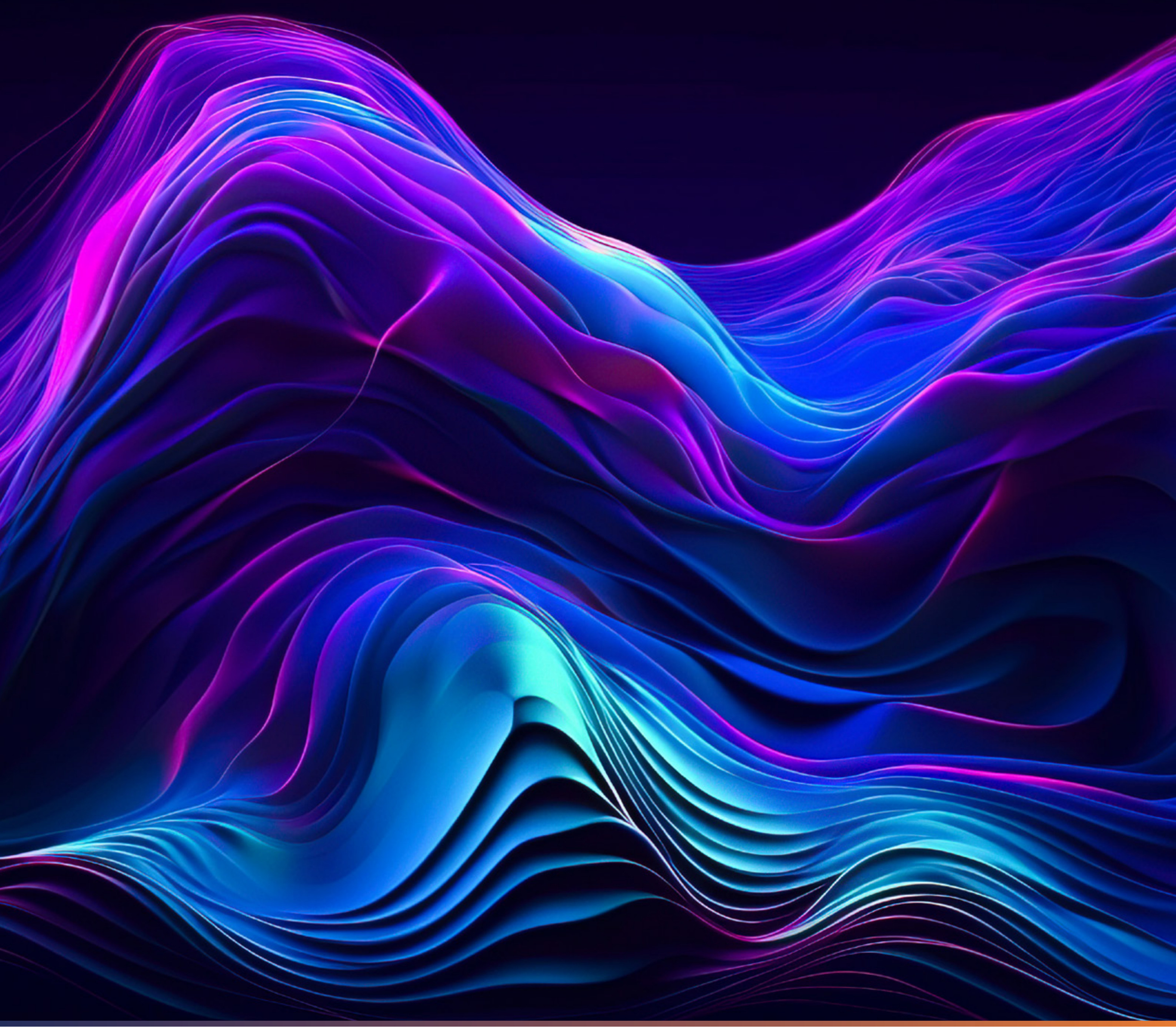
Shift Left Group

Making Business and Technology Better



Revolutionising DevSecOps:

Embracing Generative AI for Enhanced Security and Efficiency

The rise of cloud computing has driven the adoption of DevOps practices for faster software delivery. However, this speed often comes at the expense of security. DevSecOps integrates security throughout the development lifecycle to address this challenge. But implementing DevSecOps can be complex for developers already overloaded with new technologies. Generative AI offers a promising solution by automating security testing, generating documentation, writing code and recommending patches.
This whitepaper explores the challenges and benefits of using Generative AI to revolutionise DevSecOps practices.

Author: Maqsud Mohammad

INTRODUCTION

Over the last decade the software development landscape has undergone a paradigm shift due to an increase in cloud adoption. A decade ago, cloud adoption rates were relatively low with only a small percentage of companies leveraging cloud services for their infrastructure needs. Estimates suggest less than 10% of businesses were fully utilising cloud computing services at that time.

Cloud adoption rates have increased since 2020 due to the Covid19 pandemic. Research from Boston Consulting Group (BCG) highlights the staggering rise of public cloud adoption. In the past five years, spending on cloud services from the top three providers (Amazon, Microsoft, and Google) has skyrocketed by 337%, reaching $211 billion in 2023. This trend points to a relentless increase in cloud adoption, which is set to accelerate further with the advent of Generative AI.

With the rise in cloud adoption, development and deployment practices have also evolved. Many organisations have shifted from the expensive datacenter-based hosting model in favour of the cloud. Key drivers for this shift include significant cost savings and the substantial business value that cloud solutions provide. The latter includes:

- agility in operations,
- faster time to market,
- low cost of failures/quick time to recovery,
- scalability and flexibility in resource management,
- fostering innovation through enhanced capabilities.

The push for cloud adoption and increased business agility has necessitated innovative approaches, bringing microservices architecture and DevOps practices to the forefront. These methodologies have become essential for achieving success in the cloud environment.

While DevOps facilitates rapid development and deployment, it also shines a light on the limitations of traditional security practices which only happen at the end of the IT change lifecycle. The ever-expanding threat landscape, fuelled by the growing use of Open-Source Software (OSS) and the rapid exploitation of vulnerabilities, demands a proactive approach.

According to SecurityWeek a staggering 56% of vulnerabilities are exploited within just seven days of public disclosure, which is a disturbing trend. Resolution of these vulnerabilities can also take a significant amount of time.

The SANS 2023 DevSecOps survey reported that almost 50% of the organisations surveyed said it takes them anywhere from three to four weeks to patch critical security vulnerabilities. This prolonged exploitation window leaves security teams scrambling to implement patches before attackers strike.

This urgency often creates friction between DevOps and InfoSec teams, as releases are blocked due to insecure coding practices and outdated open-source software. These blockers are common sources of malware and other vulnerabilities. DevSecOps bridges this gap by integrating security throughout the Software Development Lifecycle (SDLC), thereby ensuring a more secure development process.

As organisations navigate the complex terrain of DevSecOps adoption, they encounter a myriad of challenges. These range from cultural resistance to technical complexities. However, amidst these challenges lies the transformative potential of Generative AI. By harnessing its power organisations can automate tedious tasks, bolster their security posture, and empower developers to focus on innovation and core functionalities.

Generative AI holds the promise of revolutionising DevSecOps practices, but its integration into the SDLC is not without its hurdles. From data privacy concerns to algorithmic biases and LLM hallucinations, organisations must navigate a range of challenges to fully realise the benefits of Generative AI.

This whitepaper explores the challenges faced in cloud adoption, the evolution of DevOps, its critical integration with security to form DevSecOps and the challenges in implementing new ways of working. It will also discuss the promising frontier of Generative AI and its role in overcoming implementation obstacles.

PARADIGM SHIFT IN SOFTWARE ARCHITECTURE

With the rise of cloud computing, there has been a significant shift away from monolithic architectures towards Microservices based architectures or cloud-native architectures. While some legacy systems still adhere to monolithic designs, many organisations have embraced Microservices and event-driven architectures to capitalise on the benefits of cloud-native development.

Microservices offer benefits such as scalability, resilience, and agility, aligning well with the distributed nature of cloud environments. Companies that have adopted microservices architecture to decompose complex monolithic applications into smaller, loosely coupled services have reaped substantial benefits from their cloud investments. These smaller services can be independently developed, deployed, and scaled, providing greater flexibility and efficiency.

We've observed this firsthand with our customers. It's also a sentiment echoed by McKinsey. Their surveys confirmed that only a small fraction of companies were able to capture the full value of the cloud. This indicates that most cloud transformation programmes are falling short of their potential.

However, research by BCG demonstrates that when cloud transformation programmes are implemented correctly, they can reduce infrastructure costs by up to 40%, boost productivity by as much as 50%, and accelerate time to market by up to 60%.

This can mean significant savings for senior managers, who can invest in other strategic initiatives that their companies can profit from.

Moreover, developers can concentrate on creating innovative solutions without the burden of managing server infrastructure, while marketing teams benefit from a quicker product turnaround, enabling them to capitalise on emerging trends. However, there remains a significant gap between the expected outcomes and the actual results of cloud transformation programmes.

WHERE ARE THEY GOING WRONG?

The companies that are not deriving value from their cloud investments are not paying attention to the data at their disposal. Airline pilots look at the real-time weather data to plan their journey ahead to avoid any turbulence to give the passengers a smooth and safe ride and to make the journey cost effective for the company. The same applies to your cloud transformation journey too. Just as using real-time weather data leads to a smoother flight, leveraging real-time business and technical data throughout a cloud transformation journey can lead to more efficient operations and improved decision-making for companies. This approach allows companies to make strategic investments to:

Re-evaluate their business processes:

Identify areas where cloud services can improve efficiency, agility, and scalability instead of moving all services to cloud, thereby treating it as another data centre and not a modern platform.

Modernise applications:

Many organisations, having previously invested in significant rearchitecting, might hesitate to reinvest in modernising these systems and adopt a “lift and shift” approach to transferring applications to the cloud. This often leads to inefficient resource utilisation and increased costs. For organisations to fully leverage cloud capabilities, it is crucial to either rearchitect applications into cloud-native formats or refactor existing applications to better utilise cloud features. Strategic modernisation, therefore, not only enhances performance but also optimises cloud resource use and expenditure.

Upskill their workforce:

Ensure their team understands how to utilise and manage cloud-based systems or to choose the right Cloud Service Provider who can help them navigate the complexities of cloud.

Monitor and manage cloud costs:

Efficient cloud cost management is key to maximising the return on cloud investments. Cloud sprawl, a phenomenon where unchecked proliferation of cloud resources leads to unnecessary costs, is a significant challenge as validated by research from McKinsey and BCG. It’s crucial for organisations to implement robust mechanisms to track and control cloud spending effectively. Adopting strategies such as the use of reserved instances and the implementation of stand-up/tear-down processes can play a critical role in cost optimisation. Reserved instances provide significant cost savings over standard pricing as they allow for committing to cloud resources for a predefined period at a reduced rate, which is ideal for predictable workloads. On the other hand, implementing stand-up/tear-down automation for development and testing environments can drastically reduce costs by ensuring that you pay only for the resources when they are actively being used. These tactics not only prevent financial leakage but also promote a disciplined approach to resource allocation, ensuring every dollar spent on the cloud is fully optimised.

EVOLUTION OF DevOps

Though DevOps as a term was coined before cloud computing it gained momentum and is now intimately associated with cloud adoption. The philosophy behind DevOps is to enable rapid deployments into production at a much higher cadence than waterfall approaches (e.g. a few times a week vs once every 6 months). Though it sounds normal today, it was only 10 years ago that deployments being made once a month was uncommon. In addition, to get to a release required navigating complex merges with lots of branches and extensive manual testing. More teams were following Agile software development practices which encourages iterative development through customer collaboration, rapid change, and continuous delivery. However; the deployment model was still waterfall. Software development and operations were still two different teams and deployments to production had to be negotiated. This often created bottlenecks causing delays and frustration. Businesses were losing value due to this siloed approach.

To overcome this challenge DevOps was introduced. This practice breaks siloes and encourages collaboration between development and operations teams, making the software delivery a seamless operation, with faster feedback loops.

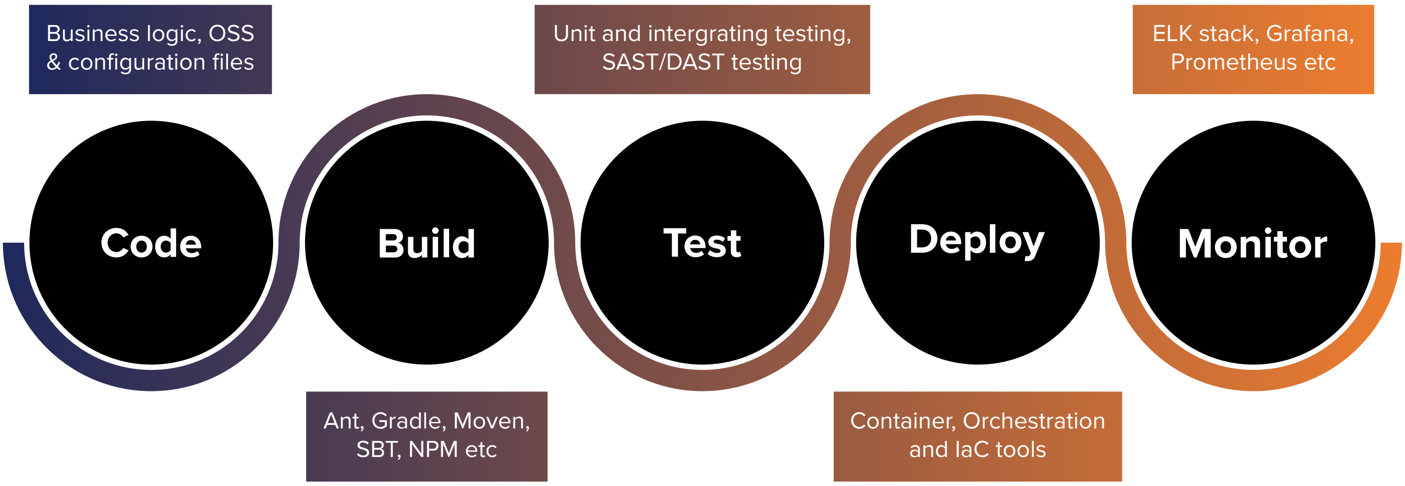
DevOps practices have evolved over the years and the advent of cloud computing has removed most of the manual processes. As well as software delivery, infrastructure creation is now done via CI/CD pipelines, making software delivery a seamless process. With cloud adoption came new development practices and with it came new challenges. The very nature of cloud being a multi-tenant deployment model where you share infrastructure with others means security becomes paramount. DevOps has significantly evolved with the advent of cloud computing. While the core principles of collaboration and breaking down silos between development and operations remain, cloud computing has introduced new opportunities and challenges that have reshaped DevOps practices. It has amplified the "continuous everything" concept of DevOps, increasing automation across all phases of the Software Development Lifecycle (SDLC). This evolution has led to more efficient, scalable, and agile development processes.

Some of the key changes are:

- Handling complexities of distributed systems introduced by microservices architecture.
- Infrastructure as Code (IaC): While the "pets vs. cattle" mindset towards infrastructure predates cloud computing, it gained prominence with cloud adoption. This shift emphasised managing infrastructure through code rather than hardware, allowing for more scalable and efficient operations.
- Containerisation and orchestration: The majority of applications that are deployed on any cloud platform are containerised, which is a way of packaging applications code and its dependencies in a standardised unit called a container. Orchestration adds another layer of complexity by adding more configuration files and scripts through which we control the container deployments.
- Monitoring and observability: Though this also predates cloud, cloud platforms offer built-in monitoring tools that track application performance and resource utilisation. These tools can trigger automatic scaling actions, adjusting resources up or down based on real-time demand, optimising costs and ensuring high availability.

With an increased emphasis on code in all phases of the SDLC, the threat surface has also increased. The key challenge that came out of these changes is Software Assurance which is defined by Office of Safety and Mission Assurance (OSNA), a branch of NASA, as "the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software functions in an intended manner."

Fig1: Modern Software Supply Chain


Business logic, OSS & configuration files
Unit and intergrating testing, SAST/DAST testing
ELK stack, Grafana, Prometheus etc
Code
Build
Test
Deploy
Monitor
Ant, Gradle, Moven, SBT, NPM etc
Container, Orchestration and IaC tools

Increasing the focus on code throughout the SDLC amplifies risks by multiplying potential entry points for vulnerabilities, originating either from our code or via external parties in the supply chain.

UNDERSTAND THE WEAKEST LINK IN YOUR SUPPLY CHAIN

According to the 2024 OSSRA Report by Synopsis, 96% of surveyed codebases contained open-source software (OSS), with 84% of these having at least one open-source vulnerability. The key takeaway is that OSS dominates codebases, and without proper safeguards to monitor for vulnerabilities, significant damage to an organisation's reputation and finances can occur. As the saying goes, a chain is only as strong as its weakest link, and it's likely that your weakest link is OSS.

Some recent examples of OSS vulnerabilities are:

SolarWinds Orion Authentication Bypass Vulnerability (2020)

Target: SolarWinds Orion Platform a SaaS based widely used network monitoring software.

Vulnerability: Hackers infiltrated the SolarWinds Orion Platform supply chain by injecting malicious code into a legitimate update. This code gave attackers remote access to systems running the Orion software through an update called Sunburst which had to be manually installed on client machines. This vulnerability also allowed attackers to bypass authentication to execute API commands.

Impact: Over 18,000 organisations, including government agencies and private companies, were potentially compromised. The attackers could steal data, disrupt operations, and launch further attacks.

Codecov (2021)

Target: Codecov, a software testing platform used by developers to measure code coverage.

Vulnerability: Hackers compromised a Docker image used by Codecov. This allowed them to inject malicious code into the build process, potentially affecting any software that used Codecov for testing during that period.

Impact: While the full extent of the breach is unknown, thousands of companies potentially had their code exposed to attackers. This allowed attackers to steal sensitive information like secrets and passwords and inject vulnerabilities into the software.

Log4j Vulnerability (2021)

Target: Apache Log4j, a popular logging library used in Java applications.

Vulnerability: A flaw in Log4j allowed attackers to inject malicious code into log messages, which could then be executed by the application which could let attackers break into systems, steal passwords and logins, extract data, and infect networks with malicious software.

Impact: Widespread vulnerability impacting millions of applications. Attackers could exploit this to execute arbitrary code on affected systems, potentially leading to data breaches, ransomware attacks, and other compromises.

OpenAI Redis Vulnerability (2022)

Target: OpenAI, a research company focused on artificial intelligence (AI).

Vulnerability: A bug in the redis-py a caching library (used by OpenAI to connect to Redis). The library manages connections to Redis and reuses them for efficiency. When using asynchronous operations (asyncio), requests and responses are queued. The vulnerability arose when a request was cancelled after being sent but before receiving a response. This left the connection corrupted. Subsequent unrelated requests might receive “leftover” data from the cancelled request, leading to the leak.

Impact: While the extent of the breach is unclear, some data from OpenAI users might have been exposed.

What we can see from these attacks is that the attackers know that the companies’ defences are strong, so they use deception to enter via a backdoor. Even Fortune 500 companies like Microsoft, IBM and the US military have been compromised by vulnerabilities in their supply chain. Companies need to adopt secure coding practices and integrate Software Composition Analysis (SCA) into their CI/CD pipeline to detect and respond to supply chain attacks. It’s crucial to stay updated on the latest vulnerabilities and take steps to patch them promptly.

DevSecOps: A SHIFT LEFT APPROACH TO SECURITY

The industry has acknowledged the scale of the threat and there are guidelines and frameworks prescribed by National Institute of Standards and Technology (NIST) and Cloud Security Alliance to promote secure development practices.

However, checks and balances still need to be built into the CI/CD pipelines. The philosophy of DevOps to build and release faster is no longer sustainable as we also must consider Software Assurance. Leaving security until the very end of the lifecycle is causing friction with InfoSec and the DevOps teams as it can delay releases, potentially creating technical/security debt. When business priorities take precedence over upgrading outdated OSS and fixing known vulnerabilities, it can exacerbate these issues, compromising both security and efficiency.

To overcome these challenges, a new philosophy called DevSecOps emerged which integrates security throughout the SDLC with the aim of providing continuous security in software delivery.

Although novel at its inception, DevSecOps has been rapidly adopted by the industry and has become a critical focus. Nearly every organisation on the cloud or in the midst of a cloud transformation journey is implementing DevSecOps in some of their key services. The SANS 2023 DevSecOps survey confirms that it's seen as a business-critical practice, not just a security concern.

CHALLENGES WITH IMPLEMENTING DevSecOps

The core concept of DevSecOps is to address security concerns in the software supply chain by incorporating new tools and practices from the early phases of the SDLC.

Developers were already experiencing cognitive overload from keeping up with a myriad of new cloud-native tools and technologies. Adding security to the mix only compounded this complexity, not just for developers but also for non-technical stakeholders. Security, traditionally an InfoSec concern, suddenly became a developer's responsibility. Without proper training in secure coding practices and security tools, this integration was bound to encounter significant challenges.

Kam and D'Arcy (2023) argue that integrating security into DevOps can lead to role transitions for developers, potentially causing cognitive overload due to the added responsibilities of managing systems operations and security testing. This increased burden can contribute to job burnout, which negatively impacts the continuous security practices required of developers, ultimately undermining software assurance.

USING GENERATIVE AI WITH DevSecOps

Generative AI has revolutionised the technology world with its ability to create novel and realistic content, including code, text, and data, impacting nearly every industry. A survey by Synk involving 537 software engineering and security team members and leaders, highlights the rapid adoption of Generative AI. A whopping 96% of teams now use AI coding tools, integrating them deeply into the software supply chain.

Generative AI technology can significantly enhance, DevSecOps in several key areas of Quality Engineering (QE) . These include:

Automated test case generation and security testing

Generative AI can create diverse and realistic test cases encompassing edge cases and potential attack vectors that traditional testing methods might miss. The comprehensive testing not only speeds up the testing process but also improves software resilience and reduces likelihood of vulnerabilities slipping through the cracks.

Security policy and documentation generation

AI can automatically generate clear and concise security policies and documentation tailored to specific projects and coding languages. This eliminates the need for manual drafting, saving time and ensuring consistency across projects.

Code generation and refactoring

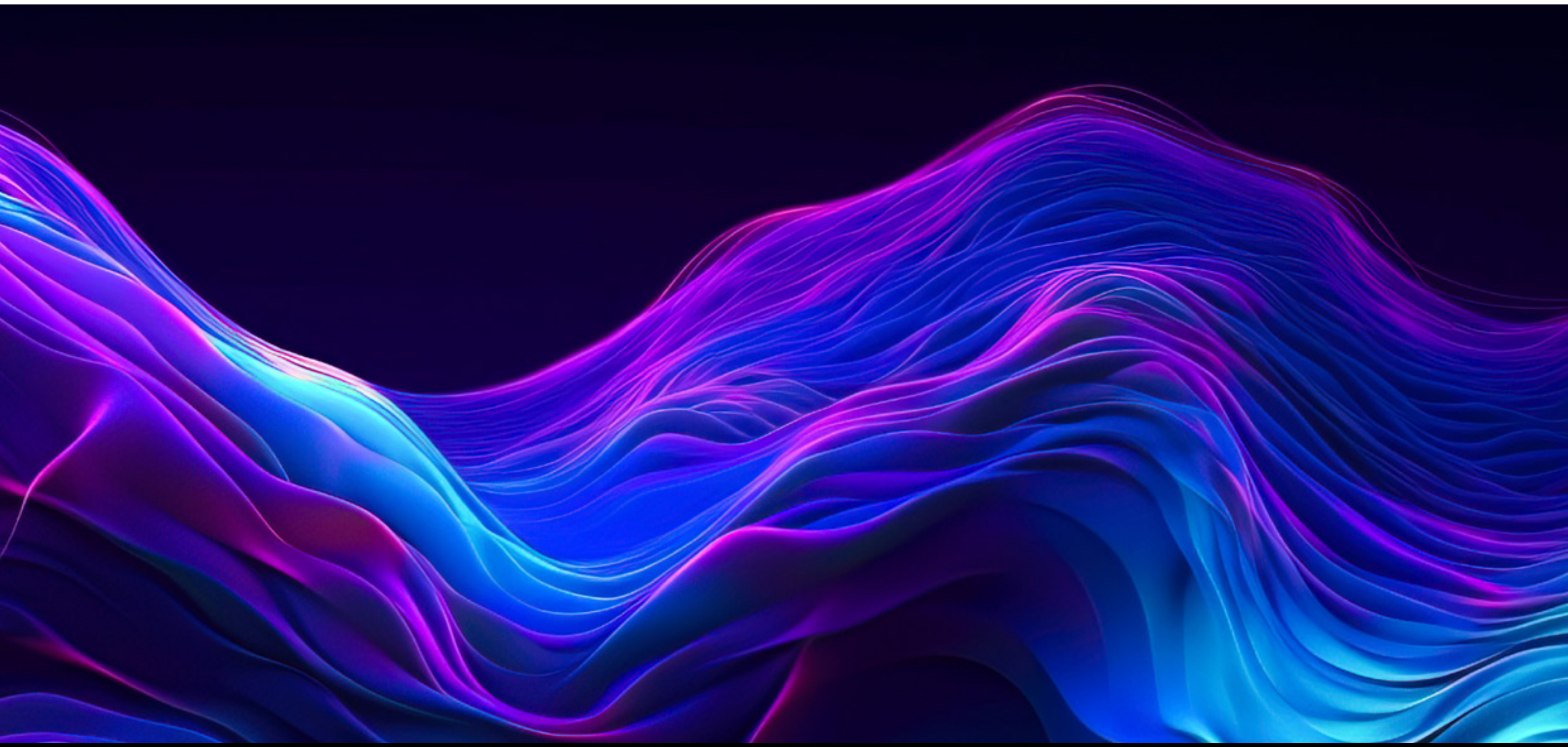
Generative AI can automate the creation of secure boilerplate code, unit tests, and comments, freeing developers to focus on complex functionalities. Additionally, AI can assist in code refactoring, including suggesting improvements for security and efficiency.

Vulnerability remediation and patching

Generative AI can analyse codebases, identify vulnerabilities, and recommend appropriate patches. This streamlines the patching process reducing the window of exposure to potential threats.

Data augmentation for security testing

Security testing often relies on pre-defined datasets, which may not capture the full spectrum of real-world threats. Generative AI can be used to augment existing datasets by creating synthetic data that mimics real-world attacks and malicious traffic patterns. This allows for more robust and realistic security testing.



BENEFITS AND CONSIDERATIONS FOR IMPLEMENTING GENERATIVE AI

Integrating Generative AI into DevSecOps workflows offers a multitude of benefits and addresses several key challenges.

These include:.

- **Increased efficiency and speed:** Automation of repetitive tasks frees up developer time and resources, leading to faster deployments and improved development velocity.
- **Enhanced security posture:** Comprehensive testing, automated vulnerability remediation, and proactive threat detection significantly reduce the risk of security breaches.
- **Improved developer productivity:** By handling mundane tasks, AI empowers developers to focus on critical functionalities and innovation.
- **Reduced costs:** Automation and proactive security measures can lead to cost savings by avoiding security incidents and rework due to vulnerabilities.

However, implementing Generative AI in DevSecOps is not without challenges and it needs some sort of human intervention at the early stages. Below is a list of some of these:

- AI models are only as good as the data they are trained on. Biased training data can lead to biased outputs especially if using open source LLMs.
- Whilst AI automates tasks, human oversight remains crucial. In their study on the use of AI in coding, Perry et al. (2023) demonstrate that AI code assistants can inadvertently produce insecure code if not carefully managed. Additionally, a Synk survey highlights that, despite their widespread adoption, Generative AI tools often generate insecure code suggestions, validating the need for human intervention before we accept AI generated code. Perry et al. (2023) found that user interaction with AI assistants significantly impacts the security outcomes of their code. Users who provided specific instructions or used security-focused prompts tended to produce more secure code. However, this implies that developers must invest additional time in crafting precise prompts and actively engage with the tools to ensure AI-generated code aligns with security best practices. This added effort can be a burden, potentially slowing down development and requiring extra diligence to mitigate risks.
- Generative AI models themselves can be vulnerable to manipulation by attackers. Security measures need to be implemented to protect the models and the data they process.
- LLM hallucinations pose a serious security risk, as they can allow bad actors to introduce malware that the LLMs may treat as legitimate and subsequently train on. This risk underscores the critical need for human intervention to review AI-generated code before it is deployed into production systems.
- With Generative AI still in its early stages, organisations face the challenge of bridging the skills gap and addressing the training and development needs of their current workforce. Ensuring that employees are equipped to effectively use these tools is crucial for producing the best outcomes.

CONCLUSION

As cloud computing continues to reshape the software development landscape, DevSecOps emerges as a crucial strategy to embed security within the SDLC, aligning with rapid deployment capabilities enabled by DevOps practices.

By automating tedious tasks, enhancing security testing, and fostering developer productivity, Generative AI has the potential to revolutionise DevSecOps practices. As AI technology matures and security considerations are addressed, organisations that embrace Generative AI can unlock faster and, more secure software development. The adoption of generative AI presents a promising avenue to enhance these integrations, offering significant improvements in security testing, policy generation, and vulnerability management.

However, the successful implementation of AI in DevSecOps requires careful consideration of the quality of training data, the need for human oversight, and the security of the AI models themselves. Organisations must navigate these challenges thoughtfully to fully harness the potential of generative AI, thereby achieving a more secure, efficient, and innovative development environment.

As DevSecOps continues to evolve, it will not only meet the current demands of software development but also anticipate future security needs, ensuring organisations can maintain resilience against an ever-changing threat landscape. Embedding security throughout the software development lifecycle, as advocated by DevSecOps, is not just a technical necessity but a strategic imperative. With the integration of AI, this approach becomes even more critical, enabling organisations to harness cutting-edge technologies while safeguarding both current and future operations.

GLOSSARY

Cloud-Native: Applications designed specifically to run in a cloud environment, utilising the cloud's flexibility, scalability, and resilience.

CI/CD (Continuous Integration/Continuous Deployment): Practices that automate the integration and deployment of code changes, enabling frequent and reliable releases.

DevOps: A set of practices that combine software development (Dev) and IT operations (Ops) to shorten the development lifecycle and deliver high-quality software continuously.

DevSecOps: An extension of DevOps that integrates security practices into the entire software development lifecycle.

Dynamic Application Security Testing (DAST): A type of security testing that analyses a running application to find vulnerabilities by simulating external attacks.

Generative AI: AI technologies that can generate new content, such as code, text, or images, based on learned patterns from training data.

Infrastructure as Code (IaC): The practice of managing and provisioning computing infrastructure through machine-readable configuration files, rather than physical hardware management or interactive configuration tools.

LLM: Large language models are a type of Artificial Intelligence model which are trained on large datasets to understand using deep learning techniques and generate new content which can be text, code, images and videos.

Microservices: A cloud-native software development technique that facilitates faster development cycles, improved scalability, and easier maintenance compared to traditional monolithic architectures.

Open-Source Software (OSS): Software that is released with a license that allows anyone to view, use, modify, and distribute the code.

Software Composition Analysis (SCA): The process of identifying and managing open-source components within a software project, including tracking vulnerabilities and licensing issues.

Static Application Security Testing (SAST): A type of security testing that analyses source code to find vulnerabilities without executing the code.

REFERENCES

https://security.googleblog.com/2021/06/introducing-slsa-end-to-end-framework.html

https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

https://about.codecov.io/security-update/

https://www.bcg.com/publications/2024/cloud-transformation-without-the-risk

https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/in-search-of-cloud-value-can-generative-ai-transform-cloud-roi

https://blog.gitguardian.com/codecov-supply-chain-breach/

https://go.crowdstrike.com/rs/281-OBQ-266/images/GlobalThreatReport2024.pdf

https://thenewstack.io/will-generative-ai-kill-devsecops/

https://www.csoonline.com/article/1311835/generative-ai-poised-to-make-substantial-impact-on-devsecops.html

https://www.accolite.com/news/an-assessment-of-how-gen-ai-has-begun-to-transform-devsecops/

https://securityboulevard.com/2023/06/chatgpt-spreads-malicious-packages-in-ai-package-hallucination-attack

https://www.techtarget.com/searchitoperations/tip/Generative-AI-use-cases-for-DevOps-and-IT

Generative AI in DevSecOps. Introduction Medium: by Bijit Ghosh

https://www.securityweek.com/vulnerabilities-being-exploited-faster-than-ever-analysis/

Zippia. "25 Amazing Cloud Adoption Statistics [2023]: Cloud Migration, Computing, And More" Zippia.com. Jun. 22, 2023, https://www.zippia.com/advice/cloud-adoption-statistics/

https://www.splunk.com/en_us/pdfs/resources/analyst-report/hbr-the-state-of-cloud-driven-transformation-2021.pdf

Gartner. "Forecast Analysis: Public Cloud Services, Worldwide, 2018-2024, 4Q20 Update."

https://www.nist.gov/itl/ai-risk-management-framework

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf

https://safecode.org/publication/SAFECode_CSA_Cloud_Final1213.pdf

https://safecode.org/wp-content/uploads/2018/03/SAFECode_Fundamental_Practices_for_Secure_Software_Development_March_2018.pdf

https://sma.nasa.gov/sma-disciplines/software-assurance-and-software-safety

https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html

https://www.ncsc.gov.uk/information/log4j-vulnerability-what-everyone-needs-to-know

https://www.techtarget.com/searchsecurity/news/252499956/Codecov-breach-raises-concerns-about-software-supply-chain

https://blog.sonatype.com/openai-data-leak-and-redis-race-condition-vulnerability-that-remains-unfixed

Kam, Hwee-Joo and D'Arcy, John, "A DEVOPS PERSPECTIVE: THE IMPACT OF ROLE TRANSITIONS ON SOFTWARE SECURITY CONTINUITY" (2023). ECIS 2023 Research-in-Progress Papers. 86. https://aisel.aisnet.org/ecis2023_rip/86

https://www.cvedetails.com/vulnerability-list/vendor_id-13554/Elasticsearch.html

https://www.cvedetails.com/vulnerability-list/vendor_id-20905/Prometheus.html

https://github.com/NeilAPerry/Do-Users-Write-More-Insecure-Code-with-AI-Assistants?tab=readme-ov-file

https://snyk.io/reports/ai-code-security/