sysdig SECURE EVERY SECOND.

POINT OF VIEW PAPER

Practical Cloud Security Guidance in the Era of Cybersecurity Regulation

Table of Contents

Introduction

Organisations face a dual challenge: meeting regulatory requirements across diverse jurisdictions whilst innovating to deliver secure, reliable, and compliant services. As the digital landscape evolves, regulatory requirements such as those outlined in the European Union's (EU) Network and Information Systems (NIS) 2 Directive and Digital Operational Resilience Act (DORA) become increasingly stringent. This regulatory landscape necessitates a proactive approach to ensure the highest security and reliability standards. Cloud service providers (CSPs) and their clients understand that mere compliance is not enough; true security encompasses a proactive, continuous commitment to excellence.

Leaders face an equally daunting task: ensuring alignment with diverse regulatory frameworks across multiple jurisdictions. The SUNBURST incident from 2020, which was renewed by the U.S. Securities and Exchange Commission (SEC) complaint against SolarWinds in 2023, underscores pervasive security vulnerabilities across industries. The incident highlights ramifications of digital supply chains and the potential impacts on critical infrastructures. Merely citing adherence to standards such as International Organisation for Standardisation (ISO) 27001, National Institute of Standards and Technology (NIST) 800-53, NIST Cybersecurity Framework (CSF), Common Criteria, and others is insufficient. Regulatory bodies and enforcement agencies demand comprehensive validations and transparent disclosure of security gaps, while enforcing penalties upon entities failing to meet these standards.

Organisations must adopt a proactive approach to security validation and evidence capture to navigate this complex terrain of overlapping regulations. Compliance is no longer a static exercise but an ongoing commitment to maturity and resilience. As security threats evolve, so too must our practices. Maturity levels vary widely across organisations and industries, necessitating contemporary and effective strategies for continuous improvement.

This analysis delves deep into the intricacies of the digital product life cycle within cloud and cloud-native architectures and connects the dots to cybersecurity regulatory requirements. The guidance offers a pathway toward effective security, enhanced reliability, and unwavering adherence to the right side of the law. It also functions as a strategic blueprint designed to elevate the quality and security of products and services while navigating the complex web of regulatory frameworks across jurisdictions.

> Compliance is no longer a static exercise but an ongoing commitment to maturity and resilience.

From Compliant to Secure and Resilient

To guide you through this regulatory journey, we developed simplified guidance based on Sysdig customer usage patterns, newer cybersecurity methodologies like **Sysdig's 5/5/5 Benchmark**, modern approaches to policy as code (PaC) and compliance as code (CaC), and industry best practices.

Understanding the current state

Cloud environments have technological and operational properties that render traditional compliance and risk management approaches obsolete. Yet the compliance and risk management worlds still struggle to bridge the gap with the engineering world, including DevOps, SecOps, and DevSecOps practitioners, who build modern products and make services functional and reliable. Here are some common challenges.

- **Levels of regulatory compliance across jurisdictions vary greatly and affect services and teams.** Digital services need to meet new expectations. For example, the EU NIS 2 Directive has specific requirements on incident reporting and introduces criminal liability for management in case of failure to comply. In contrast, the U.S. SEC cybersecurity disclosure rules focus on the property of materiality when determining if an incident must be disclosed and aim squarely at investor protection. Global organisations must update their systems and services to maintain trust, security, and quality. These changes happen quickly, so understaffed groups struggle to stay on top of updates and make necessary adjustments to protect their services. This leaves less time and resources to address potential problems on top of existing hazards.
- **Lack of visibility and automation challenge security and compliance management.** Modern operations usually include multicloud environments, crowded software supply chains, and sector-specific needs. Isolated teams and excess tools lead to strained collaboration, delayed issue identification, and increased downtime. Manual workflows, unanswered questions, undefined roles, and poorly understood tasks impede issue resolution and process enhancement. Silos contribute to critical issues being overlooked, amplifying the risk of security breaches and compliance violations.
- **Technical teams and compliance functions need more fluid integration.** Modern technology stacks are complex and sprawling, making alignment with traditional compliance approaches difficult. With growing regulatory pressure, multiple regulating bodies must be notified at different times. Organisational teams may or may not be concerned with a given regulation depending on the criticality of a provided service or the jurisdiction it operates within. This friction hinders cross-team collaboration and is a tremendous challenge to security, operations, and engineering teams, which see compliance as getting in the way of their work. In contrast, compliance teams struggle to understand whether requirements are met adequately. The lack of integration creates a larger attack surface, making it the weak link of the organisation and exposing it to potential sanctions.

Cloud security hygiene is a strategic imperative

At the heart of our approach lies a commitment to proactive risk management and regulatory compliance for cloud and cloud-native operations. Our guidance is a roadmap for organisations seeking to fortify their defences and uphold the highest security and reliability standards.

We have surfaced details based on requirements across four major regulatory frameworks and national cybersecurity strategies: the EU's NIS 2 Directive and Cyber Resilience Act (CRA), the U.S. SEC cybersecurity disclosure rules, and the U.S. National Cybersecurity Strategy (NCS). These requirements build on traditional cybersecurity hygiene and risk management obligations and introduce a few novelties, expanding the scope of rules with which organisations must comply. With respect to the language of "incident disclosure," we do not discern between public notifications (as with data breaches) and disclosures to regulatory bodies. The regulations typically speak to both, and terminology gets overloaded.

	U.S. SEC rules	**U.S.** NCS	**EU** NIS 2	**EU** CRA
Build	○	●	◒	●
Deploy	○	◒	◒	●
Run and scale	○	●	●	●
Supply chain	◒	●	●	●
Vulnerability management	●	●	●	●
Incident disclosure	●	◒	●	●
Governance and management	●	●	●	●
Upskill	◒	●	●	●

● Addresses directly ◒ Addresses indirectly/implicitly ○ Does not address

We have identified five areas of focus for organisations, all of which are important elements of a cybersecurity strategy. Organisations will address them based on their own risk priorities and expertise in given areas, and there is no inferred level of importance. Because of the uniqueness of hardware implementations of Internet of Things (IoT) and operational technology (OT) services, organisations must make appropriate adjustments based on their use cases and environments. Software aspects are mostly universal although exceptions arise, such as specific programming languages or disparate artefacts to contend with. We'll be diving into some deeper technical specifics that might feel like whiplash if you're in a leadership role as opposed to an engineering role. However, the details are important for distinguishing a robust cybersecurity program from merely a compliant program.

Realising tangible benefits

By adopting our guidance, organisations realise benefits beyond mere regulatory compliance:

- **Enhanced security**
  Strengthening defences against cyberthreats and unauthorised access through proactive risk management and robust security protocols.
- **Improved reliability**
  Bolstering service reliability and uptime by identifying and mitigating potential points of failure and ensuring robust disaster recovery mechanisms.
- **Increased program defensibility**
  Aligning with industry best practices, open standards, and regulatory standards to create transparency and foster trust among users and stakeholders while mitigating legal and reputational risks.

Securing the Cloud in the Era of Cybersecurity Regulation

Each of the five prioritised areas outlined in the following guidance is designed to enhance your organisation's cybersecurity resilience, reliability, and compliance. This includes establishing a programmatic approach to security control requirements, leveraging secure delivery methods and documenting bills of materials (BOMs) to reduce supply chain risk, realising threat detection and response (TDR) capabilities to meet disclosure deadlines, and advancing security testing through modernisation and expansion.

Let's look at some concrete examples for each focus where we highlight the relevant regulatory language, offer prescriptive guidance, cover common pitfalls, and present core discussion points to consider when strategising with leadership.

Use programmatic approaches to ensure quality and resiliency

Programmatic approaches make it possible to implement security control requirements consistently, help avoid fragility, and enable better auditability. Such approaches also ensure that deviations from the expected secure design are kept to a minimum and are continuously detectable, rather than just during point-in-time audits. These aspects are particularly important as regulatory frameworks impose supervision and enforcement, inviting entities to demonstrate compliance.

NIS 2

Member States will supervise essential and important entities to ensure they comply with NIS 2 requirements (Art. 31-37). Through a risk-based approach, designated national authorities will establish the order in which entities will be supervised. Such examination will thus require a demonstration of compliance (audit trail, etc.).

CRA

Article 23 mandates comprehensive technical documentation for products with digital components in the EU market, ensuring compliance with Annex I requirements, updated throughout the support period. Chapter V assigns Member States market surveillance authorities to enforce regulations effectively. These authorities must receive accessible data in understandable language to assess product conformity with Annex I, covering design, development, production, and vulnerability handling.

Use "as code" approaches to achieve and maintain adherence to standards and rules

Although "as code" might be a common term for DevOps engineers, it's yet to become pervasive in the security and compliance realms. Some practitioners still avoid the discipline of coding, which was historically the world of software developers. Skill sets are advancing, though, particularly in cases where developers take on security roles or security practitioners use tooling to fill in knowledge gaps. Additionally, Generative AI has come into favour as a useful tool for creating reasonably functional code without ever needing to learn the art of programming. Compliance teams also increasingly recognise the value of codifying security requirements and translating them into forms of code.

PaC approaches enable the introduction of specific internal or external rules, while CaC is more broadly the "code version" of the entity's regulatory obligations.

Hardened baselines should be established and implemented in the preferred infrastructure-as-code (IaC) format. Common IaC tools and formats include Chef, Puppet, Ansible, and Terraform, but you may also need to work with cloud-specific IaC formats like AWS CloudFormation or Azure Resource Manager (ARM) templates. Hardened baselines are readily available in most of these common formats. Alternatively, you could harden defined resources against benchmarks like the Center for Internet Security (CIS) or Defense Information Systems Agency (DISA) Security Technical Implementation Guides (STIGs). It's possible to restrict access to code via version control systems, as seen in GitOps practices. Moreover, infrastructure automation tooling may offer specific capabilities to control code changes to reduce misconfiguration risk. As EU Member States publish additional technical guidance, tuning the IaC accordingly will help ensure hardened baselines that satisfy the relevant regulatory requirements.

Newer concepts in this area are PaC and CaC. PaC approaches enable the introduction of specific internal or external rules, while CaC is more broadly the "code version" of the entity's regulatory obligations. Separating the policies from IaC provides flexibility and separation of duties that are necessary, since multiple teams touch code definitions, dependencies, or workloads throughout the life cycle. PaC aims to avoid policy violations rather than to identify them. This approach uses DevOps automation features, eliminating the need for manual processes. As a result, teams can work faster, and there is less chance of errors caused by human intervention. If we were to set a division of labour, we could highlight that PaC enhances security operations, compliance management, and data governance, while IaC focuses on infrastructure and provisioning.

One such example is the Open Policy Agent (OPA), which enables infrastructure teams to control workload instantiation and operation based on separate policies in code format (Rego in the case of OPA). Kubernetes, the de facto container orchestration engine, also provides similar mechanisms with YAML and JSON manifests and the concept of admission control. The two can also work in tandem, where OPA Gatekeeper is deployed as an admission controller in Kubernetes. Vendor-specific implementations include Styra OPA, HashiCorp Sentinel, Red Hat Ansible, or Chef and Puppet languages. Open source implementations such as Chef InSpec and OpenSCAP can get you halfway there, but they require additional engineering effort in enterprise settings.

Although there may seemingly be overlap between PaC and CaC, the two fundamentally differ in the standards they aim to enforce. CaC helps with enforcing regulatory requirements, while PaC can enforce any organisational policy. The **ComplianceAs-Code tool** is an open source security solution that companies can use to collaborate and develop additional capabilities. Created by commercial vendors and government agencies to make the Security Content Automation Protocol (SCAP) more accessible, it has since grown to include different industry standards. The tool also accommodates automation tooling. CaC and PaC are two approaches that work together in DevOps to allow the early integration of compliance in a continuous integration/continuous delivery (CI/CD) pipeline. For companies working in highly regulated environments, using both approaches is ideal.

CaC and PaC are two approaches that work together in DevOps to allow the early integration of compliance in a CI/CD pipeline. For companies working in highly regulated environments, using both approaches is ideal.

“Drift” is the term used to describe when an implemented system deviates from the intended (often hardened) design or configuration. Drift detection and drift control should also be high on the organisational priority list to help identify when discrepancies and risky misconfigurations arise. The former helps identify when drift occurs, and the latter helps prevent drift from occurring at all. Seek out these capabilities within infrastructure automation and cloud security tooling. You can take action on deviations, such as alerting teams to the problem or terminating affected vulnerable services. Realistically, not all organisations operate an immutable infrastructure, at least not universally. Workloads may change normally as a byproduct of other build and delivery pipeline elements, particularly when introducing third-party dependencies.

What can go wrong?

Infrastructure hardening is the easy part. Realistically, some services (such as HTTP Port 80 and HTTPS Port 443, common in web applications and API designs) need to be open to support business needs. This reality stresses the importance of robust access controls for humans and machines, not just from the network perspective with network access controls and firewalls. Simply closing or denying service on a port impedes the business and likely breaks application functionality.

Interservice communications and integrations may also break. We see this often when taking least privilege and zero trust too far. Organisational teams may be inclined to leave permissions more relaxed to avoid creating availability problems. The result is that identity permissions and cloud entitlements are frequently overpermissioned and unused, elevating security risk, which was echoed in **Sysdig's 2024 Cloud Native Security and Usage Report**. Access control decisions must be dynamic and consider legitimate business use, shifting operating environments, and mixed identity types.

Access control decisions must be dynamic and consider legitimate business use, shifting operating environments, and mixed identity types.

Identity and access management, authentication, authorisation, privileged access management, remote access (including virtual private network [VPN] and zero-trust network access [ZTNA]), secrets management, and more are all integral, but they are beyond the scope of what we can cover here. When regulators mention "access control," you should know that all of these technical elements are in scope.

Traditional infrastructure and data centre environments are likely not definable fully or at all with "as code" approaches. Organisational teams will find that they require other augmenting technology like OpenSCAP or Simple Network Management Protocol (SNMP) to monitor, manage, and secure all types of infrastructures. Distributed, heterogeneous environments make rationalising secure baselines incredibly difficult. Supply chains also exacerbate the problem, since end-to-end functionality to enable a business outcome often involves multiple suppliers and partners. And each of these entities has unique technology stacks.

Most practitioners start with open source software to solve problems, particularly when budgets are tight. Unfortunately, open source software support of security tools can be spotty. Developers and engineers voluntarily contribute, but many open source software projects lack full-time contributors. Without commercial backing, projects often lag behind what is adequate. Organisations must self-engineer using open source software building blocks or procure commercial tools to fill gaps. There is a growing tide of concern that the Cyber Resilience Act may inadvertently inhibit open source software contributions because of the liability it creates. Open source software contributors lack the resources to make regular code updates, participate in coordinated vulnerability disclosure (CVD), and report incidents as required by regulation.

To avoid some code quality headaches, it's important that organisations use open source software projects with adequate community support and vendor backing. Specifically, in the case of the Cloud Native Computing Foundation (CNCF) and to mitigate some of this risk, look for mature projects that have reached the graduation stage, such as Falco or OPA, as opposed to incubation projects. The decision to use open source software more heavily also means that your engineering teams will need to build some of the connective glue that is missing, such as risk prioritisation of findings, DevOps remediation guidance, and SecOps workflow creation. Most organisations will be better served by evaluating and selecting a commercial off-the-shelf solution for their cloud security that takes the form of a **cloud-native application protection platform (CNAPP)**.

To avoid some code quality headaches, it's important that organisations use open source software projects with adequate community support and vendor backing.

Leadership discussion points to consider

01 What is impeding the adoption of infrastructure "as code" and policy "as code" approaches within the current operating environments?

"As code" approaches ensure defensibility of the organisation's security program and enable effective internal and external audits. The code artefacts serve as supporting evidence for the organisation's risk management program and can be furnished to satisfy regulatory bodies in the event of an audit.

02 How does the organisation detect misconfigurations quickly without extensive post mortem analysis?

Configuration drifts impact cybersecurity hygiene which create windows of exposure for attackers. If a threat actor is successful in exploiting critical infrastructure or creating material impact, incidents must be disclosed to regulatory bodies within 24 hours, 72 hours, or 4 business days depending on the type of data exposed, locale of impacted users, or geographic region where systems operate.

03 How is the organisation mitigating the risk of misconfigurations?

Guardrails ensure that secure baselines are followed at scale and provide evidence to regulatory bodies of how the organisation operates its risk management program. Drift control is a proactive mechanism that prevents any deviations from hardened baselines and mitigates likelihood of incidents that need to be disclosed.

Mitigate supply chain risk with secure delivery

Software or digital supply chain risk has consistently ranked high on concerns by security leadership across industries. It's one of the largest risks to critical infrastructures because of how many physical and digital service providers make up a given supply chain. Threat actors, including authoritarian regimes, regularly target all elements of supply chains, including each entity's development, build, and release tooling. This includes development tools, package managers, version control systems, CI/CD tools, and infrastructure platforms, particularly container architectures. Simply put, there are many touch points in a given build pipeline. That number is greatly amplified when considering the entirety of the supply chain. **Secure delivery is the practice and discipline of ensuring that all build and release pipelines are adequately protected from internal and external attacks.**

NIS 2

Article 21 requires Member States to ensure that essential and important entities implement appropriate technical, operational, and organisational measures to manage risks to the network and information systems they utilise. These measures aim to prevent or mitigate the impact of incidents on their services and others based on an all-hazards approach. They include supply chain security and address relationships with direct suppliers or service providers. Article 22 empowers the NIS Cooperation Group, the Commission, and European Union Agency for Cybersecurity (ENISA) to conduct coordinated security risk assessments for critical services, systems, or product supply chains, considering technical and nontechnical risk factors.

CRA

The regulation imposes obligations on manufacturers and distributors of products incorporating digital elements. As defined in Article 1, Distributors encompass any entity within the supply chain, excluding manufacturers or importers, that places such products on the EU market without altering their properties. Supply chains must be integral to assessments conducted by market supervision authorities, as per Art. 43, which extends from Art. 22 NIS 2 (above). This provision enables authorities to evaluate nontechnical risk factors within supply chains of products with digital elements, particularly those posing significant cybersecurity risks.

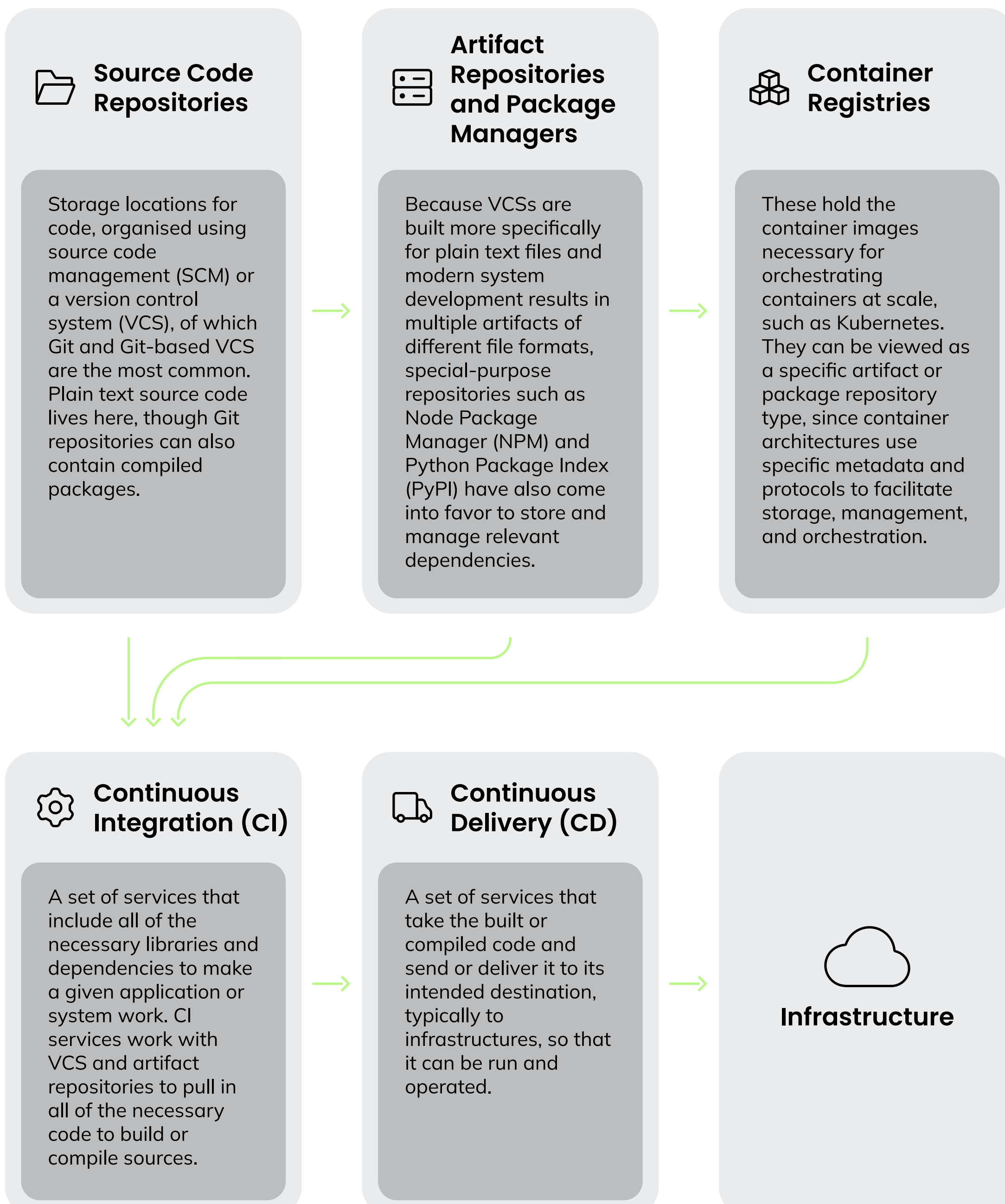
US NCS

Strategic objective 5.5 covers the security of physical and virtual supply chains. Digital supply chains are prone to attacks that put cyber resilience and public safety at risk. Critical infrastructures may be directly disrupted, or service providers to critical infrastructures, like cellular and next-generation wireless networks, can be attacked leading to disruption. Technology and service providers rely on software that must be built and delivered securely to mitigate risks to global supply chains.

Exercise care when consuming external code

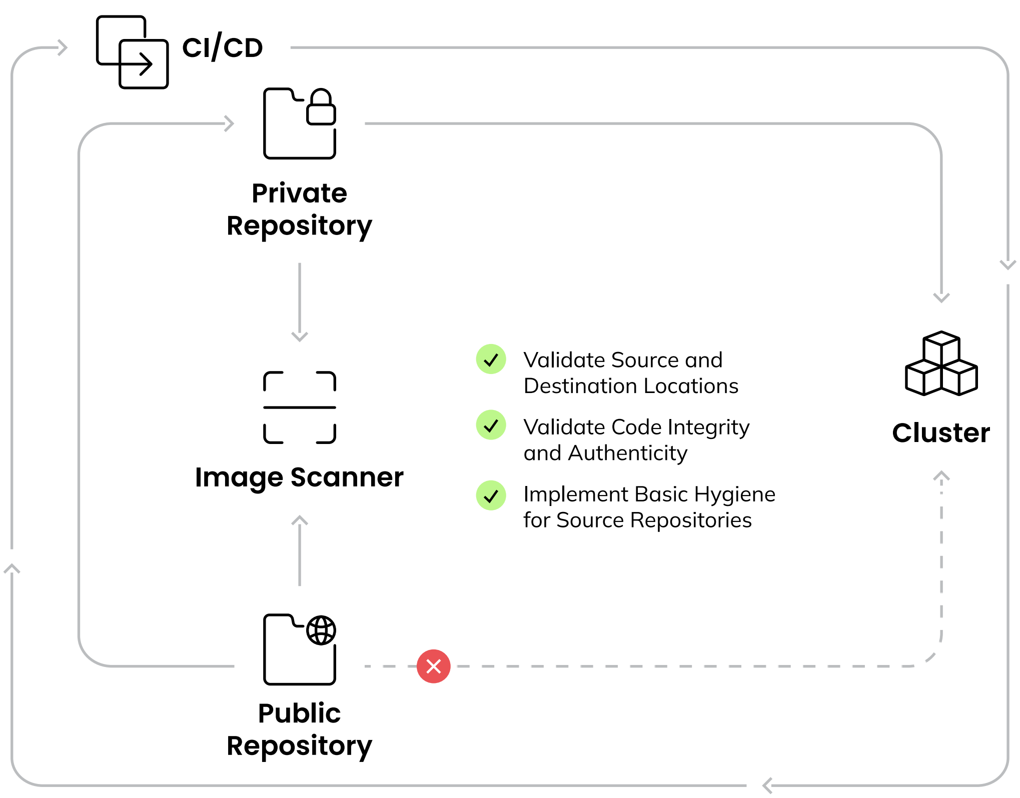
Organisations should formalise their development life cycles if they haven't already. Part of that includes establishing and standardising build and release processes. A general rule of thumb is that a specific technology stack results in a build pipeline. If your organisation builds many applications and systems, expect the number to grow dramatically. Your pipeline count might be in the tens or hundreds based on the history of development in the organisation, along with where it's headed with new technology. Some entities call this a "secure software build factory," but it is necessary to secure delivery adequately.

A typical, modern build pipeline consists of:



All of these elements require scrutinisation and adequate protection. Generally speaking, the steps you should follow are:

1. **Validate Source and Destination Locations**
   - Ensure internal or external actors do not modify targets.
   - Scrutinise and protect source and destination locations for code and infrastructure.
2. **Validate Code Integrity and Authenticity**
   - Validate code integrity using hashes.
   - Validate authors via digital signatures to ensure authenticity.
3. **Implement Basic Hygiene for Source Repositories**
   - Avoid pulling source directly from public repositories and registries.
   - Scan for known vulnerabilities.
   - Verify correct versions using dependency analysis and software composition analysis tools.
4. **Utilise Internal Sets of Proxied Private Repositories**
   - Utilise internal sets of proxied private repositories, registries, and package managers.
   - Provide a level of validation that is impossible if a connection is made directly to public sources.



Validate continuously throughout workload life cycles – specifically build, delivery, and runtime. Contributing team members may change definitions or manifests, or third-party dependencies may introduce unexpected configuration changes.

What can go wrong?

Nested dependencies complicate the practice of software composition analysis. You'll likely not have visibility into all of the dependencies within your partners and suppliers that make up the full digital supply chain, particularly if it's closed source. Best case, these providers may offer software BOMs (see next section). Realistically, they'll likely point you to their terms and conditions or a generic web page with little technical detail. Dependencies that are too nested obscure visibility and also inhibit effective scanning, making it difficult for the organisation to assess the true attack surface. Organisations often have to accept some of this risk or rely on runtime security controls to mitigate potential impacts.

Developers may not properly sign code. Modern VCSs (like Git and Git-based offerings) should provide capabilities to generate hashes automatically, but organisations also need to verify them on build and delivery programmatically. Git is also a distributed (as opposed to centralised) VCS, meaning that it is possible to copy (fork and clone) repositories anywhere. This results in copies of code in many places, making the implementation of consistent security controls in enterprises difficult and necessitating additional monitoring of development environments for highly regulated verticals.

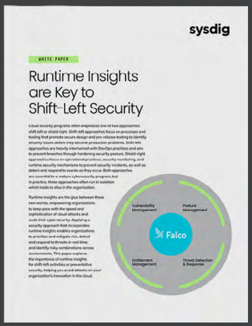
Mixed artefact types mean that you'll need multiple scanners to analyse all types of code. Even commercial offerings may not support all of the dependencies and packages you'll encounter in the organisation. Development teams may also be moving onto newer development tools and platforms, and those code artefacts may not be readily scannable.

File hashes are only one piece of the puzzle useful for verifying integrity. Validating authenticity with digital signatures requires effective key management. Signing keys or certificate pairs are types of secrets that need to be shared through trusted channels and stored in protected systems such as a public key infrastructure (PKI), key management services (KMSs), or hardware security modules (HSMs). These secrets may still be compromised, as we've seen in other attacks on major service providers. It's a problem that also arises when tackling encryption in transit, in use, and at rest, which is also necessary as per the EU NIS 2 Directive.

Entities should hedge their bets by running multiple scanners and correlating results, introducing a new type of vulnerability management problem. This leads to questions around dynamic analysis and runtime context to identify whether vulnerable code is even reachable or if a given issue is exploitable in runtime.

For more thorough coverage of how runtime context aids risk management, you can read the analysis Runtime Insights are Key to Shift-Left Security.

READ THE WHITE PAPER →



Leadership discussion points to consider

01 What security tools are integrated into delivery pipelines to cover all types of code artefacts?

Organisational teams may request funding for a tool to solve an immediate problem, but that team may be unaware of a tool's ultimate efficacy in analysing all code and identifying risks. Leadership may infer that the particular problem has been adequately addressed and possibly even attest to that stance in regulatory disclosures, when there is in fact a security gap that creates security risk.

02 How are integrity and authenticity of components being validated within pipelines?

Digital supply chain attacks take numerous forms that may target developer tooling or code repositories within or external to the organisation. To meet cybersecurity regulatory requirements, the organisation needs to continuously validate and attest to its ability to identify when code or dependencies have been tampered with.

03 What mechanisms exist that can automatically vet dependencies in builds?

Implement guardrails for engineering teams to use vetted, reasonably secure componentry, the organisation should be maintaining private registries and repositories that teams can pull from. Pulling source directly from public sources without proper scrutiny reduces visibility and amplifies the potential impacts of a digital supply chain attack.

Document digital supply chains with BOMs

The latest batch of regulations from both sides of the Atlantic frequently mentions BOMs. These digital documents outline the makeup of a given piece of software or hardware. Application and system design are heavily reliant on component and dependency reuse. No engineer develops something fully from scratch, and they readily include third-party or open source libraries in the codebase.

At a minimum, BOMs are useful for proactively identifying potential licence, quality, and vulnerability risks, all of which factor into supply chain risk. Most of the focus with BOMs has been to use them as part of security testing, particularly for package or container dependencies. Still, realistically, you can also use them as part of threat detection and system protection if tooling evolves. An organisation needs to know what goes into the hardware and software it procures and consumes so that it can also gauge the relative risk. Documenting vaguely within terms and conditions documents or licence agreements never suffices. Still, reality has hit full force because of the rates of change with code versions given agile methodologies and DevOps practices.

NIS 2

Article 21 requires Member States to ensure that essential and important entities implement appropriate technical, operational, and organisational measures to manage risks to the network and information systems they utilise. These measures aim to prevent or mitigate the impact of incidents on their services and others based on an all-hazards approach. They include supply chain security and address relationships with direct suppliers or service providers. Article 22 empowers the NIS Cooperation Group, the Commission, and ENISA to conduct coordinated security risk assessments for critical services, systems, or product supply chains, considering technical and nontechnical risk factors.

CRA

The regulation emphasises the importance of manufacturers creating and documenting software bills of materials (SBOMs) to facilitate vulnerability analysis of products containing digital elements. While manufacturers are encouraged to ensure their products do not contain vulnerable components from third parties, they are not obliged to publicise SBOMs. The regulation mandates obligations regarding SBOMs, defining them as formal records detailing components and their supply chain relationships within the software elements of products with digital elements.

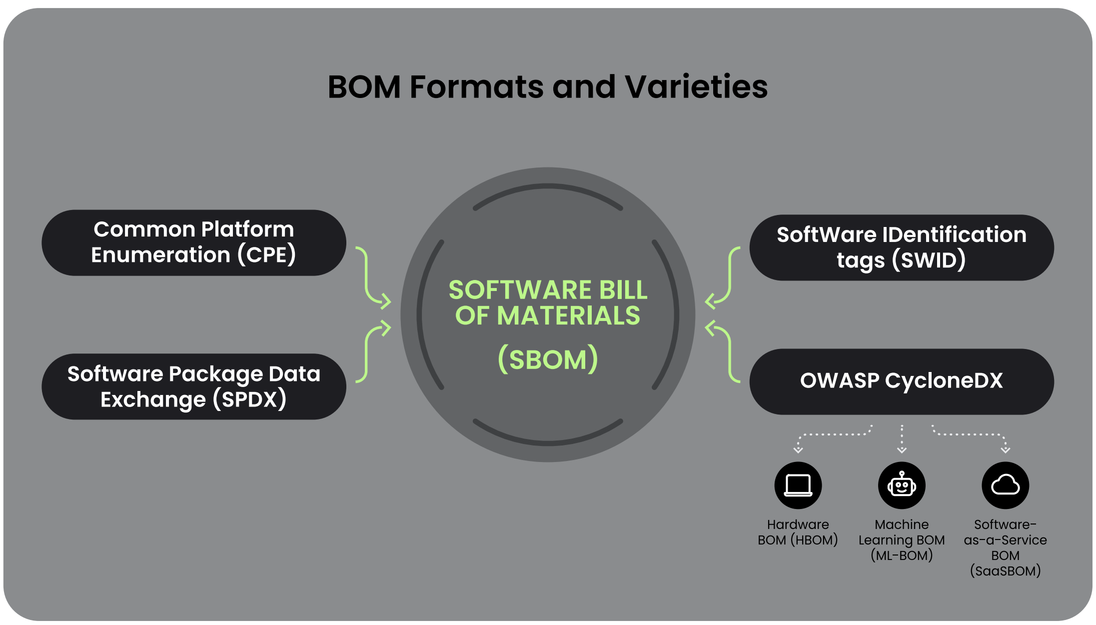
US NCS

Strategic objective 1.5 that describes the modernising of federal defences explicitly mentions SBOM efforts to help mitigate software supply chain risk, including EO 14028, “Improving the Nation’s Cybersecurity” and NIST Special Publication (SP) 800-218, Secure Software Development Framework (SSDF).

Use the right BOM for your scenario

We covered secure delivery concepts in the previous section, and BOMs are also enablers of secure delivery. Generally, a BOM is a piece of machine-readable data (such as CSV, XLS, or XML) that other tools can ingest for various use cases. BOMs may also be human-readable documents, such as for audits, but this is less useful in the digital world, since release cadences will quickly make resulting reports outdated. Dependencies can and should be validated against the BOM as they are stored within code and artefact repositories. This is useful for bolstering supply chain security and validating what suppliers attest to. The code and its composition should match what was described and what the organisation expected to procure.

Because of the diversity of hardware and software makeup, different BOM formats are necessary. Software BOMs (SBOMs) are the most well known and should be the starting point for most organisations. They only cover well-known components and are typically open source, but it is still a worthwhile endeavour. There are a handful of competing formats that include **Common Platform Enumeration** (CPE), Open Worldwide Application Security Project (OWASP) CycloneDX, Software Package Data Exchange (SPDX) and **software identification tags** (SWIDs). Within CycloneDX, there are also a variety of BOM types, including a **hardware BOM** (HBOM), **machine learning BOM** (ML-BOM), and **software-as-a-service BOM** (SaaSBOM), to account for the uniqueness of each of those target devices, code, or services and their dependencies.



Using the appropriate BOM for each use case and scenario ensures that you'll be prepared for audits from internal compliance teams or regulators. It is important to continuously maintain and dynamically generate BOMs, since all applications, services, and systems change regularly. BOMs are useful as digital evidence that your organisation is building, consuming, or operating what you describe. They can also be furnished to partners and suppliers to bolster the security of the complete digital supply chain.

What can go wrong?

It should be painfully apparent that BOM formats are unsettled. There is no one format to rule them all, and regulatory language doesn't spell out which format to use. As a result, tooling support for BOMs is still nascent, despite regulatory pressure and emphasis on national cybersecurity strategies around supply chain risks. CycloneDX is emerging as a de facto standard in application security and cloud security. However, CPE is still highly relevant in vulnerability management and CVD. Also, it's common to see SPDX in cases where legal teams are involved or there are risk concerns over open source licensing such as with **BSD+Patent** licences.

BOMs also quickly get stale because of regular changes to applications and systems. Partners and suppliers may not be doing the work to identify the dependencies they use and generate BOMs regularly. BOMs may also not be furnished to you, burdening you as the consumer. You need the entirety of the supply chain in order to understand the true risk to the organisation, and threat actors target the weakest link.

If you depend on BOMs to identify known vulnerabilities, recognise that all code exhibits flaws, including dependencies provided by commercial entities and open source projects. You might stress this with auditors, but you will need to adjust your risk appetite since you will be continuously wrestling with bugs and vulnerabilities of different severities. Risk prioritising based on other factors such as internet exposure, data sensitivity, and business criticality is not just advisable but encouraged within the NIS 2 language. Risk prioritisation enables an economical approach and a more effective cybersecurity approach, since time and resources are always at a premium.

Risk prioritising based on other factors such as internet exposure, data sensitivity, and business criticality is not just advisable but encouraged within the NIS 2 language.

To properly assess risk, organisations would need to understand all pieces of code and the related dependencies, like API connections or infrastructure elements. This level of visibility isn't supported within one BOM format today, and such a BOM also wouldn't be readily parseable by one scanner. Vendors have been slow to implement capabilities around BOMs because of a lack of clarity from regulators as to what's truly necessary, or what customers require to secure all of their technology.

Most code relies on other dependencies, referred to as transitive dependencies, and a complete map branches out quickly. You can quickly reach a point where you can't feasibly verify all dependencies quickly, or at all, at least not at build time with frequent release cadences. Dependency analysis requires an immense amount of computing power to analyse in real time and is only accurate for that complete system. Additional mitigating controls or runtime protections are often necessary given the lack of visibility.

Leadership discussion points to consider

01 How is the organisation verifying materials within procured hardware and software?

Finance and legal teams should be engaged so that contract language can be revised to mandate that BOMs be furnished by suppliers. Such digital documents are useful for understanding what the organisation is consuming, assessing the true attack surface, and identifying where digital supply chain risk may be inherent. Existing contracts may only require static disclosures which grow stale quickly and will not satisfy cybersecurity readiness requirements.

02 What BOM formats has the organisation standardised on?

The organisation likely serves as a supplier to others, even if it's just direct consumers. It also needs to maintain and furnish BOMs to ensure and bolster supply chain security. BOMs should also be appropriate for the software or system element you're describing, as an SBOM alone will not cover all aspects of the architecture. Engineering teams should select a standard or collection of standards, and ensure that those standards are appropriately documented in regulatory disclosures around the risk management program.

Modernise TDR to meet incident disclosure deadlines

Regulatory requirements are becoming more stringent: in the EU, the NIS 2 Directive and the CRA require organisations to report security incidents within 24 hours. In the U.S., the Cyber Incident Reporting for Critical Infrastructure Act (CIRCIA) will mandate notification to the Cybersecurity and Infrastructure Security Agency (CISA) within 72 hours, and the SEC mandates disclosure of material incidents within four business days. However, meaningful incident detection, response, and disclosure need more than just dumping event data into the organisation's security information and event management (SIEM) or security data lakes. Although the troves of security event data may help for periodic audits, they do little in the event of a nation-state or fast-moving threat. Modern architectures are dynamic and highly ephemeral, and organisations **are compromised in minutes, not days**.

As workloads live for only a few minutes before being terminated and reinstantiated, relevant data to support digital forensics and incident response may never be captured, let alone ingested into a SIEM. This situation makes it challenging for organisations to comply with regulatory requirements for faster detection and incident disclosure. In response, modernisation of threat detection and response (TDR) is necessary to ensure that the security operations centre (SOC) doesn't drown.

NIS 2

Article 21 mandates Member States to ensure essential and important entities implement appropriate technical, operational, and organisational measures to mitigate risks to network and information systems, considering state-of-the-art standards and implementation costs. Measures must align with the entity's risk exposure, size, and the likelihood and severity of incidents, adopting an all-hazards approach encompassing incident handling. Article 23 outlines notification obligations for national and cross-border incidents, requiring entities to promptly inform CSIRTs or competent authorities of significant incidents potentially affecting service provision. Notifications must occur within 24 hours of detecting suspicious activity, with official reports detailing impact, technical information, and incident summaries within 72 hours and one month, respectively.

CRA

Recital 35 underscores the importance of manufacturers promptly notifying designated CSIRTs and ENISA about severe incidents affecting product security. Manufacturers may also inform users about such incidents and any possible corrective measures. Article 11 mandates manufacturers to report severe incidents simultaneously to the designated CSIRT and ENISA. Severity of incidents are defined by their impact on product security. In line with NIS 2 notification timelines, the CRA requires initial notification within 24 hours of awareness and official notification within 72 hours detailing impact, severity, and mitigation measures. A comprehensive incident report is due within one month after incident resolution.

US NCS

Objective 1.4 covers updates to federal incident response plans and processes including public and private sector collaboration. Covered entities in critical infrastructure sectors must disclose cyber incidents promptly to CISA. Objective 1.5 calls out the Office of Management and Budget (OMB) zero trust architecture strategy, directing Federal Civilization Executive Branch (FCEB) agencies to gain visibility into their entire attack surface and adopt cloud security tools, among other technology approaches. Objective 2.3 describes how intelligence sharing needs to be accelerated and expanded to disrupt cyber attack campaigns. This includes warning and notifying victim, or potential victim, organisations so they can adequately prepare for cyberthreats.

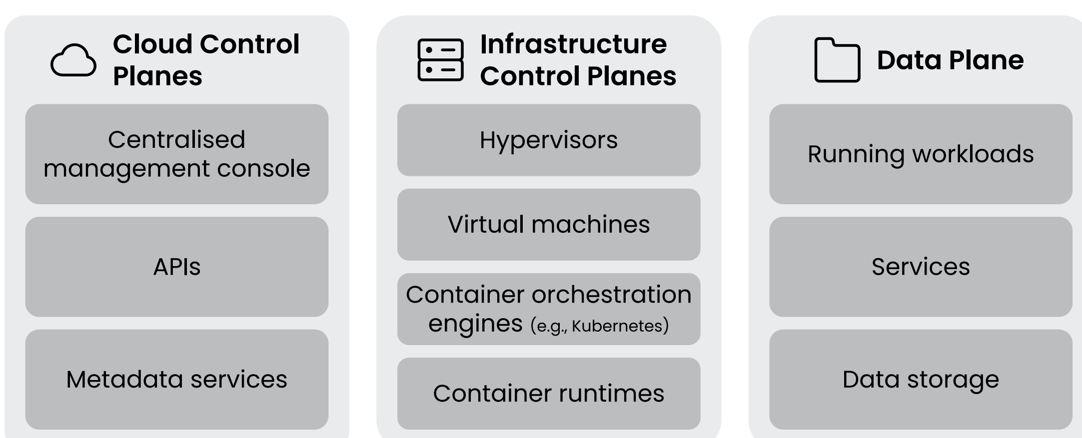
US SEC

Form 8-K Item 1.05 requires that registrants disclose any material cybersecurity incident, the related material impacts from the incident, and the material aspects including nature, scope, and timing of the incident. The materiality determination must be made “without unreasonable delay,” and the registrant must file a Form 8-K within four business days of such determination. Similarly, foreign companies must disclose material incidents in Form 6-K if they fit the definition of a foreign private issuer. Explicit technical details need not be disclosed since it could hinder response and remediation efforts.

Gather, correlate, and analyse signals across environments

Organisations need to embrace newer approaches for faster detection and incident response. In the cloud, this requires signals from a variety of sources that extend beyond expectations of traditional data centre environments. The sources identified below provide signals for operating cloud services, data storage, workload resources, and more.

Cloud-Native Architecture: Planes of Existence



Log collection doesn't look like it used to. Cloud infrastructure is distributed, elastic, and ephemeral, and cloud-native technologies introduce new, additional log sources such as CSP trails and orchestration engine logs. The data plane and its running workloads will generate system and application logs, but account for cases where workloads may be short-lived. It's important to collect relevant system calls and pre- and post-event telemetry to support digital forensics and incident response. Unless the organisation has architected things to look more like a traditional data centre environment (lift and shift), anticipate that you'll be contending with various new data sources for security events. In reality, most organisations use a hybrid cloud model, getting the best (and worst) of both worlds. Organisational teams will need to rationalise log and event data to meet regulatory requirements and disclose incidents as appropriate, on time, to the right regulatory bodies.

Seeding TDR capabilities with runtime context will give you a risk-prioritised view of the entire operating environment and improve incident response times. All of these signals need analysis in real time or near-real time (as CSPs only guarantee service to a certain point) to provide the level of TDR and digital forensics and incident response (DFIR) needed to meet incident disclosure requirements. There will be no shortage of security events in any environment in any organisation in any vertical. SOCs need to ensure that signals are useful so they're not chasing immaterial or noncritical incidents that burn resources.

Detection "as code" is gaining favour as a way to improve and automate detections without the need for manual review of event data.

It's also wise for SecOps teams to begin adopting the "as code" mantra like other organisational teams so that security can keep pace with the speed of development and release. Detection "as code" is gaining favour as a way to improve and automate detections without the need for manual review of event data. Detections can and should be written in code form that can be further divided among subject-matter experts of certain attack patterns and technology domains. The process can look a bit like GitOps (to borrow a term from modern infrastructure operations), where Git-based VCSs store and maintain code. In this case, however, the code comprises behaviour-based or rule-based detections. Forrester has taken to calling these sets of TDR-focused processes the **detection and response development lifecycle (DR-DLC)**. These detections can also be furnished to auditors and regulators as supporting evidence for the organisation's security protections and incident response process.

What can go wrong?

Application platforms and container platforms use a mix of technology, and data formats are unlikely to be universal. This topic can rabbit hole into the flavours of platforms and platform as a service (PaaS), but the simplest explanation is that there are opinionated platforms and (open) standards-based platforms. Opinionated platforms emphasise developer- and user-friendliness at the expense of high security. Even container platforms may not use the Open Container Initiative (OCI) format, or what many simply refer to as Docker. They may instead offer some hybrid that adds further uniqueness and complicates security operations. Rarely is a platform both user-friendly and high security. Log formats may also use a standard like SysLog, but it may not contain all of the relevant information needed to reconstruct an incident.

Cloud logs may not be enabled for all services, which is common in Kubernetes deployments because of the high volume of container traffic that gets generated and eats up further resources. Adequate computing power is also needed to analyse all signals fast enough to identify potential security incidents. This has been a contentious topic within national cybersecurity strategies and the burden that it places on CSPs. Who owns the cost of the necessary data storage and computing power to furnish proper log detail? Regulators and customers want to push this burden to CSPs. CSPs want it to be a customer choice (and for them to assume the cost). Regardless of how the regulatory landscape plays out, ensure that logs are enabled in all elements of the environment and at a sufficient level of detail.

The signal-to-noise ratio quickly becomes a problem for many organisations. Organisational SIEM and endpoint detection and response (EDR) deployments are likely overloaded with too much data regarding endpoints or all types of potential security events. This operational choice makes it difficult to identify application- or cloud-specific threats quickly. SecOps modernisation efforts often emphasise reducing the number of log feeds and signals, not adding more, to make the SOC more effective.

Timeliness of incident detection, disclosure, and remediation remains a high bar when factoring in all of the types of security incidents that can occur within an organisation. Incidents include ransomware attacks and (email) phishing attempts, not all of which may occur within cloud and cloud-native environments. Assessing business impacts, privacy impacts, and material impacts involves other teams outside of security and IT. Events also unfold over time. What was once nonimpacting or immaterial may become so at another point in time when issues materialise or attackers tune their efforts in attack campaigns.

Leadership discussion points to consider

01 How are security operations teams codifying threat detections?

Detection "as code" allows security to keep pace with development. TDR vendors should ideally support generation of the code within a given platform, but security teams may also need to educate themselves on interpreting and maintaining the code. The organisation's process for generating detections should be documented within regulatory disclosures since it supports timely incident detection and disclosure.

02 How quickly can the organisation identify a security event and gather relevant signals?

Cybersecurity disclosures to regulatory bodies may be necessary within 24 hours of an incident. The organisation needs to be able to quickly correlate event signals from all its on premises and cloud environments. Advanced threat actors are known to compromise distributed environments in under 10 minutes, so timely detection and response is critical.

Evolve your vulnerability management

All products and services exhibit bugs or vulnerabilities over time. Systems may also exhibit exploitable conditions because of misconfigurations or a lack of other mitigating security controls.

How to handle vulnerabilities differs widely, depending on who identifies them and who is responsible for a vulnerability-free product. Ideally, finding a vulnerability in published software or produced hardware triggers a vulnerability disclosure and documentation process. However, vulnerabilities are also disclosed “in the wild” outside a clear process for handling, producing, and deploying a patch. This approach, referred to as CVD and described in ISO 29147 and ISO 30111, prevents disclosure in the wild and provides a safe harbour to security researchers identifying and reporting the vulnerabilities.

Regulatory requirements on incident disclosure bring much of CVD into scope. By implementing notification timelines and receiving coordinator entities, the EU NIS 2 Directive and CRA establish CVD mechanisms at the national and European levels.

NIS 2

Article 12 focuses on establishing a coordinated vulnerability disclosure (CVD) framework and an EU vulnerability database, akin to the EU’s version of the National Vulnerability Database (NVD), operated by ENISA. ENISA can collaborate with third-party managed databases such as the CVE one. Each Member State must designate a CSIRT as a coordinator for CVD, serving as a trusted intermediary between vulnerability reporters and ICT product or service providers. The designated CSIRT negotiates disclosure timelines, while safe harbour provisions are implicit. Additionally, the NIS Cooperation Group aids Member States in developing national strategies encompassing CVD and, ideally, safe harbour policies.

CRA

Art. 11 mandates manufacturers promptly report any actively exploited vulnerability in their products with digital components to the designated CSIRT and to ENISA through a single reporting platform. The reporting includes an early warning, a vulnerability notification with assessment of severity and impact, and a final report within specific timeframes. The vulnerability notification process mirrors the incident notification process with specific timeframes. Lastly, users must be swiftly informed about vulnerabilities or incidents, with structured mitigation information.

US NCS

Much of objective 3.3 is directed at companies that fail to follow best practices and ship insecure products or services. These choices result in safety issues and costs that are ultimately incurred by citizens. The objective is aimed at raising the standards of care for secure software development, mentions NIST SSDF, and calls out the importance of CVD. With respect to issues in open source software, onus should also be placed on companies that ship insecure commercial offerings with vulnerable componentry. These commercial entities are most capable of taking action to prevent bad outcomes rather than open source developers or end-users.

US SEC

Regulation S-K Item 106 requires that registrants describe their processes for assessing, identifying, and managing material risks from cybersecurity threats. Item 106 also requires that registrants describe the board of directors' oversight of those risks along with management's role and expertise in assessing and managing those risks. Similarly, foreign companies must disclose cyber risk management information in Form 20-F if they fit the definition of a foreign private issuer. Explicit technical details need not be disclosed since it could potentially and inadvertently aid attackers in targeting the registrant.

Expand your suite of security testing

If you rely solely on vulnerability assessment/vulnerability management (VA/VM) tools, sometimes called network scanning or production security scanning, you will need more types of scanning to meet most regulatory definitions of cybersecurity hygiene. Such testing approaches are simply too late in the life cycle. This practice was always more about network security and satisfying compliance, sometimes at the expense of application security, API security, and mobile security. We can now add AI security to that list, which a VA/VM tool cannot assess.

The wave of cybersecurity regulations and national cybersecurity strategies stresses the importance of "secure by design" and "shift left" approaches. It is important to assess architectures early to identify potentially exploitable design problems and business logic flaws. Security testing should also occur early and often. In reality, this requires a multitude of special-purpose testing tools beyond what VA/VM alone covers, including:

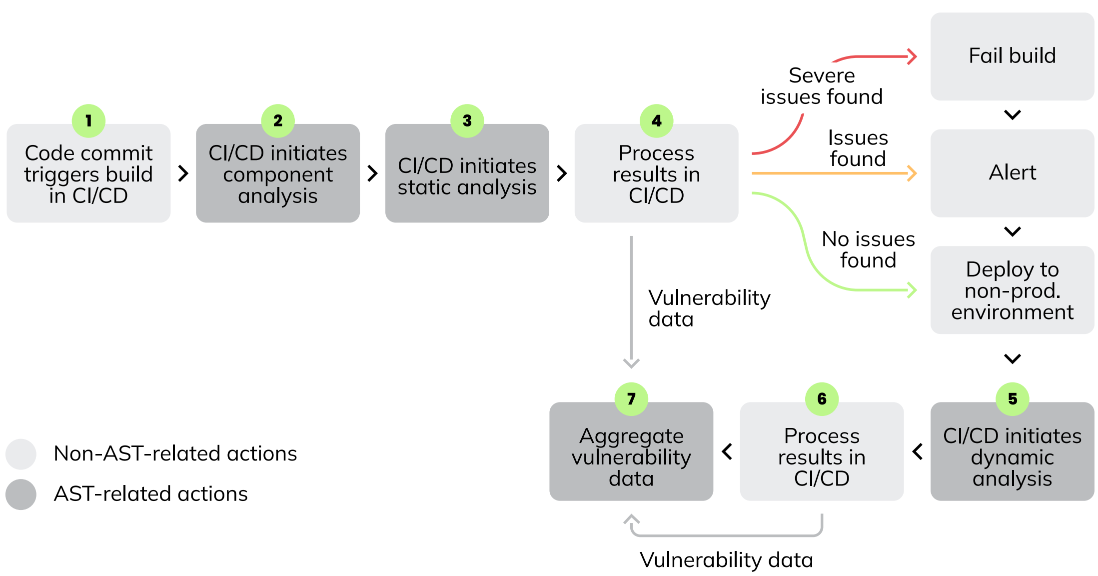
- **Software composition analysis (SCA) –** analyse dependencies such as packages and container images for known vulnerabilities, usually in the form of common vulnerability and exposure identifiers (CVE-IDs).
- **Static application security testing (SAST) –** analyse plain text source code for potentially exploitable flaws that could manifest when the code is built, compiled, and/or running.
- **Dynamic application security testing (DAST) –** analyse the running web application or web APIs for exploitable conditions like injection attacks, overflows/underruns, race conditions, information disclosures, and more.

- **Binary or protocol fuzzing –** analyse protocols and running binaries for exploitable conditions like overflows/underruns, application-level denial-of-service conditions, race conditions, and more. This is optional for most enterprise application and cloud development, though it should be employed in IoT/OT scenarios.

- **API security –** analyse web API specifications and web API traffic for interface problems, access control misconfigurations, business logic flaws, and susceptibility to automated attacks.

- **Cloud-native application protection platforms (CNAPP) –** simplify vulnerability management, risk management and scoring, and TDR in the cloud by unifying disparate cloud security solutions in one platform, which includes:

  - **Cloud security posture management (CSPM) –** analyse IaC for known misconfigurations and network exposures and ensure that a given cloud environment is appropriately hardened.
  - **Cloud workload protection (CWP) –** provide vulnerability scanning of container images and virtual machine images as well as runtime detection and protection for workloads of either type.
  - **Cloud infrastructure entitlement management (CIEM) –** analyse the human and machine identities configured within the cloud control plane and check for mispermissions or overpermissions such as access control issues.



The architecture of modern applications requires attention to the security of APIs that expose functionality and containers that power it all. API-centric designs enable functionality reuse and consumption by multiple types of clients, including web applications, mobile apps, and IoT devices. Context, usage, and purpose vary across APIs, creating different security demands. However, API analysis and protection are rarely part of the build stage, if at all. Many teams handle APIs individually, creating a heterogeneous ensemble with misconfigurations galore. It's also not uncommon for containers to be overlooked, since they are considered secure by ephemerality or obscurity, neither of which hold water. Thus, CI/CD pipelines should include API security and container security testing when using either technology.

What can go wrong?

Modern application designs result in mixed artefact types, which require different (sometimes too many) scanners. Organisations often have to concede code coverage because scanners aren't entirely adequate, or it's not possible to execute and process scans quickly enough within release windows. Static analysers require that code be in text form or easily decompilable. Analysis of compiled code (or binaries) requires additional dynamic analysers and/or fuzz testing tools. Rules must be pre-built for known vulnerabilities like published CVE-IDs, or rules built to identify broader classes of weaknesses (for example, a lack of input filtering) that can lead to potentially exploitable conditions without direct guidance on what could happen, such as susceptibility to specific attacks like cross-site scripting (XSS) or structured query language (SQL) injection. Many IT and security teams dismiss these findings as false positives, since they aren't fully actionable. Scanning also introduces slowness in releases depending on how much or how frequently the organisation consumes third-party code and executes builds.

New platforms, languages, and technology throw a wrench in the works of security scanning. A perfect example is the explosion of AI/ML, including generative AI and large language models (LLMs). With respect to security testing, someone needs to build specific scanners that "speak" the language of the given technology stack, and effective scanning often requires a blend of static and dynamic analysis methods. Today, brace for a wave of static analysers to assess the code used to build the models and analyse the training data that might improperly influence the model. It's important to verify any dependencies that go into the AI system, like any code, since known vulnerable code can also affect the security of the AI itself. Additionally, you will need special-purpose dynamic analysers and fuzzers to check for AI flaws that may emerge in runtime, such as prompt injection or sensitive information disclosure. Start with resources like the **OWASP Top 10 for LLMs** or the **U.S. NIST "Securing LLM"** recommendations until ENISA or Member States publish specific technical guidance.

With respect to security testing, someone needs to build specific scanners that "speak" the language of the given technology stack, and effective scanning often requires a blend of static and dynamic analysis methods.

CVD has yet to be widely adopted. Processes for disclosing vary greatly, sometimes because of inherent latency and human factors when multiple parties are involved. Some vulnerabilities are still reported in a way that triggers security incidents. For example, a researcher tries to wait for a supplier to acknowledge a vulnerability and goes public with their information, short-circuiting some of the expected CVD processes. Incident disclosure timelines are also not aligned with CVD processes. It's common for just one vulnerability disclosure process chain and resulting documentation to take months to complete across all parties. This stands in opposition to the one- to four-day disclosure windows that global cybersecurity regulations mandate.

Leadership discussion points to consider

01 **What is inhibiting the organisation from deploying appropriate security testing for all its technology stacks?**

Applications and infrastructure rely on many technologies. Each element of a system may require special-purpose scanners or a testing suite that's designed to handle diverse technology. Traditional vulnerability management tools will not catch all the types of issues you encounter in your operating environments, and such an approach is insufficient for meeting cybersecurity regulatory requirements.

02 **How is the organisation sharing vulnerability information and encouraging transparency?**

Coordinated vulnerability disclosure requires extensive information sharing with organisations, government entities, and security researchers. Cybersecurity regulatory authorities are promoting CVD as an effective way to improve security hygiene so the bar is being set purposefully high. The organisation's CVD processes, including what data can be disclosed, should be documented as part of its risk management program.

Stay the Course for Cybersecurity Hygiene and Resiliency

In the context of cloud security, providers and consumers strive to adhere to diverse regulatory requirements while innovating to deliver secure and compliant services. As regulations become increasingly stringent, there's a growing need for a proactive and continuous approach to meet the highest standards of security and resiliency and disclose cyber incidents, when they happen, promptly. In the case of the EU NIS 2 Directive, Member States will provide additional guidance that may be more prescriptive. You must adjust your cybersecurity strategy as the appropriate regulatory authorities publish their revised guidance.

To provide high-quality, scalable, and reliable products and services, your cybersecurity strategy should satisfy and exceed the basics outlined here.

We expect more transparency and collaboration between public and private entities. Additionally, incident disclosures will trend upward by virtue of modern system design and an expansion of what regulators want to see to protect critical infrastructures, ensure the safety of citizens, or mitigate economic impacts.

Compliance is often viewed as the low bar for actual security. Cybersecurity regulations have raised that bar and will continue to do so, effectively imposing rules and a division of labour where there is a gap. To provide high-quality, scalable, and reliable products and services, your cybersecurity strategy should satisfy and exceed the basics outlined here. Cyberthreats will advance rapidly because of the explosion of AI, and cybersecurity must keep pace.

About Sysdig

In the cloud, every second counts. Attacks move at warp speed, and security teams must protect the business without slowing it down. Sysdig stops cloud attacks in real time, instantly detecting changes in risk with runtime insights and open source Falco. We correlate signals across cloud workloads, identities, and services to uncover hidden attack paths and prioritise real risk. From prevention to defence, Sysdig helps enterprises focus on what matters: innovation.

To learn more about Sysdig, visit sysdig.com

REQUEST DEMO →

sysdig

WHITE PAPER

WP-009 REV. A 04/24