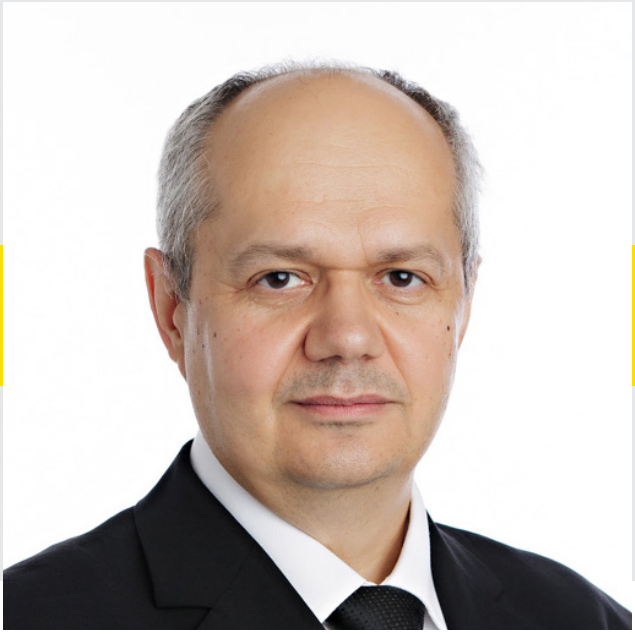
Synechron

Designing a Next Generation Pipeline for Cloud Native Application Compliance

Cloud has enabled the continuous delivery of applications at a speed that meets business demands. In a fast-paced workflow, how do financial services firms ensure that cloud native apps remain secure and compliant to various industry regulations? SMEs from Synechron answer some of the most important questions about continuous compliance.

Nenad Bulatovic

Associate Director
The Netherlands



What tooling would you demand for a next-generation pipeline?

In modern times there is increasing demand for more and more software products, lifecycles are quicker and time to market is significantly shorter. New technologies are emerging more quickly than ever, and old ones are updated more frequently. It has become almost impossible to keep track of all technologies necessary to build modern web or enterprise applications, not to mention ever-emerging security and performance concerns.

In such circumstances, Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) comes as a rescue.

Freeing developers of constantly thinking about security is what SAST can help with. It is also known as 'white box testing' because it depends on the internal structure of the application. It is not necessary for every engineer to know in great detail all about secure programming patterns. SAST tools can scan source code and its dependent libraries for any vulnerability in the CI/CD pipeline or even during development as all major IDEs have such plugins. This is not only for security, but for best programming practices as well.

There is wide choice of popular tools for IDEs, servers, and CI/CD pipelines. Just to name a few, these include: Klocwork, Veracode, HCL AppScan, Sentinel, Checkmarx, SonarQube, Micro Focus Fortify, etc. One of the key benefits are that these tools detect problems early in the development stage even when there is no running application, thereby minimizing the risks and costs of rewriting completed applications because of security issues. These tools usually analyze: configurations, semantics, dataflow, control flow and structure.

The opposite approach, but with the same outcome, is taken with DAST tools as they rely on a 'black box' approach. By their nature, these allow for a dynamic process, meaning that there must be a working version of the application. Therefore, scanning and detection methods are specifically configured to the type of the application under the testing, and for checking most suitable inputs and outputs.

DAST can detect a wide range of vulnerabilities, such as SQL injection, authentication and encryption, sensitive data exposure, insecure deserialization, insufficient logging and monitoring, and cross-site scripting attacks, just to name a few. Also, those tools can check for web services, infrastructure such as storage, networks, and more.

It is important to understand that SAST and DAST tools are neither competing nor exclusive tools, but they should be used together in a mixed/complementary mode. They've become necessities in our fast moving and software-rich world which is constantly the target of cyber attacks and exploits. Small investments at the beginning, during development and deployment, can save company reputations and significant money when used properly to defend company assets. Hence, both SAST and DAST tools should be carefully chosen and used across organizations from the very beginning of software development.

Sonal
Vaid

Principal Consultant
The Netherlands

How has cloud adoption and recent heightened awareness of privacy and data protection affected how application teams approach compliance?

Compliance is defined as the action of complying with a general set of rules and regulations. The technical application for compliance means designing and developing the software by always keeping the regulation, industry guidelines or internal controls at the center. By following these standards and guidelines, we can mitigate the risks such as a data breach, a breach notification law, an unknown external breach, or a known internal breach.

With the General Data Protection Regulation (GDPR), the European Commission intends to strengthen and unify data protection for individuals within the European Union (EU). With the non-compliance of GDPR, the penalties can be up to 20,000,000 EUR or, in the case of an undertaking, up to 4% of the total worldwide annual turnover of the preceding financial year, whichever is higher. In case of a breach of personal data, the organization breached will be required to notify the subjects of the breach, without undue delay, within a timeframe of 72 hours.

For GDPR there is no auditor as such. GDPR does not have a central authority which will give organizations a certificate of compliance. Hence, the responsibility to keep Personally Identifiable Information (PII) of EU citizens secure lies with each organization.

The application can be designed and developed keeping in view a few broad categories. The first classification is identifying the technical and non-technical controls to be placed.

1. General Awareness

The most important aspect for compliance is awareness. It is necessary to make the DevOps teams aware, and educate them by giving them knowledge of the regulation and the impact of non-compliance to the organization. The architects and DevOps experts can leverage the tools and products available in the industry as part of the enterprise architecture.

2. Protection of Information

As per GDPR, data protection for individuals within the European Union (EU) is mandatory. Hence, the applications designed with PII in mind must be developed to secure the data and ensure it is not visible to all. This can be achieved via different ways, such as static or dynamic data masking, or as to Pseudonymize or Anonymize data, character scrambling, encryption, shuffling and more. Protection of data is the key where any data transfer or data viewing takes place.

3. Separation of Duties

The work executed can be based on teams, roles, levels and departments. The authorization matrix of users can be designed with different views based on data in use, data at rest and data in motion. Only authorized team members can have access to real production data or PII. An application can function as designed but the PII must only be seen in plain text by those who have the right to know; for example: a judge, the management team of an organization, etc.

The application compliance or data usage compliance does not end with GDPR. With the increase in use of Public Clouds, Schrems II has been introduced. The European Court of Justice issued the Schrems II judgement with significant implications for the use of US Cloud services. Customers of US Cloud service providers must now themselves verify the data protection laws of the recipient country, document its risk assessment, and confer with its customers.

Paul Jones

Associate Partner
UK

What metrics should be exposed from a pipeline? What might different personas want to see? How would they like to see them?

Continuous Integration was born out of Extreme Programming in the late 1990s and started to become commonplace in the early 2000s. But it wasn't until Jez Humble and David Farley's influential *Continuous Delivery* (2010) that the concept of a pipeline really grabbed the attention of the wider industry.

With a desire to deploy as well as integrate came a need for a whole new set of tools, metrics and gates. With increased adoption of cloud in the 2010s, the industry moved to Infrastructure as Code, which further widened the scope. With seemingly everything in a pipeline, we now have a lot of data at our disposal.

Application teams want to be confident that the quality of their product is not dependent on slow, error-prone 'by eye' checks and tests:

- Their tests should run automatically, and their test coverage bar must be maintained
- Their minimum code quality bar – as assessed by formatting, linting & static analysis tools – must be maintained
- They can have confidence that security is being built into their product from the offset, including avoiding hard-to-spot coding mistakes and keeping accidental secrets out of the codebase

Security teams want to be confident that their product is free of vulnerabilities, either introduced in freshly written code or in the dependencies the project leverages. Traditional models tended to have a security review just before go-live, at the end of the 'build' phase. In high performing teams, this is no longer acceptable.

- Applications, containers and operating system images must be statically and dynamically tested for vulnerabilities
- Runtime environments are monitored for entities that need patching
- Scan results must be visible in the Merge Request to catch errors before they enter the codebase

Audit & Compliance teams want confidence that:

- They can vouch for (and give evidence for) the quality and security of code going into production
- Unapproved open source licences (including those of 3rd degree dependencies) are kept out of the codebase
- Complex regulatory demands are being met by new infrastructure and software deployments

The outputs of these tests should be shifted as far left as possible, ideally into the tests run on a given feature branch, failing that on a Merge Request, and failing that in the trunk, before release.

In which areas are our clients typically furthest back on the maturity curve? What do they do well already?

Many of Synechron's clients have begun large programmes of work to make mass migrations to cloud. We are increasingly seeing that these programmes are not only about cloud, but instead also being used as drivers of a cultural shift, tool modernization, upskilling and quality improvement.

The position of an application on the maturity curve varies greatly from team to team. Legacy applications, especially those that are largely left to 'just' be operated, unchanged, have much further to go.

Whilst there is generally reasonable adoption of Git for version control (something that could not be said five years ago when SVN was still prevalent), we do not always see adoption of Git Flow or equivalent feature-branch based development techniques.

Whilst there is generally reasonable adoption of a pipeline, we do not always see continuous integration, with pipeline execution on every commit. Some teams feel limited by legacy tooling and see Cloud adoption programmes as a driver for change here.

Although infrastructure-as-code, and in particular Terraform, has become prevalent, we do not always see adoption of testing and policy-based tooling to ensure the quality of the infrastructure that is being deployed.

Tooling and technical challenges can be more easily fixed. The harder problems to solve are cultural and organizational. Cloud skills have never been in greater demand. There is a risk that moves to, for example, Google-style Site Reliability Engineering teams may become little more than a rebranding of existing on-prem Ops as Cloud Ops.

How can this data be used to drive light touch or hands-free change control? What barriers would remain?

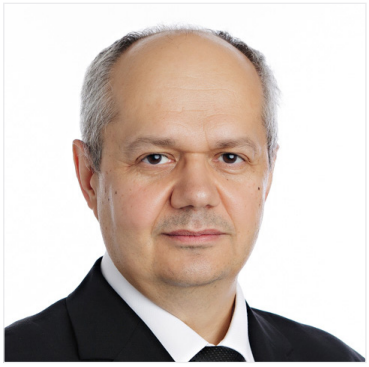
In some organizations, change control teams far removed from the product that is being modified are asked to make a risk assessment of a release, often without being close enough to understand the intricacies of a release team's work. In some organizations, teams are penalized for frequent, small releases (even when successful) either directly -- by a taint on the release team's permit to operate -- or indirectly via the burden of release paperwork and change approval wait times.

The data a pipeline can provide (including code quality and test coverage checks, proof of deployment to non-production, and use of pre-approved patterns) should be used to drive an organization towards a more automated change control and risk assessment process.

At its most effective, change control is supported by individuals close enough to the team to understand the detail of a change, but sufficiently far removed to offer an outsider's perspective. A high performing team will have individuals who can ask the hard questions that automated checks might not have caught, thereby improving the reliability of a commonly owned product or service, not requirements to complete paperwork that duplicates information already captured elsewhere.

Many organizations are still coming to grips with the cultural shift that full DevOps requires. Breaking down these team-to-team interfaces is crucial to building a high performing culture.

About our SMEs:



Nenad Bulatovic
Associate Director
The Netherlands

Nenad is a proficient solution architect with extensive experience in the financial industry, including working in several niche areas of banking, such as FX and international/global markets. He previously worked for Synechron in Paris, Serbia, US Central, NYC, India, and is now servicing Amsterdam and Canada clients. His thorough technical knowledge of development, cloud, database and interfaces combines with his many years of experience to oversee the complete technical landscape within an organization. He has experience with vendor selection, RFP/RFI processes, and has supported several supplier assessments making sure business objectives and the technical domains are well aligned.

To know more about Nenad

https://www.linkedin.com/in/nenadbulatovic/

Sonal Vaid
Principal Consultant
The Netherlands

Sonal has over 19 years of comprehensive Banking Industry, IT domain and process consulting experience with marquee clients across several geographies. She is a transformation expert in bringing ideas to life and enabling teams to focus on high-performance and delivery excellence. She has worked with clients to set up niche technologies using DevOps as a way of working. Sonal currently leads the Cloud Technology Practice at Synechron, Amsterdam in the Netherlands. She is also a certified Personal Development Coach.

See more about Sonal

https://synechron.com/profile/sonal-vaid

Paul Jones
Associate Partner
UK

Paul is an AWS, Google and Cloudera-certified IT consultant focused on topics such as software architecture, DevOps & CI/CD, Cloud and Big Data. He has strong software project delivery expertise including Agile methodologies and product management. Recently, he led a multi-disciplinary team that delivered a bank-wide Data Lake platform serving regulatory and analytical requirements. Paul has five years' experience running an international eTrading platform delivery, integration and support team resulting in a deep understanding of the FX trade lifecycle and accompanying technologies.

To know more about Paul

https://www.linkedin.com/in/paulmatthewjones/

www.synechron.com