IJSRA

International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

Int. J. Sci. Res. Arch.
IJSRA
International Journal of Science and Research Archive
Research Journal Archive, INDIA

(Review Article)

Check for updates

Integrating Security into CI/CD Pipelines: A DevSecOps Approach with SAST, DAST, and SCA Tools

Naga Murali Krishna Koneru *

Accenture Solutions Pvt. LTD, INDIA.

International Journal of Science and Research Archive, 2021, 03(01), 250-265

Publication history: Received on 17 June 2021; revised on 22 August 2021; accepted on 26 August 2021

Article DOI: https://doi.org/10.30574/ijsra.2021.3.1.0080

Abstract

Continuous Integration and Continuous Deployment (CI/CD), which was rapidly adopted by the software development industry, turned into a fast-paced process, causing new insecurity to be generated. This paper explains how we support the implementation of such DevSecOps by SDI (merging security in SD) with CI/CD process by combining SDI instruments of Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) and Software Composition Analysis (SCA) instruments. In this manner, security measures maintain equal development speed during development, while vulnerabilities are detected before their respective development stage ends. This research contributes scientific evidence with production use cases to demonstrate the usefulness of SAST, DAST, and SCA technologies in strengthening the effectiveness of CI/CD pipeline security. These tools are deployed so that the application can expose the security risks before the deployment dates, thereby ensuring that the application promotes security standards across the development teams. Security is embedded into core development procedures through DevSecOps, which performs security at each development stage rather than at the end. Risk reduction, trust levels, and compliance standards are augmented in the transition, and these are most critical in sectors that process sensitive information, such as retail and e-commerce. According to research data, security protection must be present before it comes to the market so that methods of protection can be implemented according to industry standards and meet the requirements of protecting digital systems from new cyber threats and vulnerabilities in a dynamically changing digital environment.

Keywords: CI/CD Pipelines; DevSecOps; SAST Tools; DAST Tools; SCA Tools; Retail & E-Commerce Security

1. Introduction

Recently, the Continuous Integration and Continuous Deployment (CI/CD) practice has been introduced for deployment in the exact order. CI/CD pipelines automate integration, start testing, and roll out the most reliable software faster. As this is what CI is all about, continuous testing and problems should be identified as early as the development cycle. CD manages to move to production to allow new features, new versions, and fixes to flow up with little delay and no disruption. It reduces the lead time of delivering the software. Thus reducing integration issues and, in that case, time to release to production per code change. For an agile development environment characterized by the ever-changing and transforming digital world, it is very important for organizations to continuously flow new features, such as CI/CD adoption, which constitutes an adoption of CI and CD. CI/CD practices provide extremely high speed and flexibility but also come with new security challenges. Manual testing and occasional audits cannot overcome CI/CD's ability to deploy rapidly. They stay open to attack, and users' trust is compromised because security becomes an afterthought.

It is DevSecOps, a crucial methodology of modern software development that came into being due to the increasing demand for integrating and applying security practices in CI/CD pipelines. An extension of traditional DevOps, DevSecOps positions security as a continuous process that is migrated from the end of the lifecycle to the beginning.

* Corresponding author: Naga Murali Krishna Koneru

When done correctly, DevSecOps keeps security a core theme of DevOps by including anyone and everyone in the development, security, and operations team. DevSecOps then creates security tools in the pipelines of the CI/CD, helping in the earlier detection of vulnerabilities to minimize the risk of security issues and the time spent resolving security problems. When integrated into a pipeline, SAST, DAST, and SCA provide organizations with the detection and prevention of vulnerabilities before the code gets to production. By taking this proactive approach to security, more efficient, cost-effective risk management can be made, and compliance with industry standards and regulations is maintained.

DevSecOps becomes more vital as the software development cycles keep on increasing. It ensures that security is_cached during the entire development process in fear of the rapid release of insecure applications that can damage bad business and reputation. This is especially important for those in charge of Retail and E-Commerce operations as adopting a robust security framework like DevSecOps in CI/CD pipelines is. Because retail and e-commerce platforms are held to be the ones to handle the most sensitive customer data, such as payment details, personal information, and purchasing history, they have become prime targets for cybercriminals. Data breaches, financial losses, irremediably lost customer trust, and any security vulnerabilities in the platform can occur.

With the retail and e-commerce sectors becoming increasingly digitally focused, secure, adaptable, and scalable software environments become necessary. DevSecOps practices within the CI/CD pipeline help the developers implement security within the development process to mitigate the risk of data breaches, fraud, and any compliance issues. Retailers can proactively identify vulnerabilities in the application, code, and dependencies and help customers have a secure, seamless online experience while protecting their brand and assets from possible cyber threats.

This article aims to bring new security to the CI/CD pipelines through a DevSecOps approach dedicated to retail and e-commerce. In the end, its goal is to provide technical information as to how Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA) instruments ought to be utilized in a CI/CD pipe to reinforce security and diminish vulnerabilities. In addition, this article will provide an overview of CI/CD practices and then talk about Devsecops and the place it plays in the development lifecycle. Then, it will explain SAST, DAST, and SCA tools and how they can be embedded in a CI/CD pipeline to detect and prevent security risks in real-world scenarios. This article will also discuss best practices for integrating security in CI/CD pipelines in retail and e-commerce environments. The article will end with a closer look at some of the identified trends to work towards bettering security in today's development environments. At the end of this article, Leaders of Retail and E-Commerce can know how to incorporate security in their CI/CD cycles to make their platforms secure, reliable, and by industry standards.

2. Background and Context

2.1. Evolution of CI/CD Practices

Continuous Integration and Continuous Deployment (CI/CD) have become the norm in software development by introducing fast iteration methods to develop applications. CI/CD practices involve code change that is integrated continuously into the shared repository, and the result of automated builds and tests are performed on each contribution. The integration is performed frequently to minimize the risk of conflicts and the complexity of merging large code bases. Build and automated test suites are automated to catch errors early and keep teams in front of problems before they grow into production ones. With the introduction of an agile and iterative model, CI/CD emerged as the key enabler of the development and product quality at an accelerated speed (Boda, 2020). Advantages include better code quality, quicker feedback loops, and deployment with the confidence of periodic incremental updates. In addition, CI/CD allows automated processes to be inserted in every facet of the software cycle, leading to a building divisive cultural improvement. Consequently, these teams are directed to innovate quickly. Similarly, they can retain their stability, which is not good for some industries where the products are based on the needs and conditions of the application markets.

2.2. Understanding DevSecOps

The DevSecOps model is an evolution of the classic DevOps model, wherein such security procedures are woven as part of the development and operations procedure. In other words, with this approach, security is considered a must from the beginning and not an afterthought. As software development has speeded up and cyber threats have become more sophisticated, the need for security to integrate into agile environments has become more imperative. By shifting left and bringing security into the development process earlier in the business application lifecycle, vulnerabilities can be identified and remediated before reaching production. It puts the onus on issue forethought to minimize the risk of security breaches and to simplify the fixes when the time comes (Debois, 2012). In addition, DevSecOps promotes a change in an environment where developers, operation teams, and security professionals all share the responsibility to keep a secure environment. It encourages continuous learning and the common goal to protect sensitive data and systems. It enables running the approach in the CI/CD pipeline, such as automated security testing, real-time vulnerability assessment, and compliance check, to keep the security tempo with the deployment tempo. In conclusion, DevSecOps is a framework of speed and security, putting together speed and security to help an organization accomplish business goals without compromising safety.

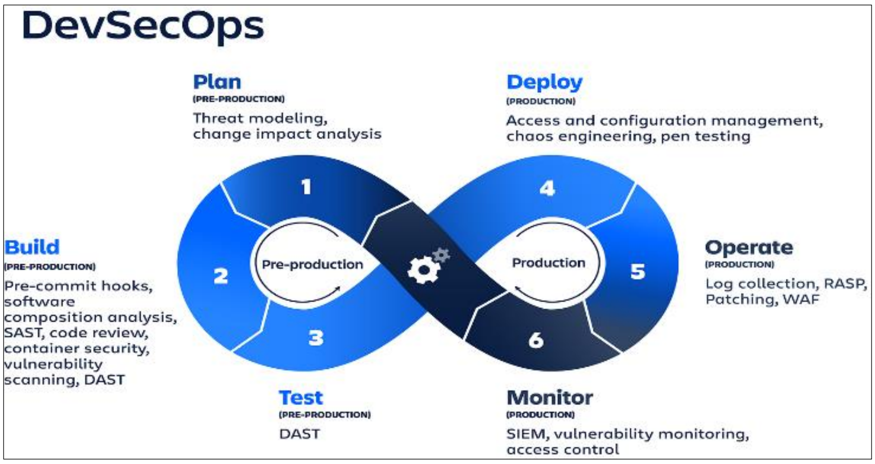

DevSecOps
Plan
(PRE-PRODUCTION)
Threat modeling, change impact analysis
Deploy
(PRODUCTION)
Access and configuration management, chaos engineering, pen testing
Build
(PRE-PRODUCTION)
Pre-commit hooks, software composition analysis, SAST, code review, container security, vulnerability scanning, DAST
Operate
(PRODUCTION)
Log collection, RASP, Patching, WAF
Test
(PRE-PRODUCTION)
DAST
Monitor
(PRODUCTION)
SIEM, vulnerability monitoring, access control
1
2
3
4
5
6
Pre-production
Production

Figure 1 An Overview of DevSecOps

2.3. Overview of Security Testing Tools

In modern software development, a range of security testing tools are used within the CI/CD pipeline for a specific purpose. Static Application Security Testing (SAST) tools are the first to analyze source code, bytecode, or binaries before those have been executed. They detect SQL injection, XSS, and Buffer overflows at the code level (Mell & Scarfone, 2005). SAST ingested into the development process means developers will get immediate feedback on what could be scraper security issues so they can fix those issues before the code is merged or deployed. Dynamic Application Testing (DAST) tools are used to test running applications using real-world attack scenarios. DAST tools provide application audits with the help of DAST tools by interacting with the application's interfaces to respond to issues like authentication flaws, insecure configurations, and runtime vulnerabilities. Real-time testing is needed to understand how an application behaves in attacked conditions. Software Composition Analysis (SCA) tools also help manage the third-party components and open-source libraries embedded into modern software (Mackey, 2018). These dependencies are continuously scanned by SCA tools, looking for known vulnerabilities, license compliance issues, and risks brought in due to external code. These tools bring the complete life cycle approach that provides security for the whole development life cycle. A combination of integrating SAST, DAST, and SCA tools within CI/CD pipelines offers a multilayered defense strategy focused on the static and dynamic security of the application.

2.4. Importance for Retail & E-Commerce

Retail and e-commerce platforms are part of this environment, and deploying and updating are commonly done quickly. These are vulnerable, especially to security risks, since the customers' data are very sensitive and there is a lot of sensitive customer payment information. As it is, the retail systems need to possess a high throughput of transactions, even when accessing a silkily user-friendly experience without compromising security. Vulnerabilities of CI/CD pipelines are common targets for cyber attackers, as they may be used to exploit insecure integrations or unpatched third-party libraries (Jawed, 2019). Billion-dollar breaches in retail systems have been demonstrated to give rise to steep financial losses, damage to brand reputation, and loss of customer trust. For example, an e-commerce platform that either does not sufficiently secure its CI/CD process or fails to secure it will inadvertently lead to exposure of the customer credentials or data related to payment, exposing the platform to any regulatory penalties and loss of consumer confidence. It follows that it is not only a technical necessity but an essential business requirement to integrate robust security testing tools and practice DevSecOps. This allows the continuous monitoring and automated vulnerability assessments of critical systems, which in practice are beneficial for retail and e-commerce organizations. These organizations are embedding security into each phase of the development, ensuring they adhere to the industry standards and creating a solid operational framework (Bansal, 2020). These practices allow retailers to adopt the new practices swiftly and gain an edge in overcrowded markets.

All these add up to a shift in the way of thinking about the software development paradigm when it comes to CI/CD practices, the emergence of DevSecOps, and the use of security testing tools. Importantly, this comprehensive approach is crucial, especially for the retail and e-commerce platforms, because vulnerabilities in security can have a significant effect. By integrating automated security measures through continuous integration (CI) and testing these security controls from the start, companies secure sensitive data, ensure that they adhere to compliance requirements, and preserve operational integrity at the highest level (Nyati, 2018). The integration of these advanced practices strengthens the overall security posture. It contributes to continued business growth by enabling applications to remain reliable, secure, and ready to address market evolution.

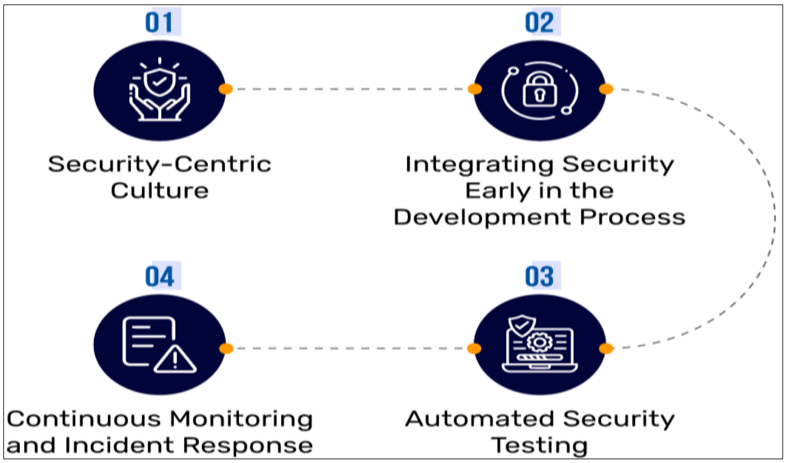

01
Security-Centric Culture
02
Integrating Security Early in the Development Process
03
Automated Security Testing
04
Continuous Monitoring and Incident Response

Figure 2 Implementing DevSecOps to Secure Your CI/CD Pipeline

3. Technical Implementation of a DevSecOps Pipeline

3.1. Integrating SAST Tools in CI/CD

Integrating Static Application Security Testing (SAST) tools into a continuous integration / continuous deployment (CI/CD) pipeline is one of the important steps in finding the vulnerabilities at the code level before the software gets deployed. This integration is an example based on the widely adopted SAST tool SonarQube. In CI/CD pipelines, SonarQube is configured to be automatically run on code commit and pull requests, looking for common security issues like SQL injections, XSS, and buffer overflows on source code (Jose, 2020). To run quality scans, one must work through the configuration process: set up SonarQube servers, integrate the SonarScan build process, and define quality gates (minimal security standards).

A YAML snippet of a practical configuration example is provided. The pipeline is broken down into several stages in the sample configuration, including an 'analyze' stage where SonarScanner is run. The YAML snippet defines that analysis is launched upon merge requests and changes to the main branch, keeping the code under constant hacking for vulnerabilities. By integrating this early, security flaws will not make it to production, and their risk will be reduced. SonarQube gives developers detailed reports of what type of vulnerable code SonarQube finds (Paananen, 2016). It is necessary in environments where rapid delivery cycles demand online feedback on code quality and the security posture on a go-live basis.

3.2. Deploying DAST in a Live Environment

Dynamic Application Security Testing (DAST) is an in-built CI/CD pipeline feature that tests a security application when it runs. For this purpose, the OWASP Zed Attack Proxy (OWASP ZAP) is a very robust tool. OWASP ZAP simulates actual attacks against an app in question and discovers flaws like a missing configuration in authentication, insecure settings, and runtime problems that cannot be spotted in a static code in this analysis. The implementation will be done by deploying OWASP ZAP in the staging environment, where the application is deployed in a state as close to the production as possible.

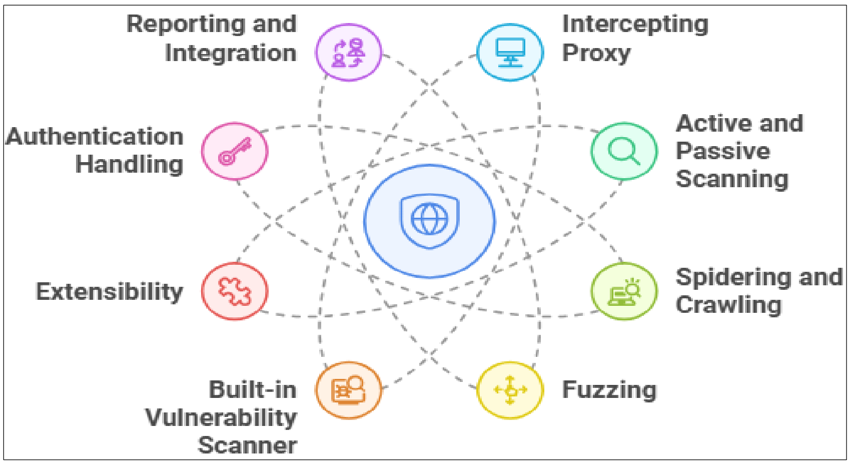


Figure 3 Key Features of OWASP Zed Attack Proxy (ZAP)

An OWASP ZAP integration with a complete git ci/cd pipeline walkthrough from 'scan', not just configuration but also background, is described. At this stage, the OWASP ZAP baseline scan command is used to run a staging URL to expose vulnerabilities, and a report is returned in HTML with a summary of the vulnerabilities (Van Rensburg, 2017). In this configuration, no matter the findings generated from the report, the generated report is stored as an artifact so that the developers and the security teams can see exactly those findings. This integration also provides an understanding of the operational security of the application and the locating configuration problems that become evident only at runtime. Security testing is taken care of by the CI/CD pipeline by automating the execution of OWASP ZAP and ensuring that security testing happens once the CI/CD pipeline fires in working; we are in the middle of a release passing new features or updates.

3.3. Incorporating SCA Tools for Open-Source Management

Today, SCA (Software Composition Analysis) tools are integral to modern CI/CD pipelines that analyze open-source dependencies and third-party libraries. These tools are essential because they find known vulnerabilities in application components being integrated and verify that dependencies do not unintentionally expose the application to security vulnerabilities. One commonly used SCA tool is Dependabot, which automates the dependency tree of any app and notifies Developers about outdated or vulnerable libraries.

In practical terms, Dependabot is configured to automatically notify when dependency updates are found within the CI/CD pipeline. The configuration involves defining a job in the pipeline to run a dependency scan using Dependabot's container image. The job runs the command to refresh dependency information, then checks out the repository to compare the dependency information with that stored in Jeweler. The dependency analysis is automated so that the analysis of dependencies is done with each build cycle and added vulnerabilities in third-party dependencies are immediately flagged to be remediated (Stringer, 2020). This is an example of continuously monitoring the security of the software supply chain integration with tools like Dependabot in the CI/CD process, which increases the likelihood of not introducing insecure components to the production environment.

3.4. Challenges and Best Practices for Tool Integration

The implementation of SAST, DAST, and SCA tools inside CI/CD pipelines delivers better security results yet brings unique management issues that organizations need to handle correctly. Organizations encounter a fundamental issue when deciding between development speed, security rigor, and operational productivity. New automated security testing implementation often causes longer development timelines and potential stoppages within the development production cycle. Organizations must develop strategies to improve integration process efficiency to manage associated performance consequences. The report states that tool integration success requires choosing appropriate software tools alongside proper configuration that enables their operation without disrupting the development process. Security best practices support the simultaneous execution of security tests because SAST, DAST, and SCA processes operate in parallel rather than following each other (Tirosh et al., 2019). The method shortens feedback times without sacrificing the complete security stance. During SAST tool deployment, developers must adjust the quality gate configurations to maintain appropriate security requirements without disrupting ongoing development activities. Security professionals should collaborate with developers to refine security thresholds and eliminate misleading positive results that create productivity hurdles.

The report emphasizes the critical need for permanent teamwork between teams working on development, operations, and security functions. CI/CD pipeline security requires organizations to adopt cultural changes toward broad security collaboration among teams. Security teams should implement regular training programs alongside cross-functional meetings and unified dashboard systems to help teams communicate better. The report demonstrates how automated remediation methods incorporating automated dependency updates and patch management help developers work more efficiently and expedite the repair of detected problems. The document discusses integration difficulties when multiple security tools need unified implementation through a harmonized pipeline system. Adopting standardized YAML scripts and centralized tool setting management will help simplify the integration process (Morris, 2016). Organizations should establish complete documentation of their CI/CD pipeline architecture, which maps the connections of security tools along with the complete workflow process. The documentation provides essential information for fixing technical problems and training new team members.

The report demonstrates that implementing DevSecOps security testing initially delays the development cycle, yet the prolonged security benefits yield greater value than brief short-term expenses. Security vulnerability reduction, better regulatory standard compliance, and superior software quality result in a robust and resilient production environment. Communities that integrate SAST, DAST, and SCA tools with their CI/CD pipeline make security development essential to their strategic software development lifecycle goals (Williams, 2019).

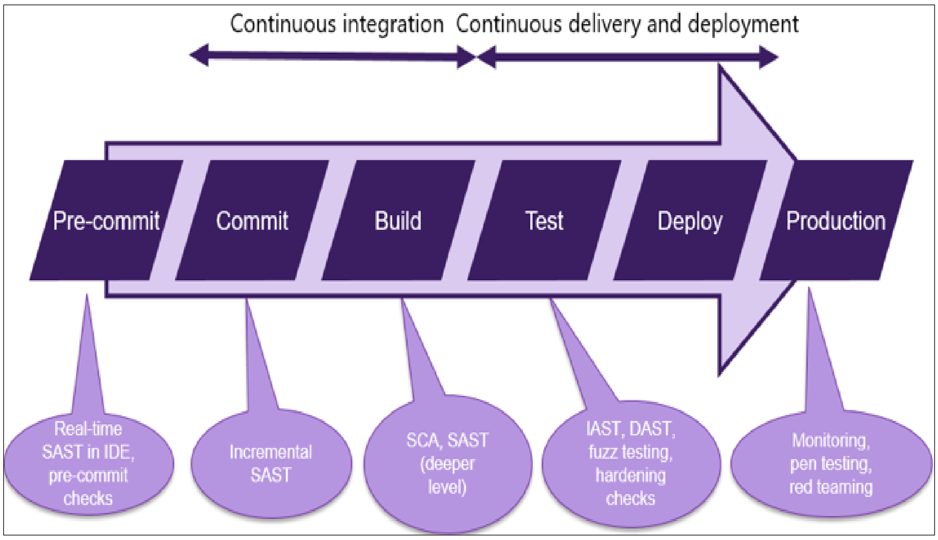

Continuous integration
Continuous delivery and deployment
Pre-commit
Commit
Build
Test
Deploy
Production
Real-time SAST in IDE, pre-commit checks
Incremental SAST
SCA, SAST (deeper level)
IAST, DAST, fuzz testing, hardening checks
Monitoring, pen testing, red teaming

Figure 4 Application Security Tools in the CI/CD Pipeline

4. Retail & E-Commerce Operations Leader Focus

4.1. Security Concerns Specific to Retail and E-Commerce

The security of retail and e-commerce operational spaces is different from that of other businesses. They require stricter technical and operational methods. Securing handling of customer data and payment processing in the digital marketplace is prehistoric. Sensitive information such as personally identifiable information (PII), credit card details, and transaction records are all an organization in this sector must protect. Robust encryption methods and secure communication protocols are required. There are encryption standards, such as AES, or protocols like SSL/TLS, to guarantee the correct transmission of messages or data storage in confidential form. System architectures include multi-factor authentication (MFA) and secure key management to decrease the probability of unauthorized access (Phan, 2018).

Along with technical defenses, retailers and e-commerce entities must comply with stringent regulatory and compliance rules. To deal with such legislation, such as the General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), or Payment Card Industry Data Security Standard (PCI DSS), firms have implemented powerful controls on the way they handle data (Seaman, 2020). These regulations are mandated strictly, and there are no exemptions from the standards outlined; failing to abide by them can come with strict financial penalties on a company and solidly tarnish its reputation. For this reason, these companies spend much money on audits for regular security, assessments of vulnerabilities, and constant monitoring of their systems. Such practices ensure that deviation from prescribed security standards is promptly identified and remediated, thus improving the security posture.

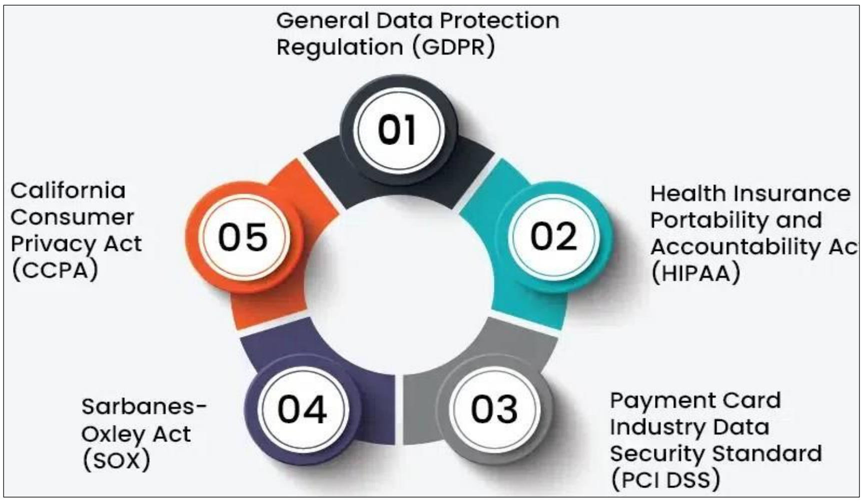

General Data Protection Regulation (GDPR)
01
Health Insurance Portability and Accountability Ac (HIPAA)
02
Payment Card Industry Data Security Standard (PCI DSS)
03
Sarbanes-Oxley Act (SOX)
04
California Consumer Privacy Act (CCPA)
05

Figure 5 Data protection: Data Protection Laws and Regulations

The emergence of cyber threats is moving too fast for retail operations and e-commerce to remain reactive to such levels of security threat. So, it is not the case that there is no problem. Malicious actors are still looking for vulnerabilities in online payment systems and customer data repositories. Security testing automation tools should be integrated into the Continuous Integration/Continuous Deployment (CI/CD) lifecycle (Deepak & Swarnalatha, 2019). These tools give continuous feedback to development teams regarding where to remediate these vulnerabilities early before they can be exploited. With these retail and e-commerce leaders focusing on secure data handling and regulatory compliance, they protect their customers and ensure a setup that lays the foundation of trust essential to ensure long-term business success.

4.2. The Impact of CI/CD Security on Retail Operations

CI/CD security practices added to the retail operation dramatically affect operations performance and business weather. By incorporating security tools such as Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA) into the CI/CD pipeline, these organizations can find vulnerabilities early (Abigail, 2020). Deploying insecure code puts it at risk of being deployed to production environments and would also result in a costly breach. This helps promptly identify weaknesses, and the remediation can be accomplished rapidly. By doing so, retail system operations remain operationally efficient, even as the company quickly rolls out new features and updates.

From a customer's point of view, security in CI/CD practices not only matters for safety reasons but also greatly contributes to increased user experience and trust. The better the perception of a brand from the consumer's point of view, the more reliable customers feel about their data being protected and the transactions being processed securely, and the higher the perception of the brand. It shows in customer retention rates and sales as customers have more trust in secure applications. Retailers prioritizing robust security strengthen their defense mechanism against loss, meaning they will suffer fewer downtime and service interruptions. Such operational benefits lead to increased customer satisfaction and a more competitive position in the market.

Real-world examples from the retail sector also underscore the importance of integrating security into CI/CD pipelines. Many such case studies show that those organizations that have adopted full security testing procedures have greatly reduced the number of vulnerabilities and breaches (Felderer et al., 2016). They are not anecdotal success stories; indeed, empirical evidence is offered for the significant effect of a properly implemented security strategy on system resilience. Implementing CI/CD security practices has also allowed retail companies to transition quickly from the development process without compromising safety and bring innovations to the market faster while maintaining robust security against new cyber threats.

4.3. Aligning DevSecOps with Business Objectives

In retail and e-commerce operations, this level of strategic DevSecOps in CI/CD pipelines gives an edge to the competition. If security practices are embedded into the development lifecycle, organizations can be sure that every new feature or update comes with rigorous security checks. This approach reduces the risk of vulnerabilities and helps the business to remain agile to deploy rapidly without the burden of excessive post-release security fixes. Secure CI/CD allows organizations to strike a balance between creativity and risk management and guarantees that the fast advancement of new technologies is not to the detriment of security (Marini-Wear, 2019).

The cultural view of shared responsibility is a consequence of implementing DevSecOps, which involves a technical change in the approach of the traditional software development model—the part where security is not isolated but is integrated into every phase of the Dev progress cycle. SAST, DAST, and SCA are automated and are continuously executed, and they are quick to give feedback to developers (Jawed, 2019). Finally, this integration causes the codebase to be consistent and reliable but resilient against new threats. Automation in security testing also reduces manual overhead. Available resources are then freed up to allow them to be invested in strategic business initiatives. This engineering leads to an overall result of a robust and scalable infrastructure that can support sustained growth and operational excellence.

To use DevSecOps methodology, IT and operations teams coexist with the security team. The cross-functional aspect of this means that security is part of the planning strategy and daily operations. Shared accountability and regular communication lead to a sense of being an enabler and not a hindrance (Kim et al., 2017). As such, this alignment translates to retail leaders enabling continuous operations, rebuilding customer trust, and mitigating risks associated with digital transformation programs. Retail and e-commerce organizations can achieve a harmonious balance if DevSecOps aligns with overarching business objectives, thereby providing retail and e-commerce organizations with long-term success.

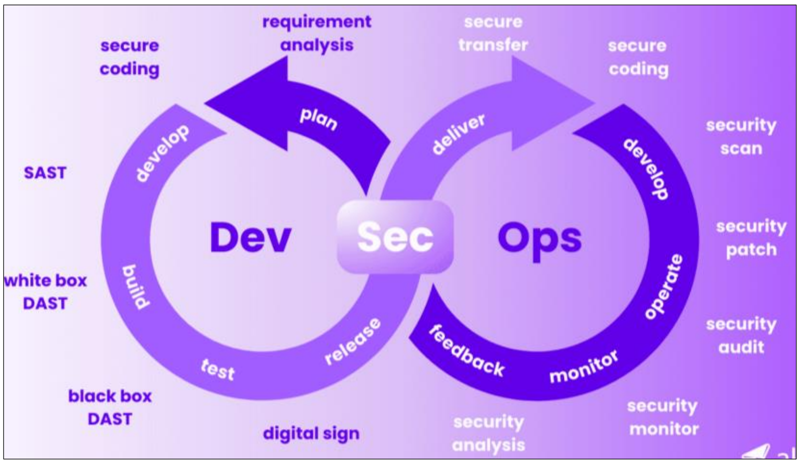


Figure 6 Best Practices for DevSecOps

5. Best Practices for Integrating Security in CI/CD

One of the parts of modern software development is that security practices are integrated within CI/CD pipelines.

5.1. Establishing a Security-First Culture

Setting up a security-first culture is fundamental in enabling security to be integrated into CI/CD pipelines. The development and administrative teams encourage investments in broad, in situ, and ongoing training. This way, not only will all team members know about possible weaknesses, but it will hold all the organization's members accountable for its security. Regular security workshops, simulated attack exercises, and certification, which include the latest security threats and defenses, should be implemented by the leaders in technology and operation to advice technologies and operation leaders. When it comes to customer data and financial transactions, as in the cases of retail and e-commerce, the imperative to build a security-first mindset gets amplified. A security culture first relies on proactively identifying and remedying vulnerabilities, resulting in less likelihood of breaches and more overall integrity of the system (Ayereby, 2018). This increases trust from the various stakeholders and compliance with regulatory standards, as well as reduces the risks experienced during the rapid development cycles.

5.2. Automation and Regular Testing

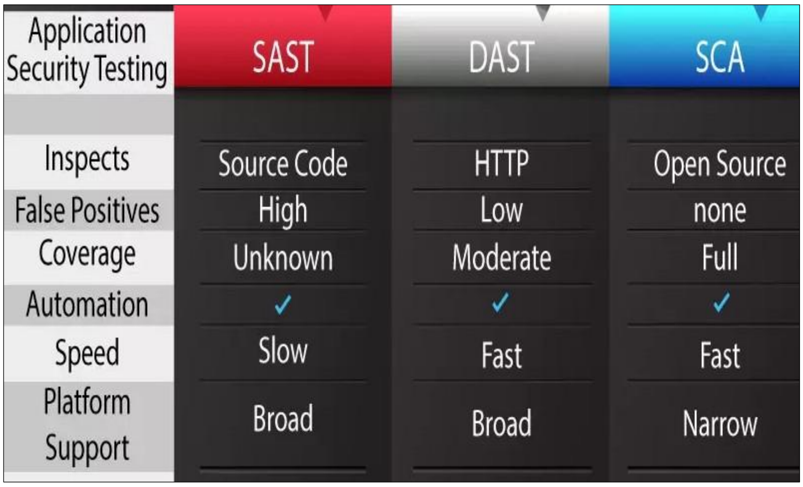
Security integration in CI/CD pipelines cannot be accomplished without automation. Integrating automated scanning and alerts into the development process is an important step to allow vulnerabilities to be identified early and remediated quickly. In this practice, the SAST, DAST, and SCA tools are appointed to schedule regular scans to protect against changes in the code, configuration, and dependency. An example is that organizations define automated workflows in the popular CI/CD platforms to run security testing whenever the code commit, pull request, or deployment is performed (Shajadi, 2019). This kind of automation reduces the room for human error and shortens the feedback loop, meaning that fixes will be done before deployment to production. In addition to improving development pipeline security, automation plays a role in operational efficiency. Taking advantage of these automated processes, retail and e-commerce companies, part of the e-commerce industry, which usually work under strict deadlines and high customer demand, can achieve agility without jeopardizing the security factor. By embedding automated testing into CI/CD workflow, organizations bring the scale and agility of an automated security infrastructure.

5.3. Continuous Monitoring and Feedback Loops

Continuous monitoring is critical to detect if security measures are not continuously effective. Creating feedback loops between operations, security, and development teams allows one to set up real-time vulnerability management and proactive risk mitigation. This involves monitoring practice that uses tools to track the system performance and report abnormal activities to alert to possible security incidents. This makes it possible to keep tracking breaches or erratic patterns continuously and act accordingly quickly, quickly performing incident response and remediation. Moreover, the feedback loops help incorporate the lessons learned from the previous vulnerabilities in future development cycles.

For example, after a security incident or periodic audits, all stakeholders should receive detailed reports to improve strategies and policies. As in the case of cyber-attacks, such as social engineering, continuous monitoring offers the last line of defense when it can be imposed in retail and e-commerce settings, where the impact of a security breach can be quite severe for both the business and the customer. Eventually, real-time feedback and ongoing monitoring permit organizations to dramatically change to the evolving threat while keeping an immaculate and tough CI/CD pipeline.

5.4. Tool Selection and Integration Strategies



Application Security Testing	SAST	DAST	SCA
Inspects	Source Code	HTTP	Open Source
False Positives	High	Low	none
Coverage	Unknown	Moderate	Full
Automation	✓	✓	✓
Speed	Slow	Fast	Fast
Platform Support	Broad	Broad	Narrow

Figure 7 SSAT, DAST, and SCA Applications Security Testing

Choosing the right security tools to help build a successful security integration into CI/CD pipelines is an integral task. Therefore, given technical capabilities, ease of integration, scalability, and support for the organization's specific development environment, SSAT, DAST, and SCA tools should be implemented systematically by the organizations (Nimmo, 2018). These tools should be evaluated on a clear set of criteria for finding various vulnerabilities and actionable insights while working well in existing CI/CD pipelines. Also included in the tool is its compatibility with various programming languages, support for the containerized environment, updates with regularity, and community (Watada et al., 2019). Retail and e-commerce tend to rely on third-party libraries and frameworks often; therefore, the selected tools must manage proprietary or open-source components in a retail and e-commerce environment. Tools are integrated into the CI/CD pipeline in two parts: an idea of optimal integration and a phased approach, with the tools checked in a controlled environment before being installed across the entire CI/CD pipeline. Rolling this out slowly will allow the organizations to tweak the tool configurations and alert threshold tuning without deleterious effects on development speed or operational efficiency during integration.

5.5. Practical Examples and Lessons Learned

Gaining proper security integration into CI/CD pipelines is hugely beneficial, and drawing upon practical examples from retail and eCommerce applications can make a big difference. Most organizations adopting a DevSecOps approach using the SAST, DAST, and SCA tools have reported better results (Morales et al., 2020). For instance, SonarQube was successfully integrated as a component of the CI/CD pipeline of a leading e-commerce platform that eliminated its backlog of vulnerabilities by flagging potential security issues automatically for every code commit. A second retailer deployed OWASP ZAP on a staging instance and quickly picked up on configuration errors and runtime vulnerabilities. They point out that we need to understand common pitfalls such as false positives or tool misconfigurations, and as a note, point out the need for continuing training and refinement of the tool.

The lessons learned from implementing these best practices are that development security and operations teams must work together. The opportunity to review and refine security protocols among departments and adjust security protocols through regular debriefs makes this time valuable (Fan et al., 2016). In the long run, a real methodical approach based on practical CICD experiences and always-changing best practices helps to create a more secure, faster, and more resilient CI/CD pipeline that can adapt to the volatile nature of modern software development.

6. Empirical Results and Discussion

6.1. Summary of Empirical Data from DevSecOps Implementations

Physical evidence shows that DevSecOps has improved security posture and CI/CD pipeline efficiency across several organizations. Metrics specific to the security aspect, such as the number of vulnerabilities to be detected and how quickly to fix them, give clear indications of how strong the integration of the security practices as part of the CI/CD pipeline is. Organizations adopting DevSecOps reduce 30-50% of security vulnerabilities in production (Heilmann, 2020). This improvement is partly because of SAST, DAST, and SCA tools, which require being proactive and detecting critical flaws in the earlier phase of the development life cycle, making it easy to reduce the risks that critical flaws get to the production environment.

Key performance indicators (KPIs) such as the "mean time to detection" (MTTD) and the "mean time to resolution" (MTTR) were spiked when DevSecOps was implemented. An average 40% reduction in MTTD (time from a commit to its being seen) was achieved. Among many organizations, MTTR, the time to fix the vulnerability after detection, was reduced by 35%. This also leads to higher adoption of automated security scanning within the CI/CD pipeline, thus ensuring that the security standards being met are expressed as OWASP Top 10 vulnerabilities and any other industry-specific security controls. This resulted in DevSecOps implementations that resulted in better visibility into security risks and vulnerabilities being addressed timely and proactively.

6.2. Impact on Vulnerability Reduction and Developer Productivity

Including security tools and their integration into the CI/CD pipeline affects vulnerability reduction, especially with large, complex retail and e-commerce platforms. Among several case studies, a notable result is reduced setup vulnerabilities (issues that can result in data breaches or stronger security-related incidents). In one retail deployment, automated SAST tools, SQL injection, and cross-site scripting (XSS) vulnerabilities were reduced by between 60% and 70% (Rodríguez et al., 2020). When combining DAST tools like OWASP ZAP with manual testing, organizations have also found that 45 percent of runtime vulnerabilities, such as authentication and session management flaws that always tend to be missed by manual testing, are reduced.

Considering their productivity effect, developers have enjoyed a net positive gain, though some saw security testing slowing down development cycles. Several organizations have pointed out that continuous improvement (CI/CD) pipelines with automated security checks improved developer productivity. It enabled developers to detect and fix vulnerabilities at the development stage, thus reducing the need to perform cost-intensive rework in later testing or post-production stages and speeding up the release cycle. For instance, time spent manually looping through security reviews was cut in half as SAST, SCA, and other automated tools gave developers fast feedback so they could resolve issues without becoming blockers (Smith, 2016). Also, assuming responsibility for security by development and operation teams brought about a culture of shared responsibility, enhanced collaboration among departments, and sped up the resolution of security problems.

6.3. Balancing Security and Deployment Speed

DevSecOps adoption is an arduous journey when the balance between speed of deployment and the need for good security is in question. Security integration to the CI/CD pipeline aids in reducing vulnerabilities and becoming more compliant; it may cause delays unless appropriately tuned. In the early iteration, some organizations saw less smooth deployment times as security scans and checks were added. Both the challenge and its instances of the highest magnitude were acute in those environments with the highest releases, namely retail and e-commerce platforms, where changes can be deployed as often as every day or even several times per day.

A set of optimization strategies was applied to reduce the penalty on the deployment speed. One such strategy is the parallelization of the CI/CD pipeline. Teams started parallel scanning, in which various security tools (SAST, DAST, and SCA), albeit in parallel on different application parts, cut down on overall scan time (Dashevskyi, 2017). Besides this, SonarQube and OWASP ZAP were tuned to scan only crucial code parts or newly changed code to reduce the overhead. An additional optimization meant prioritizing security tests to match the application's risk profile. High-risk components (such as payment gateways or authentication mechanisms) were tested slightly more thoroughly and at greater intervals than low-risk ones. Risk-based testing allowed retail and e-commerce platforms to have high velocity yet maintain the critical vulnerabilities to be identified and resolved in time.

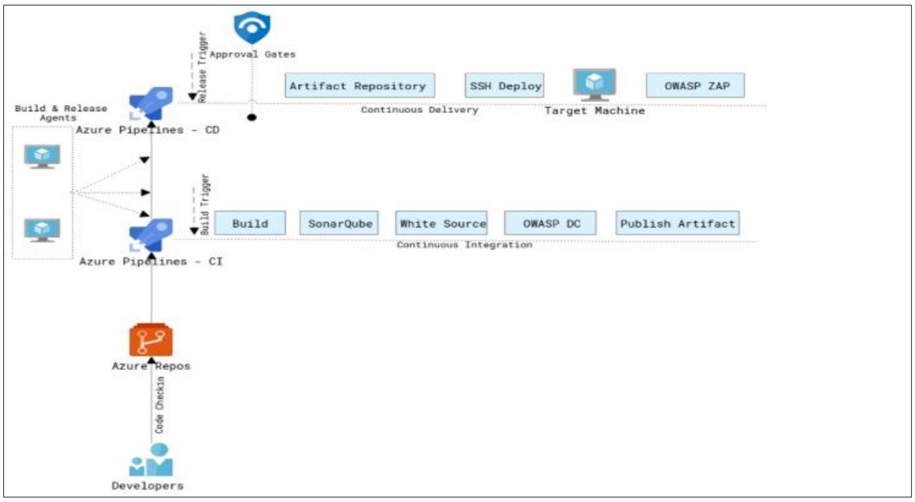


Figure 8 Injecting security in CI/CD pipelines with SonarQube, OWASP DC and OWASP ZAP

6.4. Discussion on Lessons Learned and Feedback from Retail Deployments

Although several lessons were learned over the process, feedback from retail and e-commerce deployments has been overwhelmingly positive. Choosing the right tools for the business requirements was one of the key lessons learned. With constantly changing technology stacks and existing platforms encompassing feature-rich e-commerce and retail platforms, security tools for these platforms should be conducive to integrating these tools into the existing CI/CD pipelines. Organizations that chose technologies that can be deployed flexibly and at scale, such as SonarQube, which can merge with various CI/CD systems, had it much easier to deploy security at scale (Candel, 2020).

Continuous education and training were another lesson. Initially, many development and operations teams and fast-paced retail resisted the extra overhead of security scanning. After initially opting out of taking security into the workflow, it became tangible, with the security reducing the incidences and the hotfix count needed post-deployment. Organizations that made a regular DevSecOps investment for developers and operations had easier transitions and quicker uptakes on security practices.

Among the most interesting things that people responded with is that security should not be viewed as a bottleneck or separate process to development. DevSecOps is an enabler that delivers secure and quick delivery (Ahmed, 2019). For retail deployments, security testing moved to the top of the "shift left" strategy that integrates security into the early stage of development rather than in the last. It helped turn the security culture from reactive to proactive and improve the organization's overall security posture.

7. Future Considerations and Trends

7.1. Emerging Security Technologies in CI/CD

The reality is that CI/CD pipelines are becoming more and more critical as the security landscape of CI/CD pipelines emerges in today's cutting-edge software development. Advances in artificial intelligence (AI) and machine learning (ML) have automated the detection of potential security flaws by a system that has accomplished vulnerability detection. Modern AI/ML algorithms can do so well with codebases and system behavior so large (such as codebases that have something like 5,000 paths and system behavior such as running), along with intelligently analyzing wide swaths of it to discover small anomalies or subtle patterns that could indicate vulnerable behavior. These systems are given historical data, which they learn and continuously update their threat models to know the known and new security risks. This allows organizations to run real-time vulnerability assessments in the initial development process and avoid manual code review dependency. AI-enabled security tools can provide actionable insights, remediation steps, and the ability to prioritize problems by severity. These technologies must become extremely precise in order for them to achieve this precision in order for these technologies to reach this extreme precision (Baumann et al., 2026). Consequently, these will become key security technologies in protecting a CI/CD environment from ''sophisticated'' cyber threats.

7.2. Future-Proofing CI/CD Pipelines

Two (or more) things are equally important for future-proofing CI/CD pipelines in order to be able to have continuous security in the face of changes in cyber threats and changes in regulations. Organizations must adapt to dynamic standards of compliance and expect new vectors of attack in order to protect continuous integration/deployment. Proactive ideas also involve integrating threat intelligence platforms that, without data from multiple sources, foresee risky rises and take the right measures to protect the systems. Continuous security and security assessments, regular audits, and policies that adapt to CI/CD pipelines are all required. Enterprises need flexible frameworks that can be quickly added with new security measures as they evolve. Automated compliance checks check code, configuration, and dependencies against current regulatory requirements to reduce non-compliance risk. This accelerates continuous improvement in security management such that improved security is introduced over the continuous development workflow. This ensures that organizations are prepared for today's events and that regulatory changes will threaten them.

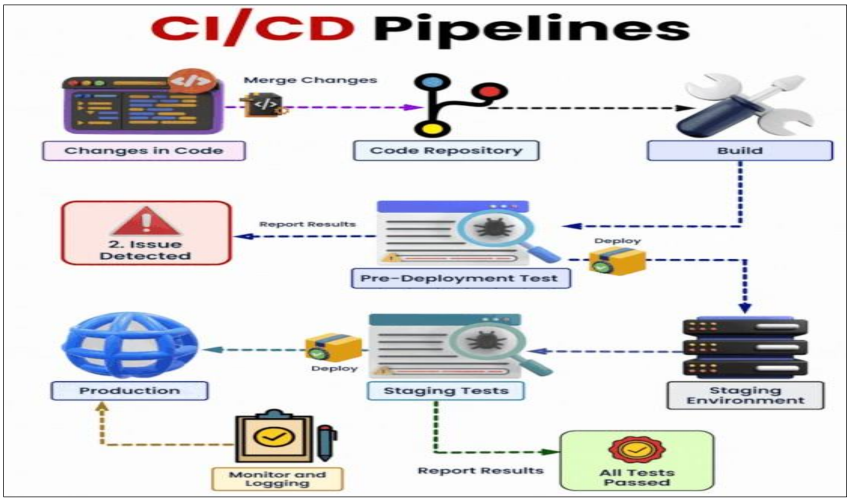


Figure 9 An Overview of CI/CD Pipelines

7.3. Integrating Advanced Analytics and Monitoring Tools

A big step towards proactive security management is advanced analytics and monitoring, which were then integrated into CI/CD pipelines. Organizations can use big data and predictive analytics to ''eat'' extremely large volumes of operational and security-related information, and by applying this overwhelming amount of input, they can identify patterns that are indicative of underlying vulnerabilities. Predictive models are now being built using these historical incident data and real-time performance metrics to give early warnings on anomalous behavior, which may lead to a security breach. These CI/CD environment analytics enable them to provide information about how the system is performing, the users' activities, and deviations from the normal operational baselines (Kumar, 2019). Normally, the security teams will require visualization platforms, which also work as dashboards to present very complicated data in an understandable form so they can make decisions on the data they have obtained. These systems can predict possible problems that may or may not occur and have the means to detect such problems well before major issues that would otherwise necessitate drastic remedy measures. Integrating advanced analytics into organizations' CI/CD pipeline can bring the above two capabilities to deployment — proactive defense posture with a high probability of success, low risk to process, and most optimized ops.

7.4. Strategic Roadmap for Retail & E-Commerce

To deploy the CI/CD security roadmap for future CI/CD security challenges, retail and e-commerce operation leaders are critical. Customer data and online transactions are sensitive; innovation and advanced security will be needed in the retail industry as this is very competitive. Leaders could start by automating the vulnerability detection process and cleansing the remediation by integrating AI/ML-based tools for automating the same (Opderbeck, 2019). The above would enable us to begin with the monitoring systems based on advanced analytics and continuously monitor the CI/CD pipeline's security posture. One such implementation plan is to break out into an installment order, which can happen only after the series of pilot implementations have counted the effectiveness of new security technology before launching new security technology extensively in all operation teams. It should also entail periodic training sessions for the people, all of whom are made through regular training sessions so that all are qualified on the most up-to-date security practices and tools. Retail and e-commerce leaders also have to ensure that their CI/CD security frameworks can be adapted to address changes in regulatory demands and the dynamic threat environments. Security investment is being used to attain business objectives while at the same time lowering risk and maintaining the agility needed to compete. Retail and e-commerce enterprises viewing technology adoption with the strategic vision of staff training and regulatory compliance are expected to be able to define a secure, efficient, and resilient CI/CD pipeline in anticipation of what is to come.

Organizations need a multifaceted approach to adopting CI/CD security by organizations in the future and the trends are introduced. Modern CI/CD (or build and deploy or resilience) infrastructure is bound together by emerging technologies, proactive future-proofing, advanced analytics, and a strategic road map. Cyber threats are growing more advanced, and the regulatory requirements are getting more stringent, thus complicating the task of modernizing security practices for organizations and, more specifically, businesses operating in the retail and e-commerce sectors. It will protect the leaders to continue leading by AI/ML innovations, run full of comprehensive monitoring, and set out forward-looking strategic planes. Companies can decrease the risk by incorporating risk mitigation into the CI/CD pipeline at the very beginning and guaranteeing such a pipeline is robust and adaptive. They can work quickly, even in the most fast-paced modern software development environments.

8. Conclusion

Since then, software development practices have changed tremendously, and we have continuously integrated and developed. This increasing development of the technology field has come with new security challenges, notably the security of customer data in such sectors as retail and e-commerce. Due to the increased complexity and sophistication of the threat landscape, a major strategy to stay secure with applications has become the DevSecOps approach of security integration into the CI/CD pipeline. By integrating security into the core of the development process, these vulnerabilities can initially be identified and fixed while the vulnerability is still in the early life cycle before reaching production. For instance, protecting sensitive customer data, from personal to financial to transactional, is necessary in retail and e-commerce industries. Security breaches affect customer trust, financial losses valued in thousands of dollars, and damage that is unbearable to the company's reputation. This allows organizations to send security practices like Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Software Composition Analysis (SCA) as part of the CI/CD pipeline to build real-time protection against cyber threats. These tools help to catch vulnerabilities in static code. By running applications on them with the other eye open in case, we do not catch them in that formal code analysis, and third, by keeping an eye on taking all the dependencies we use and ensuring that none of them can introduce insecure components into our code base.

Security is integrated with the CI/CD pipeline, which reduces the vulnerabilities that flow to the production pipeline. The proactive part of this method is that security issues are caught during the current development process and corrected as early as possible. In other words, it saves us from running costly post-deployment patches, which may or may not cause data breaches, and makes the application more secure in the long run. This approach further aligns with the industry standards and compliance protocols such as the GDPR, PCI DSS, and CCPA for retail and e-commerce platforms that must be stringent and compliant. Continuous security and compliance in the CI/CD pipeline support regulatory penalties and protect customer's data at all times. While beneficial, there is no easy work to integrate security in CI/CD pipelines. One of the main problems is deployment speed vs. completeness of security checks. As the time to market a release reduces, the development team is under pressure to meet the deadline for the security of a release. In some cases, security testing will speed up the process, but at the cost of having a sense of agility and security. Integrating multiple security tools in a CI/CD pipeline can also be hard; actions must be taken to cement the integration of these tools without a lot of extra resources and knowledge to keep them working. Organizations must devise ways to automate and speed up their security test processes without choking a development workflow.

This is overcome by employing an automation approach, collaboration, and continuous improvement. To solve the problem of deployment speed, organizations can automate security checks and integrate them into every stage of the development pipeline. With SAST, DAST, and SCA automated testing tools, developers will receive fast feedback from vulnerabilities to identify and solve them quickly and without delay in the release. Parallel testing and cloud security solutions can also reduce the overall negative effects of the security scanning on that deployment timeline. The second important aspect is collaboration among development, operations, and security teams to deploy DevSecOps. Security should no longer be viewed as something kept separate from our role. Instead, it should be shared responsibility with all stakeholders. It also benefits in developing an environment of collaboration and common responsibility among the organization as it brings in security at various stages of the development cycle. However, teams can also receive regular training and share the knowledge to ensure everyone is updated with the new security practice and prepared to battle at least a few more threats that turn up.

While DevSecOps is technically a need in CI/CD pipelines, it has been an imperative business need for retail and e-commerce leaders. In today's marketplace, customer data protection and a guarantee of data integrity during the digital transactional environment are basic needs to maintain customer trust and lead in the competitive field of the digital marketplace. Embedding security into a retail and e-commerce organization's development process as much as possible will reduce the risk of a security breach, improve operations, and thus also improve customer experiences. However, looking into the future, it is clear that technologies like Artificial Intelligence and Machine Learning will be major parts of the security component introduced in the CI/CD pipeline. They may provide us with deeper insights into how our applications behave to assist in pointing out vulnerabilities more effectively or quickly. Such technologies also help validate valid CI/CD pipelines, enabling them to be more resilient to new security threats via their ability to predict and respond to them in real-time. These trends point to the retail and e-commerce leaders playing second fiddle in these areas, having done nothing to keep abreast of the changing technology and regulation and investing in security solutions at any cost, primarily through adaptive CI/CD pipelines.

It is imperative that modern applications secure and remain resilient and that security be integrated into CI/CD pipelines. Organizations can preemptively identify and remediate those vulnerabilities to ensure compliance and preserve customer data using a DevSecOps approach and some of those tools, such as SAST, DAST, and SCA. Like any process, integration security into the CI/CD process presents challenges, but the pros of securing CI/CD capabilities outweigh the cons. Retaining security in the development process is not optional for retail and e-commerce businesses, but it is a fundamental strategic approach for assuring a healthy, long-term business. This time, we will develop a strategy to enforce a proactive and intense security posture to help prevent the increasing threat landscape, keep the organization at the leading edge, and secure its digital assets.

References

[1] Abigail, L. (2020). Application Security Tools: Enhancing Software Defense in the Digital Era.

[2] Ahmed, A. M. A. A. (2019). DevSecOps: Enabling security by design in rapid software development (Master's thesis).

[3] Ayereby, M. P. M. (2018). Overcoming data breaches and human factors in minimizing threats to cyber-security ecosystems (Doctoral dissertation, Walden University).

[4] Bansal, A. (2020). System to redact personal identified entities (PII) in unstructured data. International Journal of Advanced Research in Engineering and Technology, 11(6), 133. https://doi.org/10.34218/IJARET.11.6.133

[5] Baumann, M., Krause, M., Overgaard, J., Debus, J., Bentzen, S. M., Daartz, J., ... & Bortfeld, T. (2016). Radiation oncology in the era of precision medicine. Nature Reviews Cancer, 16(4), 234-249.

[6] Boda, V. V. R. (2020). Balancing Speed and Safety: CI/CD in the World of Healthcare. Journal of Innovative Technologies, 3(1).

[7] Candel, J. M. O. (2020). DevOps and Containers Security: Security and Monitoring in Docker Containers. BPB Publications.

[8] Dashevskyi, S. (2017). Security assessment of open source third-parties applications.

[9] Debois, P. (2012). DevOps: A Software Revolution. IEEE Software, 29(3), 80-85.

[10] Deepak, R. D., & Swarnalatha, P. (2019). Continuous Integration-Continuous Security-Continuous Deployment Pipeline Automation for Application Software (CI-CS-CD). International Journal of Computer Science and Software Engineering, 8(10), 247-253.

[11] Fan, M., Petrosoniak, A., Pinkney, S., Hicks, C., White, K., Almeida, A. P. S. S., ... & Trbovich, P. (2016). Study protocol for a framework analysis using video review to identify latent safety threats: trauma resuscitation using in situ simulation team training (TRUST). BMJ open, 6(11), e013683.

[12] Felderer, M., Büchler, M., Johns, M., Brucker, A. D., Breu, R., & Pretschner, A. (2016). Security testing: A survey. In Advances in Computers (Vol. 101, pp. 1-51). Elsevier.

[13] Heilmann, J. (2020). Application Security Review Criteria for DevSecOps Processes.

[14] Jawed, M. (2019). Continuous security in DevOps environment: Integrating automated security checks at each stage of continuous deployment pipeline (Doctoral dissertation, Wien).

[15] Jawed, M. (2019). Continuous security in DevOps environment: Integrating automated security checks at each stage of continuous deployment pipeline (Doctoral dissertation, Wien).

[16] Jose, C. R. (2020). Exploring Security Process Improvements for Integrating Security Tools within a Software Application Development Methodology (Doctoral dissertation, Colorado Technical University).

[17] Kim, G., Behr, G., & Spafford, E. H. (2017). The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win. IT Revolution Press.

[18] Kumar, A. (2019). The convergence of predictive analytics in driving business intelligence and enhancing DevOps efficiency. International Journal of Computational Engineering and Management, 6(6), 118-142. Retrieved from https://ijcem.in/wp-content/uploads/THE-CONVERGENCE-OF-PREDICTIVE-ANALYTICS-IN-DRIVING-BUSINESS-INTELLIGENCE-AND-ENHANCING-DEVOPS-EFFICIENCY.pdf

[19] Mackey, T. (2018). Building open source security into agile application builds. Network Security, 2018(4), 5-8.

[20] Maduranga, H. (2020). State-of-the-Art Cryptographic Protocols and Their Efficacy in Mitigating E-Commerce Data Breaches on Public Clouds. Journal of Computational Intelligence for Hybrid Cloud and Edge Computing Networks, 4(10), 1-11.

[21] Marini-Wear, N. (2019). Qualitative Case Study Software Security in DevOps (Doctoral dissertation, Capitol Technology University).

[22] Mell, P., & Scarfone, K. (2005). A Technical Guide to Information Security Testing and Assessment. NIST Special Publication 800-115.

[23] Morales, J. A., Scanlon, T. P., Volkmann, A., Yankel, J., & Yasar, H. (2020, August). Security impacts of sub-optimal DevSecOps implementations in a highly regulated environment. In Proceedings of the 15th International Conference on Availability, Reliability and Security (pp. 1-8).

[24] Morris, K. (2016). Infrastructure as code: managing servers in the cloud. " O'Reilly Media, Inc.".

[25] Nimmo, P. (2018). Chasing the white rabbit to find a white elephant: Exploring limited/non-use of education technology in Mpumalanga, South Africa.

[26] Nyati, S. (2018). Revolutionizing LTL carrier operations: A comprehensive analysis of an algorithm-driven pickup and delivery dispatching solution. International Journal of Science and Research (IJSR), 7(2), 1659-1666. Retrieved from https://www.ijsr.net/getabstract.php?paperid=SR24203183637

[27] Opderbeck, D. W. (2019). Artificial intelligence in pharmaceuticals, biologics, and medical devices: present and future regulatory models. Fordham L. Rev., 88, 553.

[28] Paananen, T. (2016). Analyzing Java EE application security with SonarQube. https://www.theseus.fi/bitstream/handle/10024/109174/Paananen_Timo.pdf?sequence=1

[29] Phan, K. (2018). Implementing resiliency of adaptive multi-factor authentication systems.

[30] Rodríguez, G. E., Torres, J. G., Flores, P., & Benavides, D. E. (2020). Cross-site scripting (XSS) attacks and mitigation: A survey. Computer Networks, 166, 106960.

[31] Seaman, J. (2020). PCI DSS: An integrated data security standard guide. Apress.

[32] Sethi, N., & Sharma, A. (2018). A Survey on Static Code Analysis Tools. International Journal of Computer Applications, 179(47), 1-6.

[33] Shajadi, A. (2019). Automating security tests for web applications in continuous integration and deployment environment.

[34] Smith, S. (2016). B2B tech startup marketing and cloud computing: utilization and impact of cloud marketing automation software (Doctoral dissertation, University of Oregon).

[35] Stringer, J. (2020). Declaration patterns in dependency management: a thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Computer Science at Massey University, Manawatū, New Zealand (Doctoral dissertation, Massey University).

[36] Tirosh, A., Horvath, M., & Zumerle, D. (2019). Magic Quadrant for Application Security Testing.

[37] Van Rensburg, A. (2017). Vulnerability testing in the web application development cycle. University of Johannesburg (South Africa).

[38] Watada, J., Roy, A., Kadikar, R., Pham, H., & Xu, B. (2019). Emerging trends, techniques and open issues of containerization: A review. IEEE Access, 7, 152443-152472.

[39] Williams, L. (2019). Secure software lifecycle knowledge area issue. The National Cyber Security Center