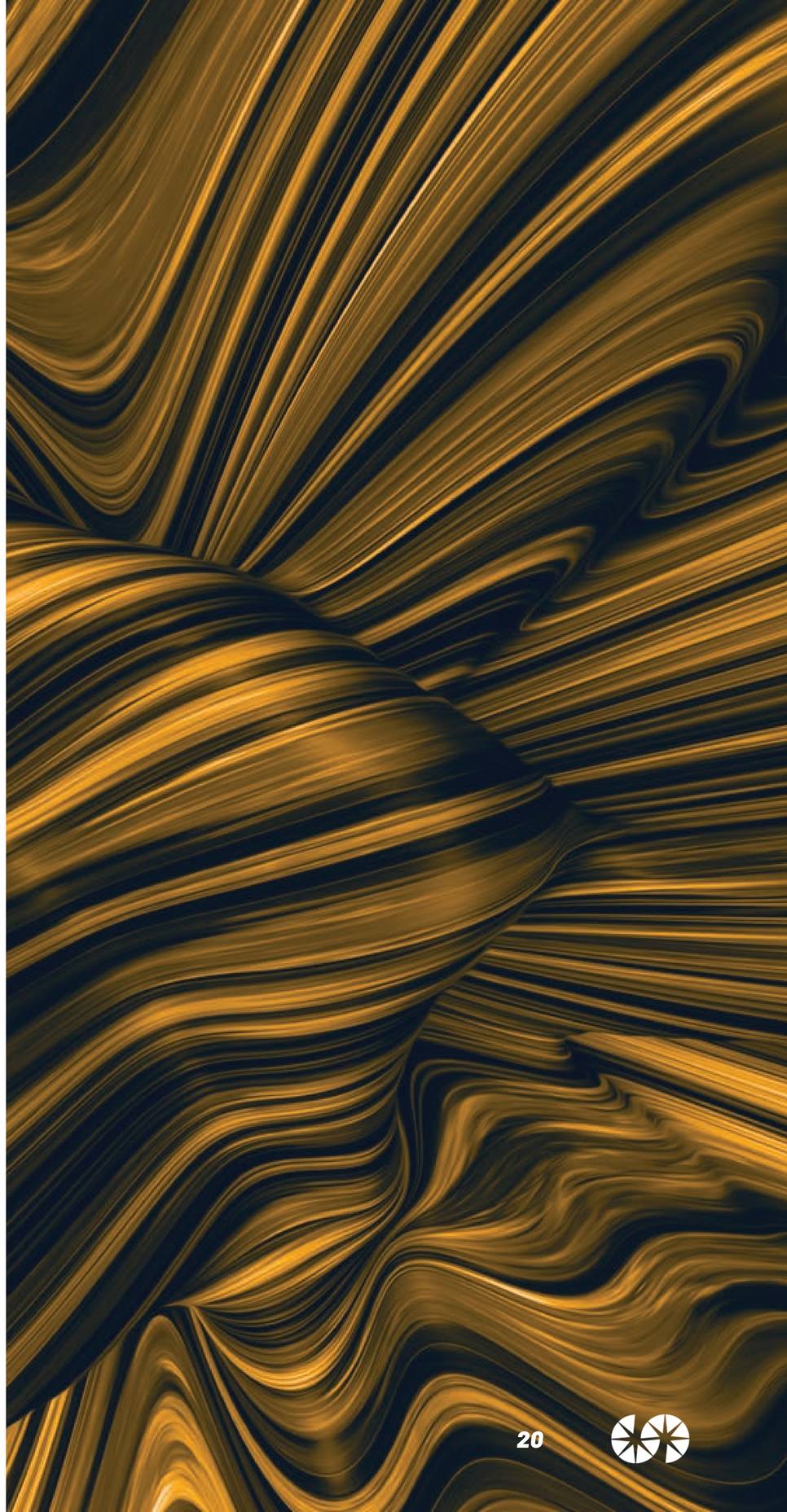


GLI ATTACCHI CLOUD NATIVE

Le applicazioni Cloud Native sono spesso progettate per essere eseguite tramite workload **effimeri**: girano su container immutabili, oggetti facilmente rimpiazzabili che scalano di continuo.

Tuttavia **la natura effimera di questi ambienti può essere un'arma a doppio taglio** in tema di sicurezza. Infatti, se da un lato gli aggressori non sono in grado di raggiungere facilmente un luogo virtualmente stabile all'interno di un sistema e, pertanto, devono cambiare continuamente target e tattica, dall'altro lato le tracce dei malintenzionati si perdono più facilmente negli ambienti effimeri.

Attacchi come il ***Groundhog Day Attack***, o il ***Poisoning the Well*** sono quasi all'ordine del giorno. Nel primo caso, l'attaccante si appropria solo di alcuni dati sensibili in modo molto rapido - per poi ripetere l'intero attacco da capo, sfruttando la scalabilità dell'infrastruttura. Nel secondo, l'attaccante riesce a fare *injection* di codice dannoso in una libreria open source per far sì che l'applicazione nel cloud possa interrogare un server esterno remoto sotto il proprio controllo e ricevere istruzioni malevole.



COME DIFENDERSI?

Abbiamo parlato delle sfide della Cloud Native Security e di come cambiano le minacce alla sicurezza e gli attacchi nel Cloud Native. Abbiamo anche preso atto dell'importanza della responsabilità condivisa. Entriamo ora nel cuore della seconda parte di questo white paper e impariamo a difenderci.

Esistono varie **tecniche difensive** per prevenire gli attacchi, ridurne l'efficacia, e addirittura vanificarne a priori l'effetto.

Per esempio, prima abbiamo parlato di nomi di immagini legittime per ingannare l'utente finale: come ci si potrebbe difendere da questa forma di attacchi? Alcune soluzioni a questo tipo di minaccia potrebbero essere:

- imporre un rigido controllo sui repository pubblici che possono essere utilizzati;

- fare scansioni automatiche delle immagini;

- sfruttare le firme digitali sulle immagini prodotte in modo da garantire il solo utilizzo di risorse del registro dei servizi (chiamate "artefatti") che siano state controllate.

Più in generale, ci sono dei **principi guida cardine di cui servirsi per evitare potenziali falle e migliorare l'AppSec**. Per prevenire gli attacchi questi principi devono essere definiti fin dalle prime fasi di configurazione dell'infrastruttura e degli applicativi. Vediamo di seguito quali sono.

ASSOCIARE AD OGNI FUNZIONE UN RUOLO MINIMALE

Abbiamo già menzionato questo principio e vale la pena ribadirlo. Anche conosciuto come *least privilege principle*, questo principio consiste nell'**assegnare un set quanto più ristretto e univoco possibile di autorizzazioni a qualsiasi entità** al fine di limitare i possibili danni in caso di exploit della stessa.

METTERE IN SICUREZZA LE DIPENDENZE

Molte dipendenze estratte da npm (Node.js), PyPI (Python), Maven (Java) o altri repository rilevanti sono soggette a bug o exploit.

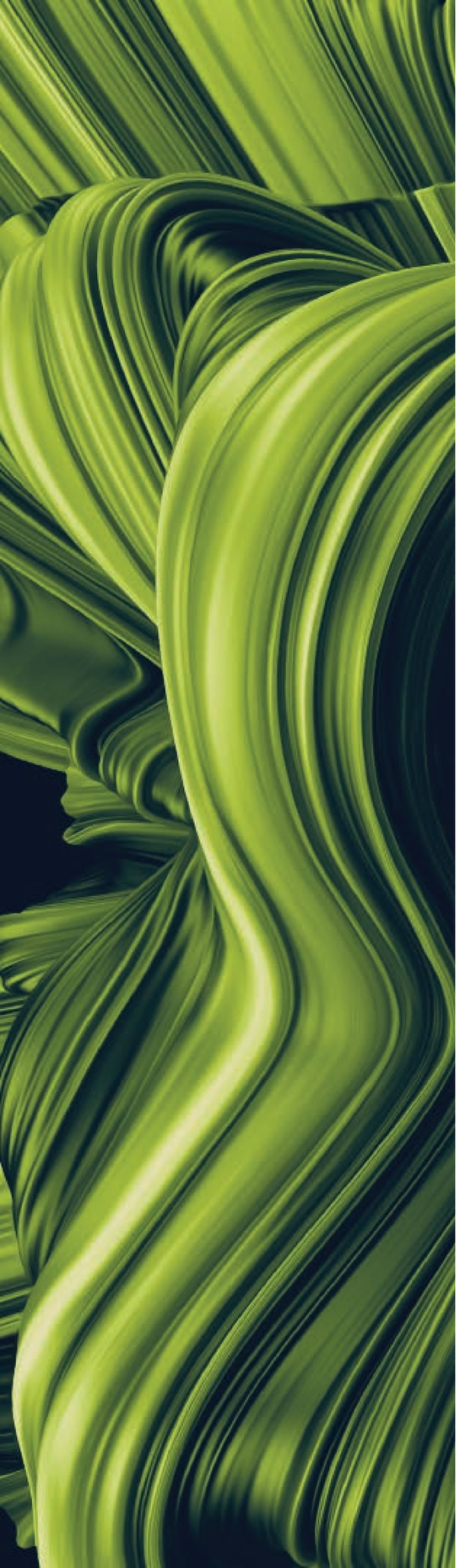
Diventa cruciale **l'accesso a database affidabili e strumenti automatizzati di controllo** (spesso integrati nei container registry di alcuni cloud provider, come per esempio **in Container Registry di Google Cloud** oppure **Harbor**). L'uso di questi strumenti impedisce l'utilizzo di nuovi pacchetti vulnerabili e permette di essere a conoscenza di potenziali CVE prima che queste vengano sfruttate da terzi.

LIMITARE I RISCHI DELL'AUTOMAZIONE

Vale in particolar modo - ma non solo - per le parti di *Infrastructure as Code* (IaC): è importante parametrizzare e **testare la propria codebase per evitare che gli errori si moltiplichino**. L'approccio IaC porta con sé lo stesso rischio di qualsiasi altro tipo automazione: se si commette uno sbaglio, questo viene facilmente replicato su tutte le macchine e gli apparati dell'infrastruttura.

Questo accade, ad esempio, anche perché i sistemi come quelli basati sul paradigma dell'*Infrastructure as Code* sono tendenzialmente complessi da capire. La loro gestione richiede infatti una notevole capacità di astrazione che permette di prevedere non solo quello che si fa ma anche le conseguenze di quello che è stato fatto.

Limitare i rischi di replicazione degli errori significa anche poter fare **rollout di una nuova versione del software** in modo sicuro e controllato su tutti gli ambienti (qa, test, produzione), senza lasciare più spazio di manovra all'attaccante che vuole sfruttare una particolare falla di sicurezza.

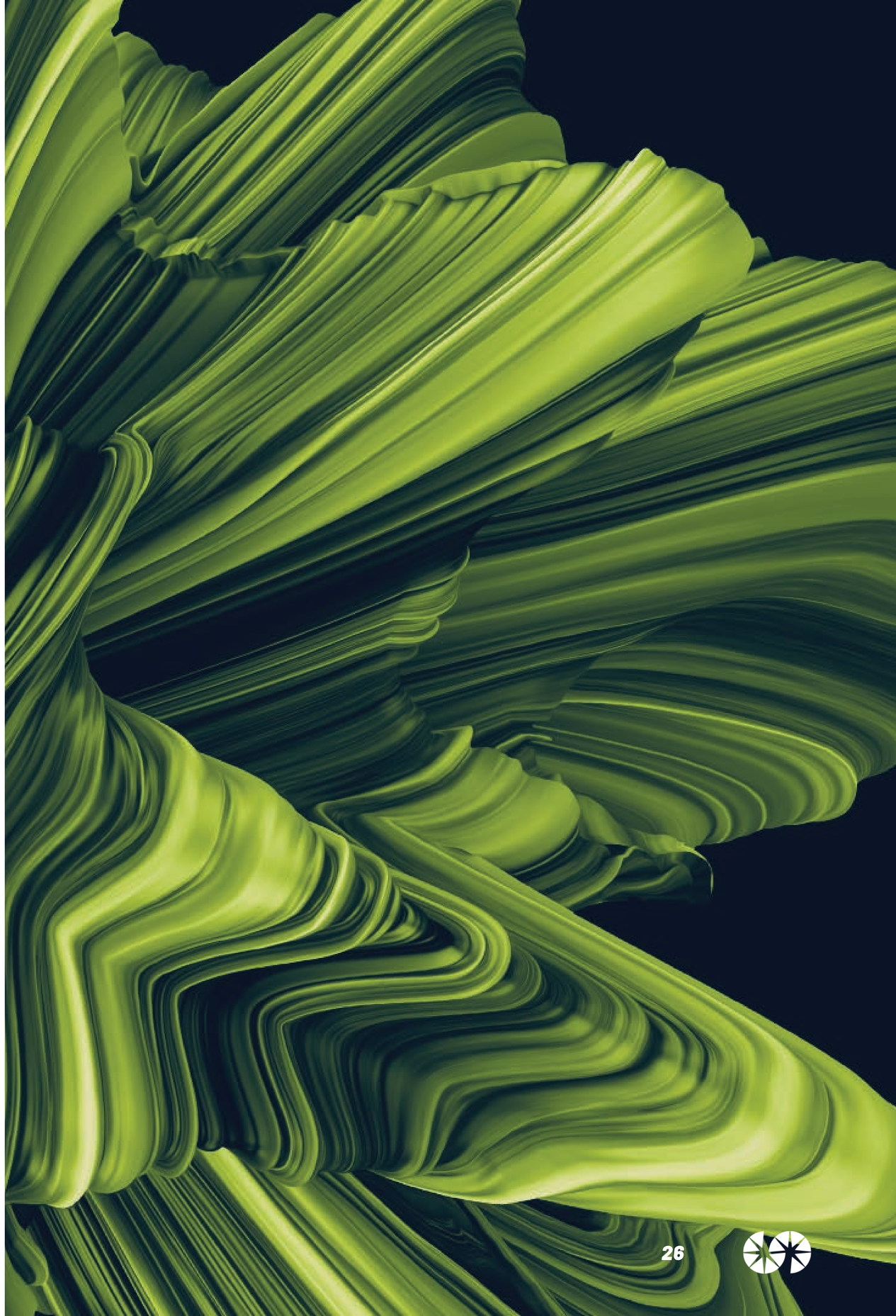


CLOUD NATIVE SECURITY PLATFORMS

Molti vendor mettono a disposizione delle **piattaforme per la security delle applicazioni Cloud Native**. Queste piattaforme permettono - senza richiedere particolari investimenti in tempo e risorse - di adottare soluzioni gestite o parzialmente gestite che includono intelligenza artificiale (AI), automazione, intelligence, rilevamento delle minacce e capacità di analisi dei dati senza bisogno di configurazione a basso livello.

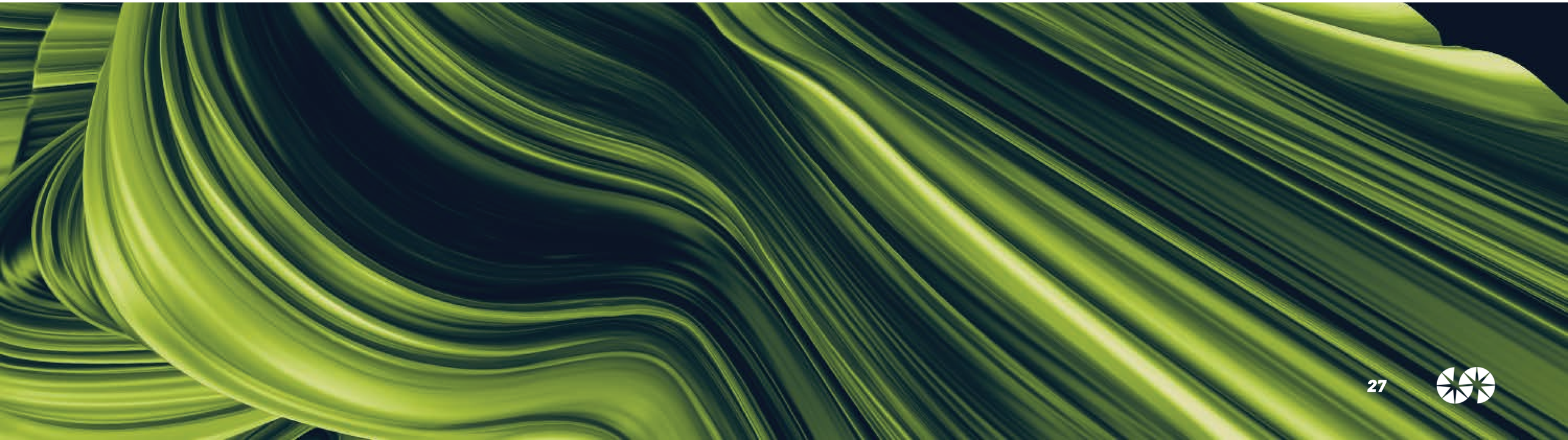
Gartner, già citata prima, le chiama "**Cloud Native Application Protection Platforms**" **(CNAPP)**. Mette così l'enfasi sul bisogno da parte delle aziende di focalizzarsi su soluzioni di sicurezza Cloud Native che forniscono un approccio completo a tutto il ciclo di vita del software, anziché su un mix di strumenti di sicurezza diversi tra loro.

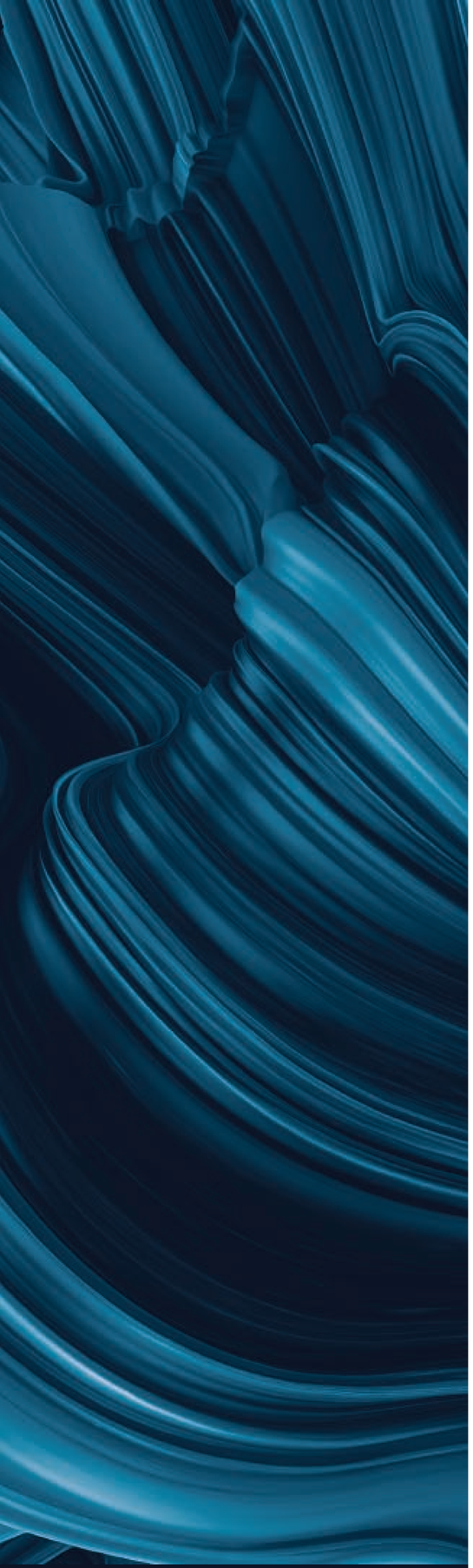
Da questo punto di vista le **CNAPP raccolgono in un unico approccio differenti modelli di sicurezza Cloud Native**: Cloud Security Posture Management (CSPM), Cloud Service Network Security (CSNS) e Cloud Workload Protection Platform (CWPP).



Mentre un'offerta di vere e proprie soluzioni CNAPP ancora non è presente sul mercato, esistono **varie soluzioni che si occupano di singole aree della sicurezza Cloud Native**. Tra queste citiamo Halo di CloudPassage, mirata alla protezione dei workload, la soluzione modulare Qualys, che comprende gestione compliance e vulnerability scanning, Prisma Cloud di **Palo Alto Networks**, Cloud One di **Trend Micro**, Cloudhealth e Secure State di **VMware**.

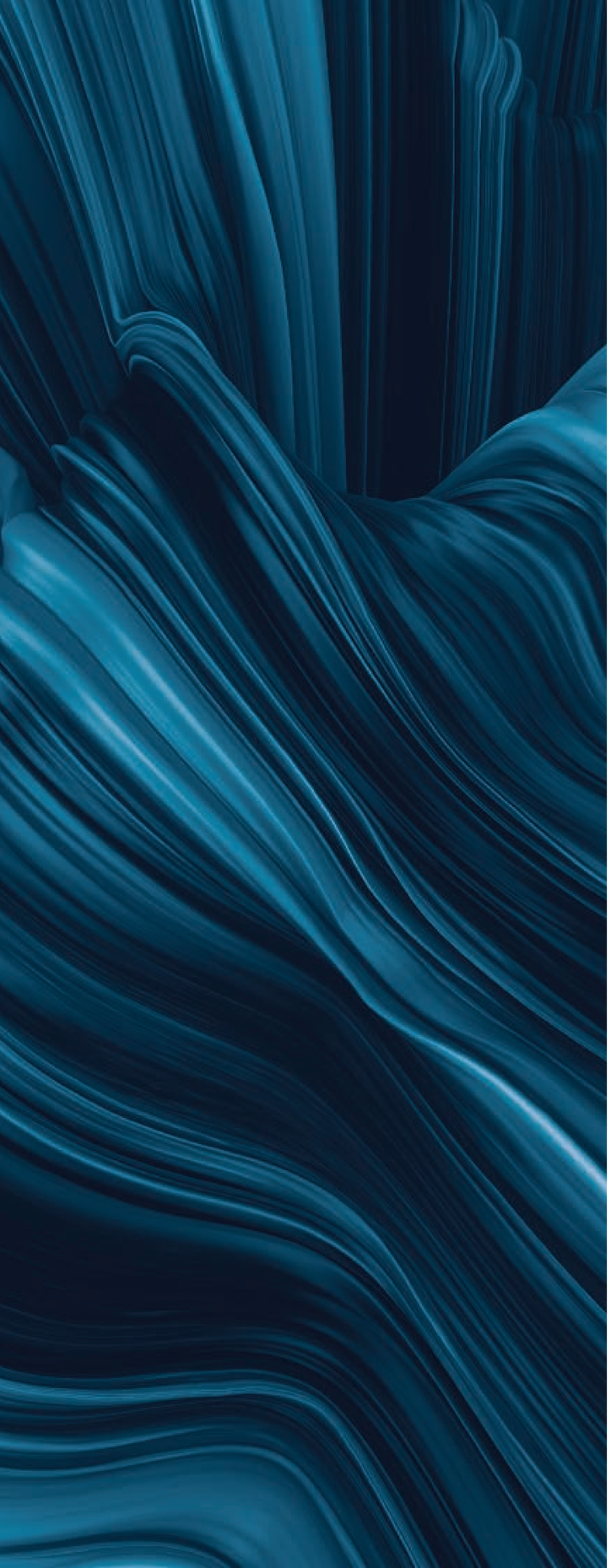
Le Cloud Native Security Platforms sono l'ideale per cominciare ad avere maggiore visibilità dei problemi, talvolta anche efficaci quanto basta a prevenire le maggiori vulnerabilità, ma non sostituiscono il cambio culturale che deve avvenire internamente ai propri team. Come già sottolineato, **l'approccio security first** deve essere condiviso all'interno dell'organizzazione. Proprio per questa ragione si parla di DevSecOps.





L'APPROCCIO DEVSECOPS

La sicurezza non può essere un'aggiunta tardiva o un ripensamento a posteriori: invece, deve essere sviluppata assieme all'applicazione. Questo però va in conflitto con **l'approccio DevOps** che, se da un lato rende più veloce e flessibile lo sviluppo delle applicazioni, dall'altro prevede che le pratiche di sicurezza vengano realizzate alla fine del ciclo di sviluppo e vengano testata una sola volta, alla fine.



Per risolvere questo potenziale ingorgo che mette oltretutto molta pressione sugli specialisti della security, è nato un approccio che estende e integra quello delle DevOps. Si chiama **DevSecOps e rende la sicurezza parte integrante delle DevOps**.

Si tratta di un framework di collaborazione che espande l'impatto della filosofia DevOps, **aggiungendo pratiche di sicurezza al processo di sviluppo e distribuzione del software**. È un cambiamento importante che promuove anche una cultura diversa per la collaborazione tra gli ingegneri del software e gli specialisti della sicurezza, chiamata "**Security as Code**".

Quello delle DevSecOps è un mercato in forte crescita, che secondo una ricerca di **IndustryARC** arriverà a quota 6,5 miliardi nel 2025. **L'impatto sullo sviluppo del software è immediato**: non si possono commutare modifiche alla codebase sul repository se non sono state validate dai test di sicurezza. Esistono vari strumenti per verificare la presenza di vulnerabilità (**Anchore**, **Clair**, **Dagda**) e per automatizzare i test (ci sono tra gli altri **Selenium**, **Katalon**, **Ranorex** e **SmartBear**).

“

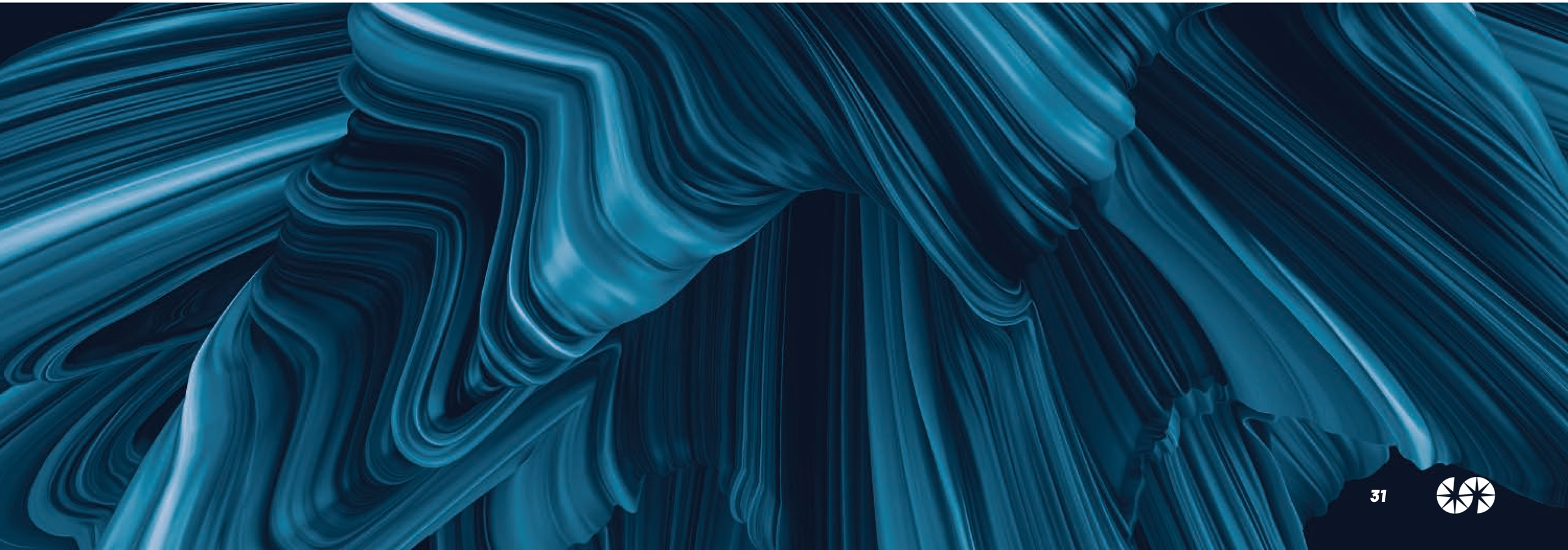
Il mercato DevSecOps raggiungerà quota 6.5 miliardi di dollari nel 2025

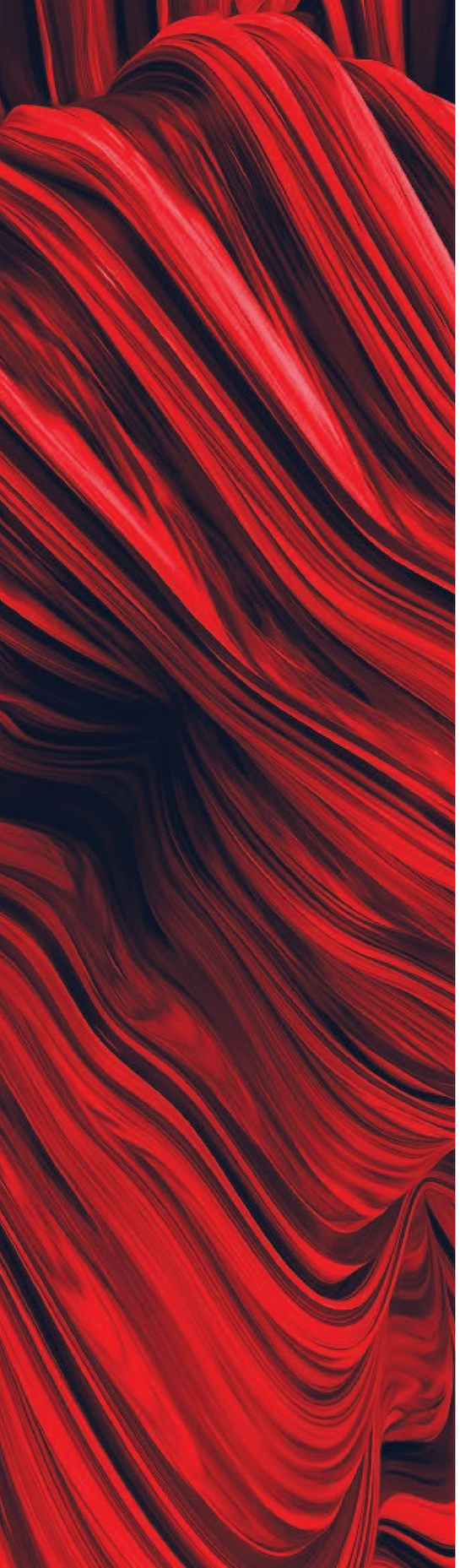
”

Fonte: IndustryARC

Un aspetto molto importante, tuttavia, è quello della formazione del personale. Dato che **nelle DevSecOps non esiste più un team specifico dedicato alla sicurezza**, occorre organizzarsi per i programmi di certificazione, workshop, laboratori pratici ed eventi come gli hackaton.

Inoltre, un aspetto sempre più rilevante, soprattutto in alcuni settori come quello finanziario, è la **compliance** e quindi la necessità di inserire degli auditing come parte della pipeline CI/CD. Esistono tool dedicati anche per questo come **Netwrix**, **Libryo** e **Integrum**.





TECNOLOGIE PER LA CLOUD NATIVE SECURITY

La **CNCF** fornisce un landscape molto interessante di tecnologie che possono aiutare a implementare la security attraverso i vari livelli di astrazione di un'infrastruttura e/o un applicazione Cloud Native.

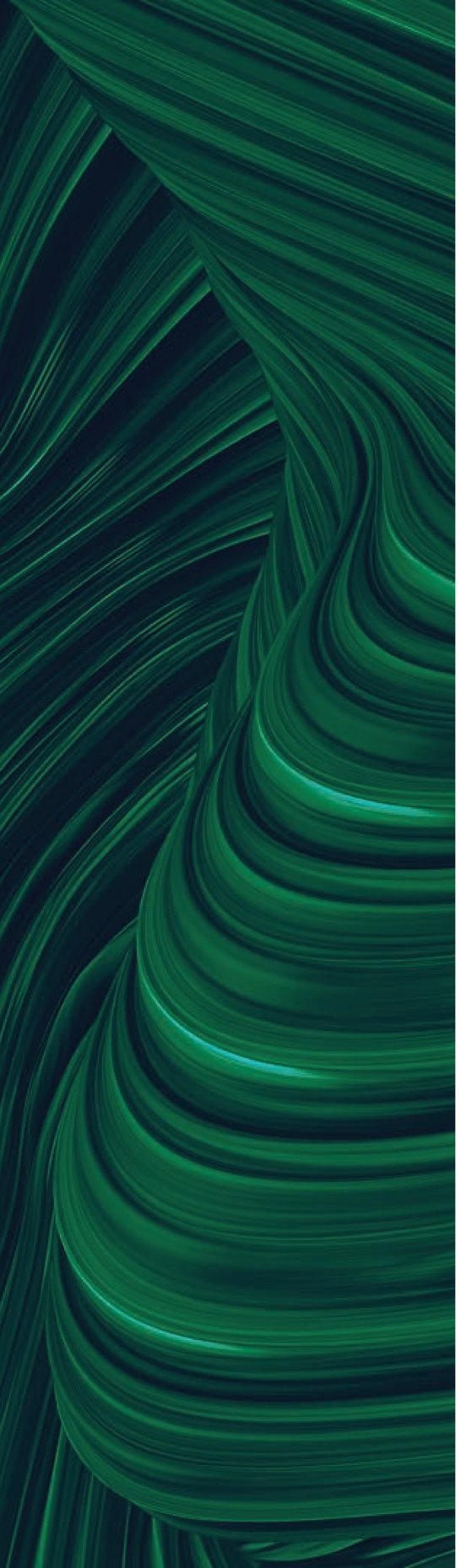
Kyverno, ad esempio, è un engine per policy progettato per Kubernetes. Con Kyverno, le policy vengono gestite come risorse Kubernetes e non è necessario un nuovo linguaggio per scriverle. Ciò consente di utilizzare strumenti familiari come kubectl, git e kustomize per gestire le autorizzazioni. Le policy Kyverno possono convalidare, modificare e generare risorse Kubernetes. Inoltre Kyverno viene rilasciato con una CLI che può essere utilizzata per testare le policy e convalidare le risorse all'interno di una pipeline di CI/CD.

L'**Open Policy Agent** - progetto open source e domain-agnostic - cioè che permette di descrivere qualsiasi tipo di invariante nelle policy - fornisce un linguaggio dichiarativo di alto livello che consente di costruire le policy sotto forma di codice e fornisce delle semplici API. Quando l'applicazione deve implementare autorizzazioni, interroga le API OPA fornendo dati in input tramite un documento JSON, il quale descrive criteri di autorizzazione specifici su microservizi, Kubernetes, pipeline CI/CD, API Gateway, etc. che saranno applicati dalle API stesse.

Snyk è un tool che permette di trovare e correggere automaticamente vulnerabilità nelle dipendenze open source, nel codice della tua applicazione, nelle immagini Docker e nei cluster Kubernetes. È anche in grado di identificare e correggere configurazioni non sicure in codice Terraform e Kubernetes.

Calico - sempre open source - può essere utilizzato per applicare sicurezza di rete sui container, sulle macchine virtuali e workload basati su host bare metal e per facilitare la messa in atto della **zero trust security**.

Notary è un progetto open source che implementa The Update Framework (**TUF**). Il TUF definisce un insieme di librerie, formati di file e utility che possono essere utilizzati per proteggere i sistemi di aggiornamento software nuovi ed esistenti, quali gestori di pacchetti, sistemi di aggiornamento delle singole applicazioni o di librerie software. In pratica, Notary si occupa delle operazioni necessarie per creare, gestire e distribuire i metadati necessari per garantire l'integrità e l'affidabilità dei contenuti, semplificandone le procedure.



I 7 PUNTI CHIAVE DELL'ULTIMO CLOUD NATIVE SECURITY SUMMIT

Siamo giunti all'ultima parte di questa guida alla Cloud Native Security. Ci sembra utile citare i punti salienti dell'ultimo **Cloud Native Security Summit di Capsule8**, evento che riunisce i professionisti della sicurezza più esperti e gli innovatori del settore per discutere di sicurezza in ambito Cloud Native. Nel recente summit l'attenzione è stata portata principalmente su 7 punti chiave, vediamoli di seguito.

1. CHE COS'È LA RESILIENZA INFORMATICA NEL MONDO CLOUD NATIVE?

Sappiamo che, nonostante l'applicazione dei principi di sicurezza, a volte gli incidenti sono inevitabili. Secondo Rob Duhart, Responsabile della Sicurezza di Google, la resilienza consiste nel saper fornire una risposta alla domanda «***quanto sei preparato per quando accadrà quel determinato incidente?***». Facciamo un esempio concreto.

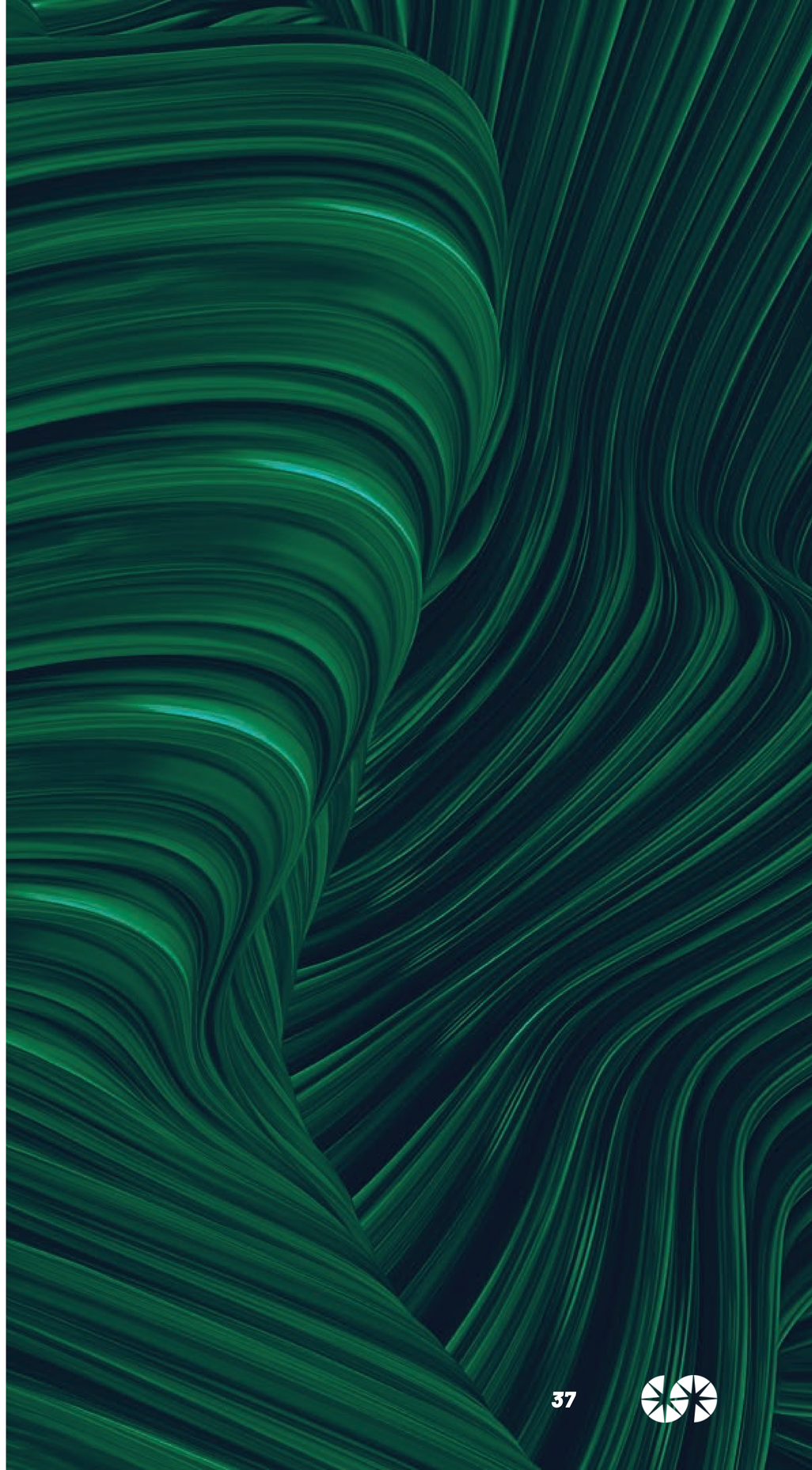
Molti sviluppatori utilizzano sempre più **funzioni serverless**: in uno scenario tipico, le funzioni lambda sono concatenate da Step Function. Se un Lambda fallisce, significa che un passaggio fallisce e l'intera Step Function termina. Come evitare che questo accada? In questo caso la soluzione per rendere meno vulnerabile il sistema è disaccoppiare il più possibile.

Riportando questo esempio in un modello di comportamento generale: è essenziale **giocare d'anticipo**. Le organizzazioni devono valutare i propri sistemi fin dalla progettazione dell'architettura, disporre piani di rilevamento e risposta agli incidenti, testare i propri piani e creare alert di sicurezza. Solo così è possibile ripristinare uno stato stabile quando si verifica l'inevitabile.

2. OTTIENI VISIBILITÀ SENZA COMPROMETTERE LE PRESTAZIONI

Secondo Chaim Mazal, Responsabile dell'IT Security di ActiveCampaign: «*Bisogna mettere in piedi logging, strumenti, dashboard appropriate e garantire che i Team di Sicurezza abbiano alta visibilità: bisogna conoscere esattamente quali sono i workload giornalieri all'interno dell'infrastruttura*».

Tuttavia, ottenere visibilità non è vantaggioso se questo traffico di dati aggiuntivo ostacola le prestazioni delle applicazioni in produzione. Vanno identificati strumenti di sicurezza appropriati in stretto coordinamento con i team DevOps e SRE per raggiungere un **equilibrio che non sacrifichi le operazioni di routine**.



“Bisogna mettere in piedi logging, strumenti, dashboard appropriate e garantire che i Team di Sicurezza abbiano alta visibilità: bisogna conoscere esattamente quali sono i workload giornalieri all’interno dell’infrastruttura.”

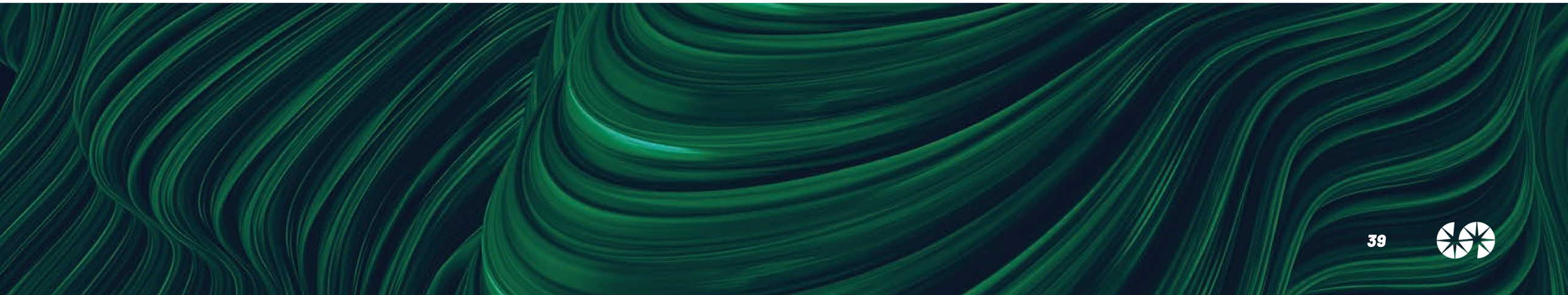
Chaim Mazal, Responsabile IT Security di ActiveCampaign

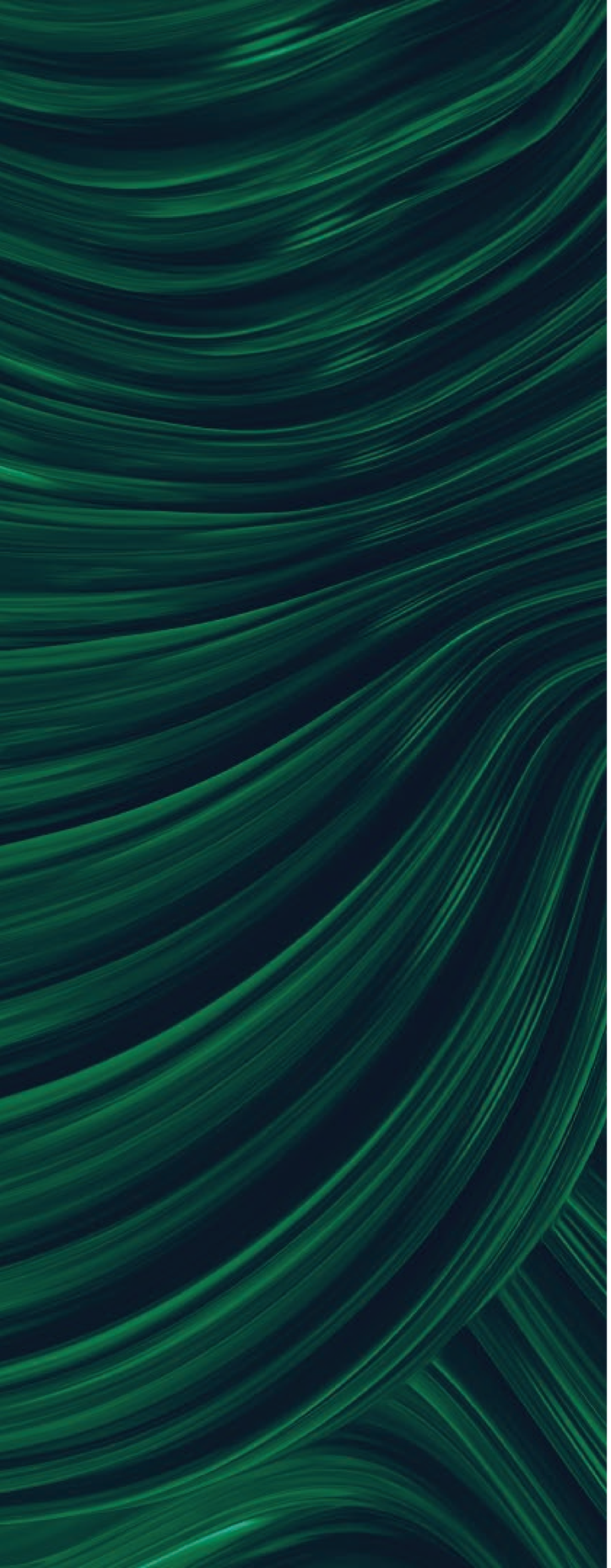
3. CAMBIA IL MODO DI RILEVARE I PROBLEMI

Avere la visibilità non basta: è richiesto anche un livello più granulare rispetto al passato (perché devono rendere visibili i comportamenti di container e funzioni serverless) e la stessa natura effimera dei carichi di lavoro cloud rende il raggiungimento di questo obiettivo più impegnativo.

Uno dei vantaggi dell'approccio Cloud Native, però, deriva dalla sua natura fortemente automatizzata, che garantisce un **approccio decentralizzato verso il rilevamento e gli avvisi**. Per esempio, alcune organizzazioni adottano un particolare modello di SOC (*Security Operations Center*).

Un SOC è un esempio tipico di software **SaaS (Software as a Service)**, cioè di software che opera nel cloud come servizio in abbonamento. In questo modello tradizionale il SOC viene fornito come una forma di competenza per la sicurezza informatica dell'azienda noleggiata dall'esterno. Alcune aziende, tuttavia, adottano un modello di SOC dedicato o SOC interno, che viene realizzato sotto la responsabilità del CISO (Chief Information Security Officer dell'azienda) con personale interno e che si integra con le metodologie di sviluppo DevSecOps.

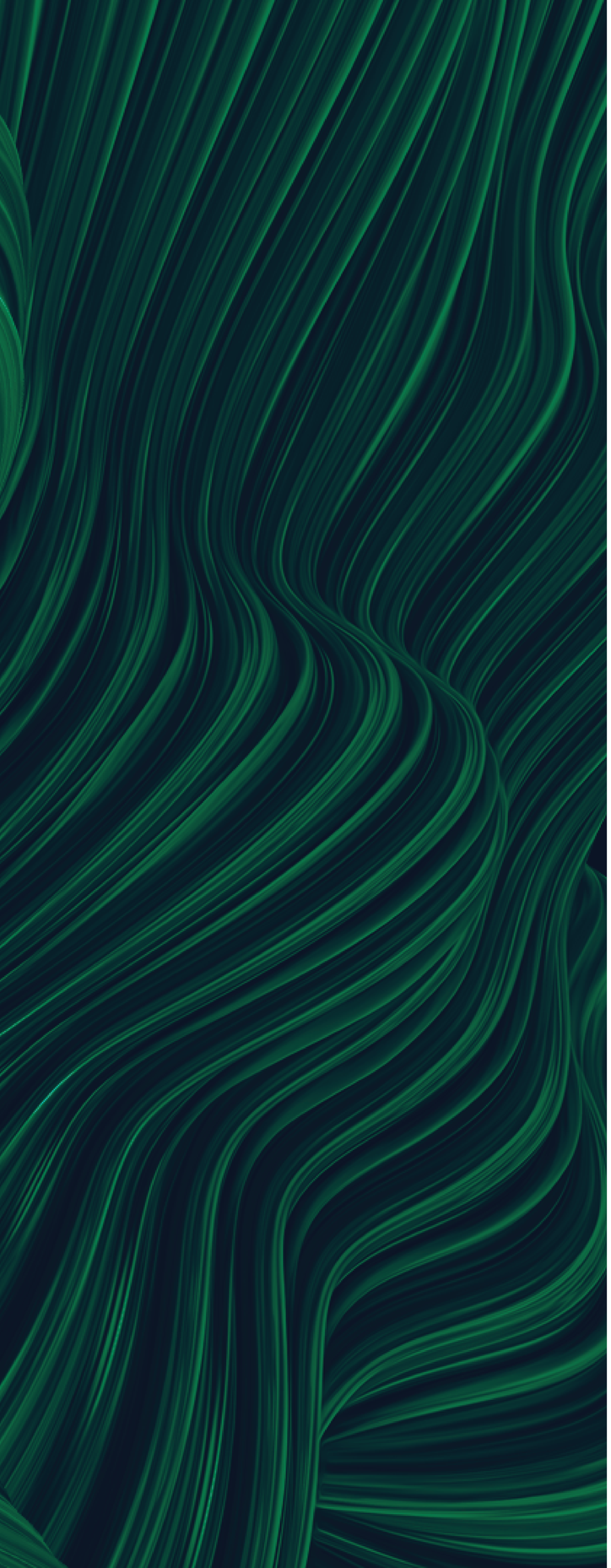




4. GLI ALERT DEVONO ESSERE "AZIONABILI"

Il paradigma di progettazione Cloud Native ha molti vantaggi, tra i quali c'è anche quello di avere accesso a molte funzionalità di logging. Detto in un altro modo: sviluppare applicazioni Cloud Native apre a un mondo di soluzioni avanzate per il **logging**, monitoraggio e più in generale osservabilità. Infatti, avere una maggiore quantità di dati è sicuramente meglio per indagare un problema, anche se l'attivazione di molti canali di logging può influire sulle prestazioni e causare molto rumore.

Kathy Wang, CISO di Very Good Security suggerisce a questo proposito: «*Spesso vedo situazioni di alerting eccessivo. [..] Avere un sistema di alerting veritiero e affidabile è davvero importante: questo perché nessuno vuole essere avvisato per cose per cui non si può far nulla, o per le quali non è necessario fare nulla*».



Cloudflare e Snowflake coinvolgono anche i propri team DevOps e di sviluppo al fine di costruire **pipeline e alert che riducano il più possibile i falsi positivi**.

Più in generale, è necessario capire che i sistemi di monitoraggio sono complicati da progettare e difficili da realizzare, soprattutto in ambienti che cambiano costantemente. Spesso i meccanismi di osservazione (logging, monitoring) vengono configurati solo in un secondo momento, alle volte solo dopo che si sono verificati dei problemi, e non vengono sempre aggiornati quando cambiano i carichi di lavoro.

Per questo è necessario mettere in atto delle buone prassi per il monitoraggio e gli alert: bisogna creare degli standard per il monitoraggio, implementare una strategia **Monitoring as Code**, che è una delle opportunità di lavorare in un ambiente Cloud Native. Spostarsi da un monitoraggio di livello base ("Il sistema sta funzionando?") a uno più sofisticato con dashboard avanzate ("Quali sono le vulnerabilità di sicurezza?", "Quali livelli di compliance sono stati raggiunti?") e, infine, definire cosa è urgente e qual è l'impatto di ciascun tipo di alert.

5 . ADOTTA UN APPROCCIO DI VALUTAZIONE CONTINUO E BASATO SUL RISCHIO

Per creare **avvisi affidabili è necessario rivalutare la strategia di analisi di sicurezza**, al fine di evitare di raccogliere dati in modo casuale. Alcune organizzazioni si affidano a framework come il MITRE ATT&CK, o il NIST CSF per guidare la propria strategia di rilevamento, ma spesso non è sufficiente.

Secondo Kathy Wang: «*[..] bisogna capire quali sono i principali rischi che affrontiamo come azienda in termini di dati appartenenti ai clienti, comprendere la classificazione dei dati e quindi stabilire le priorità di conseguenza [..]*».

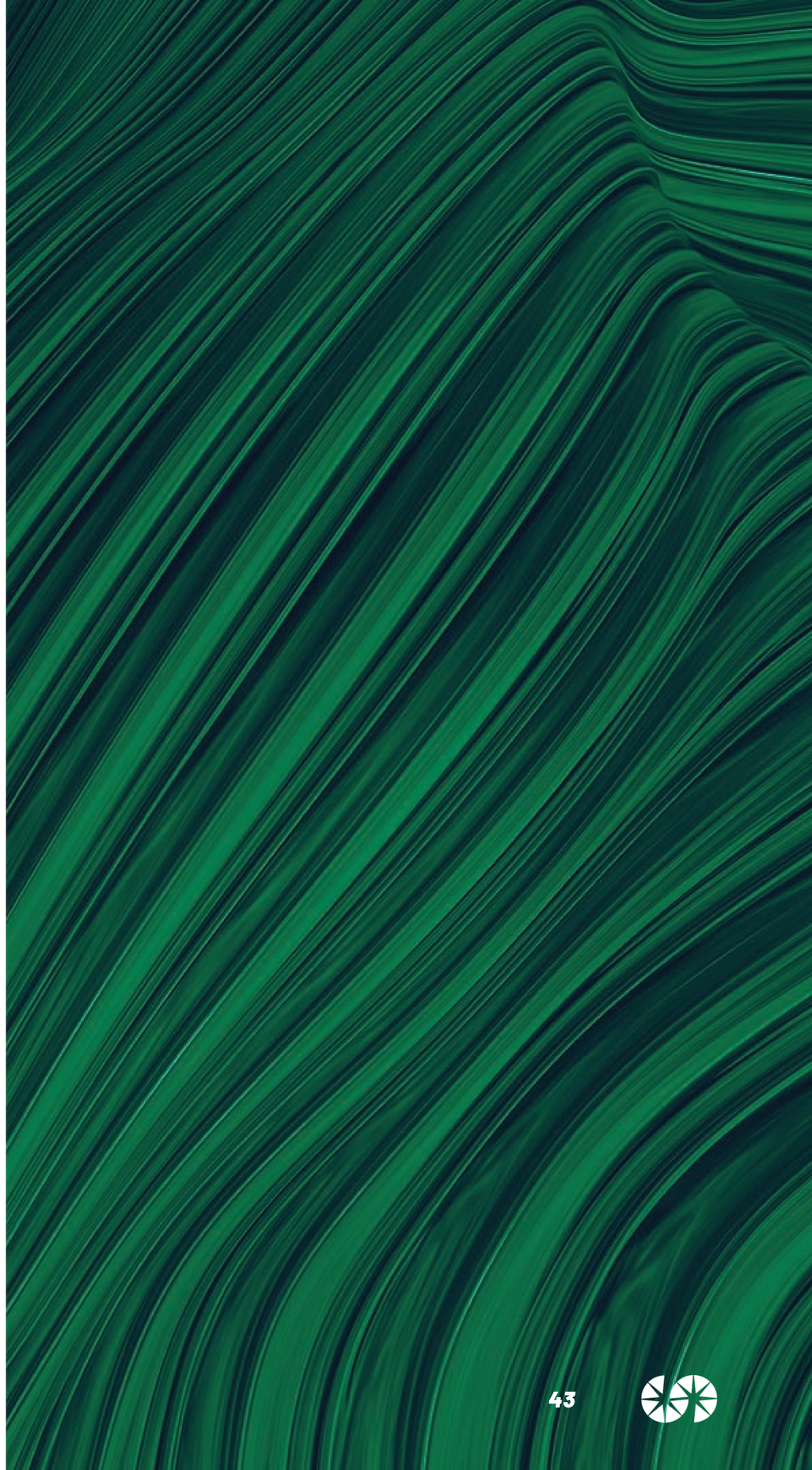
Per dare priorità ai dati raccolti e generare avvisi affidabili, alcune organizzazioni adottano una **strategia basata sui rischi e sulle minacce invece di una basata sulla copertura**. La classificazione del rischio, la gestione del rischio e la sua facilità di sfruttamento vengono utilizzate per dare la priorità ai rilevamenti. Questo approccio aiuta anche a trovare quell'equilibrio critico tra dati e prestazioni di cui abbiamo parlato in precedenza.

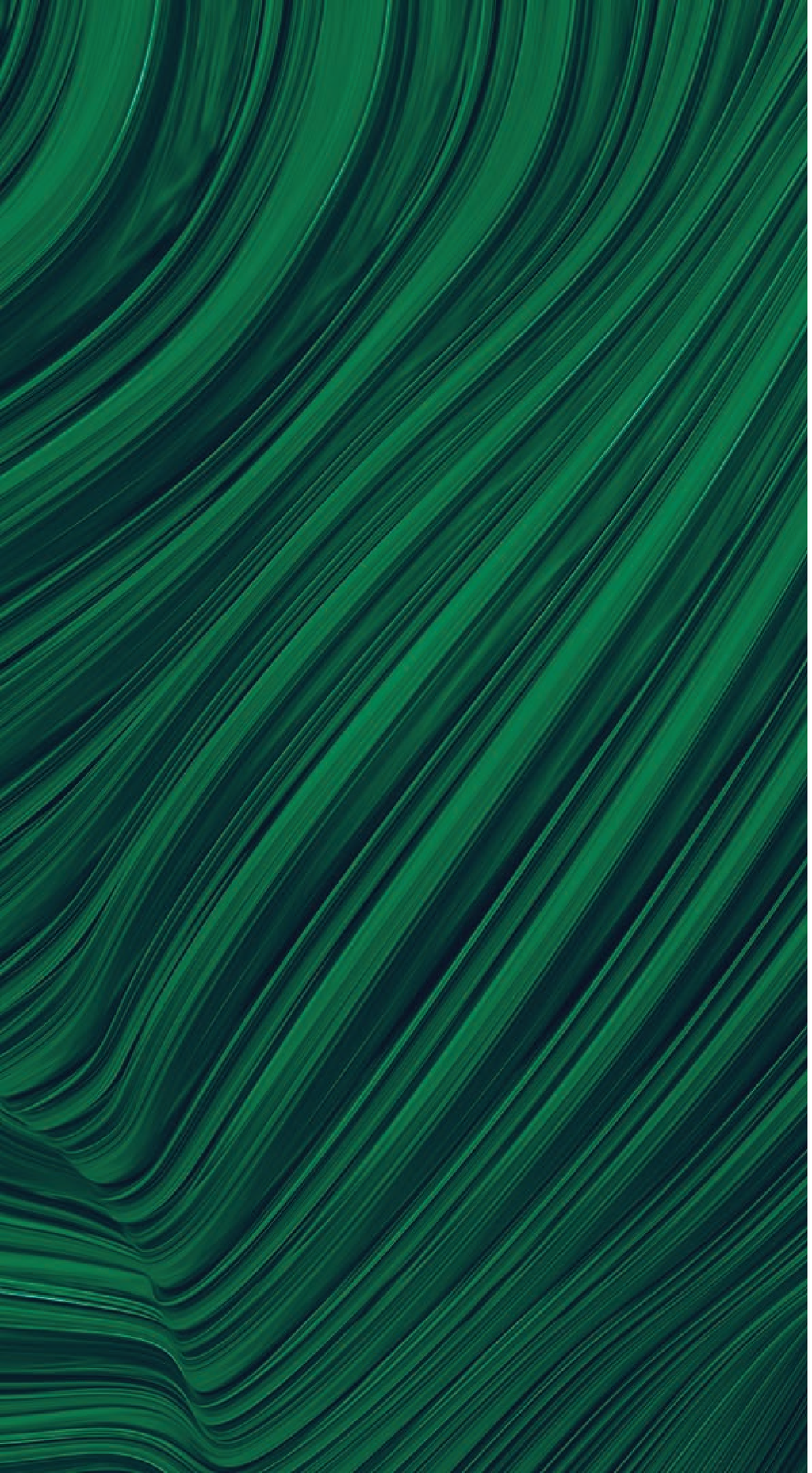
6. PRESTARE ATTENZIONE ALLA COMPLIANCE E COSTRUIRE UNA STRATEGIA

Come la sicurezza, anche la compliance è una responsabilità condivisa tra il fornitore di servizi cloud e il cliente. Per questo è necessario che l'azienda sviluppi una strategia vera e propria e non si adatti semplicemente all'offerta commerciale fatta dal fornitore di servizi cloud.

In altre parole, non basta basarsi esclusivamente sulla compliance prevista dal proprio provider cloud. È necessario investire tempo ed energie per impostare una strategia che consideri gli aspetti più critici per l'azienda.

Al Faiella, Direttore della sicurezza informatica di Unqork, suggerisce di adottare un **duplice approccio** in cui si valutano i requisiti di compliance non solo in base a ciò che la legge richiede di fare, ma anche in termini di rischi reali per la propria attività.





7. MIGLIORARE LA RESILIENZA SIGNIFICA COSTRUIRE UNA CULTURA DELLA RESILIENZA

Ribadendo la necessità di costruire una cultura della resilienza informatica, Nick Espinosa, Chief Security presso Security Fanatics ha osservato che «*ci sono molte cose che si possono fare per migliorare la resilienza informatica, ma il problema più grande che abbiamo è la cultura della sicurezza*». **Ancora una volta, la sicurezza non è solo un problema dei Team di Security**.

Il processo di identificazione e valutazione del valore delle risorse chiave all'interno dell'organizzazione fornisce un denominatore comune per parlare dei punti deboli della sicurezza tra diversi team, giungere a un consenso collettivo e elaborare un piano efficace e attuabile.

CONCLUSIONI

Riassumendo gli insegnamenti del CNSS 2021 e i concetti chiave di questo white paper: **la sicurezza è un approccio, non solo un obiettivo da perseguire**. Un approccio che deve essere necessariamente condiviso, che richiede organizzazione e collaborazione tra i team di sicurezza e DevOps.
Quanto è matura l'adozione di strategie volte a migliorare la sicurezza delle tue applicazioni nel mondo Cloud Native?

I problemi di sicurezza come malware, configurazioni errate, API non sicure, vulnerabilità dovute alla mancanza di patching controllato e periodico, credenziali erroneamente accessibili sono più o meno critici per il funzionamento del tuo business?

I team di Sviluppo, di Operations e di Security della tua organizzazione condividono un approccio security-first?

Sono solo alcune delle domande cui ci auguriamo tu possa dare una risposta ora. Se così non fosse, ricorda: in tema di sicurezza **porsi le giuste domande vale più di una risposta**, destinata comunque a cambiare nel tempo.

Rimane una certezza: portare questi interrogativi in azienda e favorire la cultura della sicurezza è oggi indispensabile per costruire basi solide di un'organizzazione duratura e di successo.

SPARKFABRIK