International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

DevSecOps: Securing Infrastructure in the Age of Automation

Prashant K. Prasad

Sr. Regional Sales Director and Head - Product Business, USA

ABSTRACT: Cloud Native Infrastructure, Infrastructure as Code (IaC), containers, and CI/CD pipelines were used by organizations for better scalability and deployment times. This article explores how DevSecOps can enable security in automated infrastructure environments, and the ways that it integrates security in a continuous manner. DevSecOps adoption was evaluated and its effects on the performance of security using a quantitative research approach. The results indicated that 82% are conducting IaC security scanning, 76% have security part of their CI/CD pipelines and 71% are doing container vulnerability scanning. The research also revealed that vulnerability detection was up to 73% faster, configuration error incident avoidance was up to 67% and deployment reliability increased by up to 72% post adoption of the DevSecOps initiative. The results have revealed that DevSecOps as a key set of principles is to ensure that infrastructure configuration is secure and scalable.

KEYWORDS: DevSecOps, CI/CD Pipelines, Infrastructure as Code (IaC), Cloud Security, Kubernetes, Container Security.

I. INTRODUCTION

Cloud computing, automation and microservices are emerging as new ways to build and manage infrastructure very fast and it is causing a change in the way organizations build and manage infrastructure. Performing deployment was enhanced by technologies like Infrastructure as Code (IaC), Kubernetes, Docker and CI/CD pipelines. But these advances brought with them new security challenges such as misconfigured deployments and increased attack surface. Traditional security methods don't work in highly automated environments. DevSecOps became an important development of DevOps, bringing security into the development and deployment process. This paper will review the impacts DevSecOps offers in today's auto-pilot driven cloud computing environment, from a security, operational performance, and governance perspective.

II. RELATED WORKS

Infrastructure as Code and Security

Infrastructure as Code (IaC) was one of the biggest pillars of DevOps that enabled organizations to automate infrastructure provisioning, infrastructure configuration and deployment. Now infrastructure could be managed with executable scripts and infrastructure templates, rather than manual configuration. This new uniformity, scalability and deployment reduced time. However, studies revealed that as IaC increased in usage, security and operational risks also became an issue with scripts that included bugs, or misconfigurations.

A number of studies emphasized serious issues like outages, security break-ins and inconsistencies with configuration can be caused by IaC scripts. Studies discovered correlations between many script attributes and defects of the IaC scripts, such as code complexity, the owner of the script or the rate of changes in the scripts [1]. They found fourteen characteristics related to code and several process related characteristics that contributed to the infrastructure defects. These results illustrated that when infrastructure automation is automated, it doesn't necessarily mean reliability. On the other hand, configuration management systems are able to replicate configurations at scale and can quickly inflict vulnerabilities in various environments if the scripts are of poor quality. Another key topic reported in the literature is that of the prevalence of syntax and configuration issues in IaC environments.

IaC defects were identified in studies of several repositories like Mozilla, OpenStack, Wikimedia, and Mirantis, which revealed that a significant number of defects were due to configuration errors and syntax issues [4][6]. According to the empirical studies, there are high correlation of both hard coded strings and large size of scripts with IaC artifacts deficiencies [7]. The results have been reaffirmed the need of coding securely and automated enforcing policies in infrastructure automation.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

Therefore, to fix those problems, researchers suggested automatic methods to improve the quality of IaC. To locate patterns in errors that have been recurring in the past in infrastructure code, Han et al adopted machine learning based techniques [2]. They analyzed Puppet artefacts and extracted 41 error patterns, which led to 30 rules for pre-deployment checking of the artefacts for infrastructure issues. In essence, these methodologies were fully aligned to DevSecOps principles as they brought the processes to focus on quality and security earlier in the development process. Another significant improvement was the ability to automatically generate infrastructure code with the help of Model-Driven Engineering (MDE). The DICER approach was proposed which converted platform-independent models into scripts for IaC [3]. The method enabled them to significantly minimise manual coding work and promote uniformity of deployment across intricate data-abundant settings. The outcomes of this research referenced standardized modeling of infrastructure and leveraged the automation to reduce human error while managing infrastructure in a scalable way. But, even in an advanced level of automation, security validation could not be bypassed as automated infrastructure could not be avoided and had to be validated for compliance and vulnerabilities.

DevSecOps and Continuous Security

Today Continuous integration and continuous deployment (CI/CD) pipelines have revolutionized software delivery processes. Automated pipelines became more important for organizations to send out updates in a quicker timeframe and at a more regular interval. These practices made things easier on agility, on the other hand, there were problems in terms of security, reliability and governance issues. The current security approach was not going to be able to catch up with highly automated deployments, researchers said.

Researchers performed a systematic literature review on a range of different organisations and their practices towards continuous integration, continuous code delivery and deployment [5]. Some of the key practices they found as vital to secure, reliable automation were automated testing, security validation, build monitoring and improving build deployment reliability. The research focused explicitly that security should not be considered a stage of its development, which will take place separately, after the development process.

DevSecOps was the evolution from DevOps to proactive security over the reactive security. The focus on manufacturability and security in early development stages as encouraged by Secure DevOps principles shifted the focus of the organizations towards securing vulnerabilities before deployment [10]. The researchers claimed that the conventional software security strategies have led to tensions between the software development team and security team as security review was perceived as a delay or blocker. But Secure DevOps looked to address this issue by trying to integrate security into workflows and promote collaboration between teams. Research shows that DevSecOps is more about reducing attack surface and infusing security knowledge across the software development lifecycle [10]. The study has outlined the need of a secure soft system design in which the system must be operable even if an attack does take place.

Security in the deployment pipelines was another key point that was raised in the literature. With the advent of automated delivery pipelines, organizations had to worry about the security of their delivery pipelines, as its attackers started to attack the CI/CD system as well. Researchers found automated compliance validation, vulnerability scanning and deployment proficiency testing as essential practices for better secure deployments operations [5]. These practices progressed to eventually become the norm in DevSecOps pipelines.

The study also revealed that infrastructure in today's world was more complex which increased the need for a contribution to continuous validation. The introduction of multi-cloud, microservices and container systems created a lot of configuration complexities and vulnerabilities. Policy enforcement and real-time monitoring, therefore, were critical aspects of secure operations, and were increasingly done automatically. These problems were solved in DevSecOps through the security testing practices to be conducted in every step of the deployment process, such as testing of the source code, infrastructure, container images and monitoring at runtime.

Architecture and Containers

Adoption of cloud-native architectures/systems, microservices and containers fundamentally changed the way enterprise infrastructures operate. These technologies enhanced the flexibility and the scalability, but at the same time they increased the attack surface and pose new challenges in operation. The research studies have highlighted that software architecture is a key factor in the success of continuous deployment and DevSecOps adoption.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

Several architectural principles that helped DevOps deployment to be secure and reliable, such as DevSecOps were introduced by the researchers [9]. Some of these were the ability to make small, independent deployment units, make isolated changes, and have a centralized logging system and a robust test system. These themes resonated with microservices which are often employed in DevSecOps architectures due to their ability to enable organisations to deploy smaller and more autonomous services.

The adoption of containerization has taken cloud-native transformation to the next level, as well, with Docker and Kubernetes making their mark. While containers had advantages of deployment portability/scalability, they also introduced new security issues. Bad container images, insecure registries and shoddy runtimes turned into a big problem in the enterprise world. Consequently, DevSecOps was now growing its attention on container security measures like scanning images, monitoring at run-time and managing access control.

A growing number of IaC and cloud orchestration standards have helped in the knowledge of secure infrastructure automation. Researchers commented on the relevance of infrastructure blueprints and standards for cloud orchestration like TOSCA for defining the cloud deployment architecture [8]. These standards did help organizations to manage complex cloud applications in a standardized way with their deployment specifications. Improved consistency via ways of standardization and fewer configuration mistakes, both of which are crucial for securely managing infrastructure.

Another common theme in the literature was the fact that people were increasingly starting to adopt centralized logging and observability tools. Distributed systems grew in size and complexity and the need for improved visibility into their infrastructure behavior and security events arose. These platforms, systems, and tools for anomaly detection quickly became a necessity to rapidly identify threats and respond to incidents. The technologies provided the DevSecOps goals by way of enabling continuous enjoyment and operational intelligence throughout cloud environments.

III. METHODOLOGY

The research adopted in this study is a quantitative research method for analyzing the impact of DevSecOps practices on infrastructure security in highly automated environment cloud and DevOps. The study will use a structured online survey questionnaire that will be sent to practitioners involving DevOps, cloud engineering, cybersecurity, infrastructure and software development. The respondents will be picked from the organizations that are actively practicing DevOps and/or DevSecOps. The study will survey some 150 to 200 IT enterprises, cloud service providers and IT enterprise technology teams. Purposive sampling method will be adopted due to the nature of the research where participants who have experience in working with automated infrastructure and security function are needed.

The study will be based on a questionnaire which will have a demographic and technical component. The demographics section will ask questions that include job role, years of experience, organization size and platform usage of the cloud. The technical section is going to be based on the 5-point Likert Scale ranging from “Strongly Disagree” to “Strongly Agree” and will gauge how well DevSecOps practices are implemented. It will focus on the most important DevSecOps areas covered by the survey, such as automated vulnerability scans, security validation of Infrastructure as Code, policy-as-code enforcement, container image scans, identity and access management, centralized monitoring, and continuous compliance checks.

The output parameters will encompass the decrease in security events, efficiency of deployment, compliance performance, speed of detection and reliability of infrastructure. The study will also analyze the company's vulnerability and response time if organizations that have fewer vulnerabilities or faster response time use security automation methods more than those organizations that do not use any security automation methods.

Descriptive statistics will be used to analyze data collected. The responses will be summarized of descriptive statistics, percentage, mean and standard difference. Pearson correlation will be used to analyze the relationship between implementing DevSecOps operation and the security performance of infrastructures. Using regression analysis can also help to understand the relationship between DevSecOps practices and security outcomes, as well as the strength of this relationship. A statistical method will be used in analyzing data, which includes SPSS or Microsoft Excel.

The questionnaire will be designed to have the most accurate and reliable answers possible by re-engineering it with other existing research studies on DevSecOps and Infrastructure as Code. The survey with IT professionals will be done first with a selected group of them in order to pilot it before the rest of the group will complete the survey. Clarification and consistency of questions will be done based on the feedback received from the pilot test. The survey

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

will still be conducted voluntarily and information from the respondents will be confidential to ensure ethical participation in research.

IV. RESULTS

DevSecOps in Infrastructure Environments

The results of the study indicated that for companies with automated deployment environments and cloud native infrastructure the adoption of DevSecOps had gone up significantly. The majority of respondents stated their organization had already integrated security into the CI/CD as a separate process rather than having existing security reviews as a post-deployment process. The results mirrored an increasing trend by the industry towards continuous security validation in highly automated systems.

82% of respondents indicated that their organization adopted Infrastructure as Code (IaC) for provisioning and managing infrastructure to a very large extent. The findings showed there's been a growing trend of organizations seeing security automation as part and parcel of digital transformation, instead of it being an optional activity.

The results also revealed that companies that run in a multi-cloud environment were more DevSecOps mature than those that use a traditional on-premise environment. Those who intended to operate security manually in numerous cloud platforms described operational challenges that needed to be overcome, and noted that automated governance and policy enforcement is necessary.

Table 1: DevSecOps Security Practices

DevSecOps Practice	Organizations Using Practice (%)
Infrastructure as Code (IaC) Security Scanning	82%
CI/CD Pipeline Security Integration	76%
Container Vulnerability Scanning	71%
Policy as Code Enforcement	68%
Centralized Logging and Monitoring	74%
Automated Compliance Validation	63%
Runtime Container Security	59%

Results also showed that using multiple DevSecOps practices together lead to higher consistency in the organization's operations and reduced interruption of operations. Companies with automated Security checks that were done early in the development lifecycle reported smoother deployment processes, and fewer delays in the security checks were performed at the end of the development lifecycle.

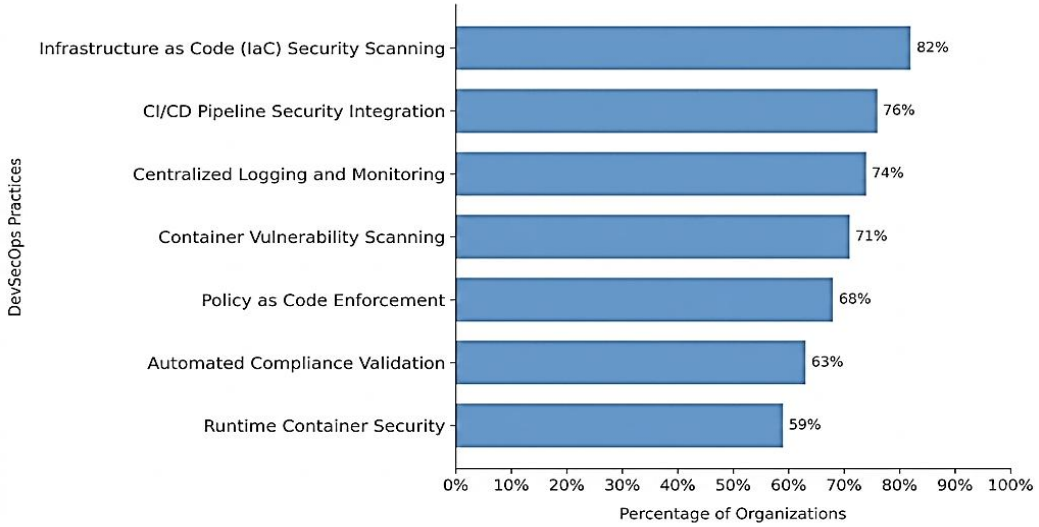

Fig. 1: Adoption Rate of Major DevSecOps Practices

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

It was also revealed that against the backdrop of growing use of Kubernetes, container security emerged as one of the top priorities. Risks in deployment reduced due to image scanning and secure management of the registry systems were the major advantages and were highly emphasized by the respondents working with containerized environments. There were more security alerts and disruption of operations reported for organizations with no container security controls.
One other interesting thing that was witnessed was an increase in the implementation of the policy-as-code. A number of organizations said they had automated governance policies to help them meet compliance requirements in the cloud. This was a significant time-saving and security management streamlining in the absence of manual auditing. The results indicated that DevSecOps practices were turning into part of basic infrastructure operations.

Infrastructure Security and Risk Reduction
The quantitative analysis showed a good correlation between DevSecOs adoption to the better outcome for the security of the infrastructure. But companies that had developed a mature DevSecOps were able to report fewer incidents of security breaches, decreased number of misconfigurations, and quicker remediation of security vulnerabilities as compared to those using traditional security.

A large development was one of the findings that was doing Infrastructure as Code validation. The respondents reported that the pre-deployment scenarios benefited from the help of automated scanning tools of IaC. This substantially lowered risk of insecure templates for infrastructure in the cloud, and the misconfiguration of cloud resources. When manual infrastructure reviews were no longer enough, many participants explained that automated deployments happened far too often for them to be able to follow up on the process as is typical with manual infrastructure reviews.

Table 2: Security Improvements

Security Outcome	Improvement Reported (%)
Reduction in Configuration Errors	67%
Faster Vulnerability Detection	73%
Improved Compliance Management	65%
Reduced Security Incidents	61%
Faster Incident Response	69%
Improved Deployment Reliability	72%

The research also revealed that companies that have adopted automated vulnerability scanning, part of their CI/CD pipeline, saw upped identification of software vulnerabilities. Security teams would know if vulnerable libraries, insecure dependencies and coding issues exist within applications prior to them entering into production environments. This simplified the remediation activities, thereby lowering the remediation costs and complexity.

A policy-fueled automation resulted in a better synchronization of the governance throughout cloud platforms, the results showed. Compliance software for automated self-verification explained by those working across hybrid and multi-cloud that the software made everything simpler to manage by regulatory requirements, and lowered the chance of policy violation. In sectors like financial services, sectors where adherence was likely to be stricter, like healthcare, and telecoms, it was particularly critical.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

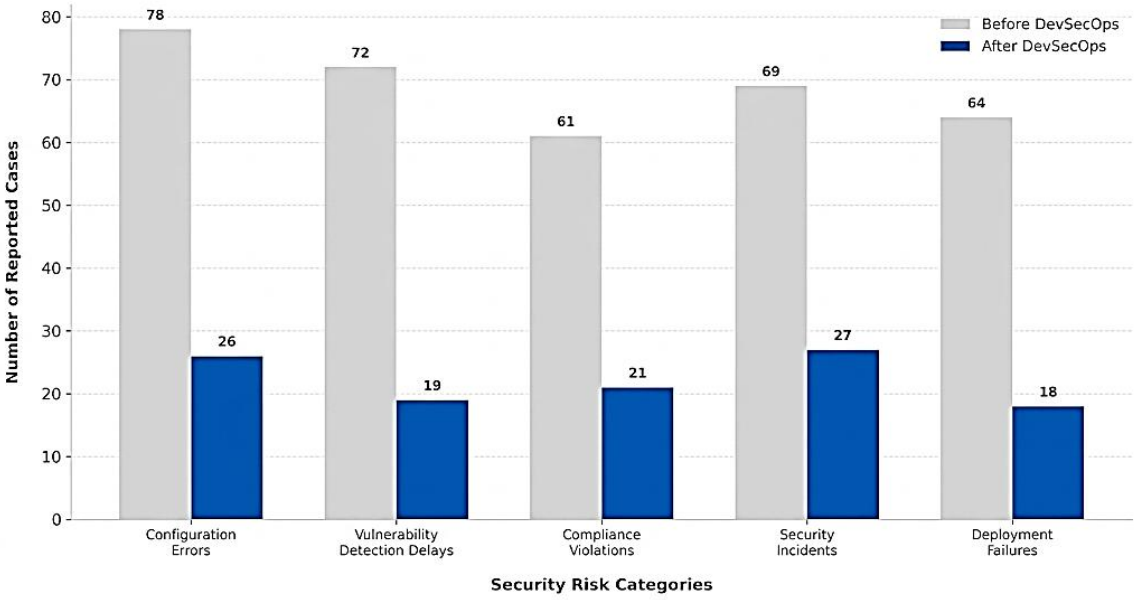

Fig. 2: Comparison of Security Incidents

The other key operational learning was related to operational resilience. The organizations that relied on centralized monitoring and observability claimed they were able to detect abnormal behavior and security issues at quicker speeds. There were new solutions like Logging (ELK Stack and Splunk) that provided better visibility into distributed environments with a faster investigation of incidents. Many of the respondents also said that they started using an AIOps-based monitoring system to detect anomalies automatically.

Automated organisations had greater confidence in effectiveness of deployment reliability and maintenance downtime due to failure to operate as per the deployment failing due to configuration. This discovery reinforces the notion that including security as part of automation workflows boosts the performance of operations and infrastructure.

Organizational and Cultural Impact

The study revealed that the technology solution wasn't the only thing that was required to implement a successful DevSecOps solution, but also the change in organizational and cultural environments. Several of the respondents detailed that it was a lack of communication and delays between the different teams associated with development, operations and security. DevSecOps contributed to lowering these hurdles by fostering collaborative processes and shared responsibility.

Embedding security team members directly into DevOps teams was beneficial to communication and issue resolution, said a significant proportion. Companies that set up joint KPIs in teams found that the task of coordinating between teams was smoother and their deployment was faster while functioning in line with protection policies.

Table 3: Organizational Benefits

Organizational Benefit	Respondents Agreeing (%)
Better Collaboration Between Teams	78%
Faster Release Cycles	74%
Improved Security Awareness	81%
Reduced Manual Security Work	69%
Improved Governance Consistency	66%
Increased Operational Transparency	72%

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

The results indicated that the use of DevSecOps helped to build an organization's security culture. Teams saw security as more of an objective rather than a task of a security team and involved others where possible. Secure coding practices were more deeply engaged by the Developers and practices in securing coding had more knowledge of monitorable and compliance manageable by the Operations teams.

Training and skill building was also a key area of focus in the respondents, when speaking about training and skill transfer in DevSecOps transformation efforts. One common hurdle was a lack of familiarity with automated security tools and cloud-based technologies among employees in many organisations. Those companies that prioritized on-going learning programs, however, had smoother transitions and better engagement results.

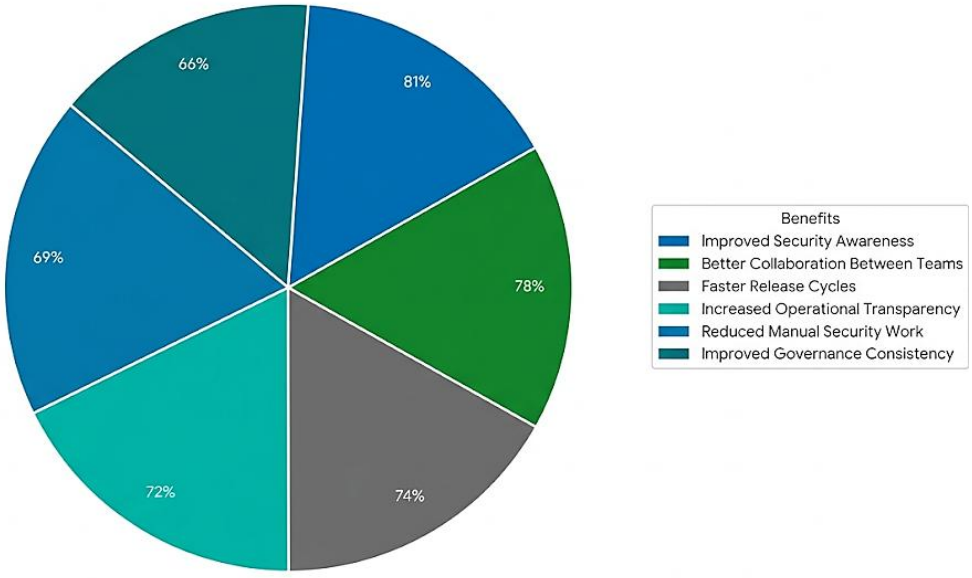

Fig. 3: Organizational Benefits Through DevSecOps

It was observed that in successful implementation management support was an important component. Companies that had solid support by their executives were more inclined to have a central governance model in place, commitment to investing in automation tools and a clear DevSecOps policy. Despite the potential for resistance to change, leaders that expressed a commitment to change reduced resistance and encouraged a cultural change in the long run.

The research also showed that collaborative DevSecOps-based organizations also reported fewer bottle necks in deployment. Conventional release methodologies are sometimes behind schedule since security was evaluated towards the end of the development process, resulting in delays. With continuous security validation, on the other hand, responses were speeded up and the deployments were deployed more effectively, while maintaining the compliance.

Multi-Cloud Security and Automation Maturity

The final part of analysis was on how DevSecOps assisted in the management of security in a multi-cloud and highly distributed environment. The study revealed that multi-cloud deployments were more complex for organizations, with boundaries that were inconsistent, identity management challenges, and lacking a consistent governance framework.

The vast majority agreed DevSecOps made it easier to have a consistent approach to security in the cloud. There was an increase in visibility and a decrease of inconsistencies in policy enforcement and monitoring across platforms with the help of automation. Policy-as-code (PaC) organizations indicated better governance and compliance management capabilities whether deployed on public or private clouds.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

Table 4: Challenges Faced Before DevSecOps Adoption

Security Challenge	Organizations Reporting Issue (%)
Inconsistent Cloud Security Policies	72%
Manual Compliance Processes	68%
Delayed Vulnerability Detection	66%
Poor Visibility Across Environments	71%
Configuration Drift	63%
Slow Incident Response	59%

The study results showed that automation of infrastructure does not necessarily create a secure operation. There were higher operational risk for those that dedicated only to deployment speed and didn't take security controls into consideration. The vulnerabilities which were identified also tended to be exacerbated if these could quickly be spread across environments as a result of the rapid automation, as explained by the respondents.

The data indicated that companies with the more mature DevSecOps process were more likely to have robust infrastructure as they moved forward with implementing DevSecOps initiatives across their business. These organisations have been able to cope with the increased complexity of their infrastructure with automated governance, ongoing monitoring and integrated compliance validation. A majority of respondents envisioned DevSecOps continuing to evolve with the arrival of new technologies, with Artificial Intelligence (AI) for operations, serverless applications, and an edge infrastructure drawling the biggest interest from them.

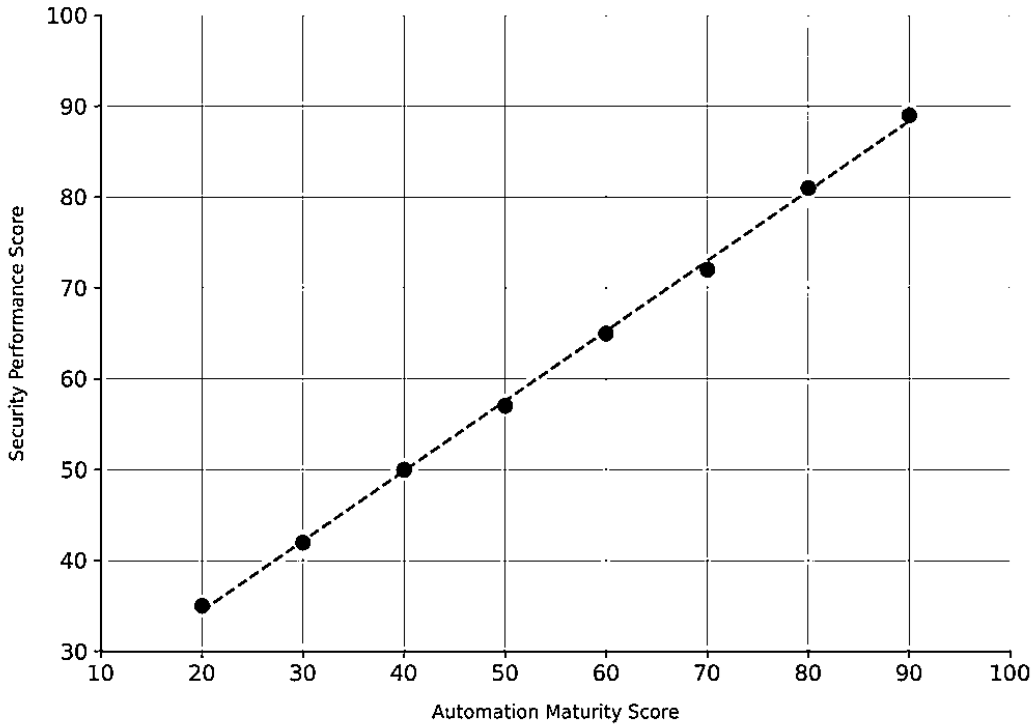

Fig. 4: Automation Maturity and Security Performance

One other significant insight was the value added to decision making for distributed applications and systems with a centralized level of observability and intelligence. The availability of the real-time monitoring tools helped in organization to detect unusual behavior at a much faster rate and to minimize response timing during incidents. This would boost the operational resilience and enhance the reliability of the services that could be provided by the organizations while working in huge automated systems.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 2, Issue 1, January- February 2019||

DOI:10.15662/IJRPETM.2019.0201003

V. CONCLUSION

The study concludes that DevSecOps is an important part of enabling today's infrastructure to be automated and secure. The results revealed that companies using DevSecOps practices had enhanced vulnerability discovery, adherence with regulations better managed, fewer configuration problems and more reliable deployment practices. Security integration into CI/CD pipelines, validations of IaC, container scanning and policy-as-code enforcement enable organizations to lower risks of their cloud-native systems. It also emphasized the need for a strong partnership between development, operations and security teams to effectively deliver on implementation. In today's fast-paced infrastructure landscape, DevSecOps is a crucial approach that will help enterprises become fast and secure in order to survive digital transformation and to be able to effectively withstand any risks with cloud operations.

REFERENCES

[1] A. Rahman, "Characteristics of defective infrastructure as code scripts in DevOps," *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, pp. 476–479, May 2018, doi: 10.1145/3183440.3183452. Available: https://doi.org/10.1145/3183440.3183452

[2] W. Chen, G. Wu, and J. Wei, "An Approach to Identifying Error Patterns for Infrastructure as Code," *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 124–129, Oct. 2018, doi: 10.1109/issrew.2018.00-19. Available: https://doi.org/10.1109/issrew.2018.00-19

[3] M. Artac, T. Borovsak, E. Di Nitto, M. Guerriero, D. Perez-Palacin, and D. A. Tamburri, "Infrastructure-as-Code for Data-Intensive Architectures: A Model-Driven Development Approach," *2018 IEEE International Conference on Software Architecture (ICSA)*, Apr. 2018, doi: 10.1109/icsa.2018.00025. Available: https://doi.org/10.1109/icsa.2018.00025

[4] A. Rahman, "Anti-Patterns in Infrastructure as Code," *Proceedings - 2018 IEEE 11th International Conference on Software Testing, Verification and Validation, ICST 2018: 434-435*, pp. 434–435, Apr. 2018, doi: 10.1109/icst.2018.00057. Available: https://doi.org/10.1109/icst.2018.00057

[5] M. Shahin, M. A. Babar, and L. Zhu, "Continuous Integration, Delivery and Deployment: A systematic review on approaches, tools, challenges and practices," *arXiv (Cornell University)*, Mar. 2017, doi: 10.48550/arxiv.1703.07019. Available: https://doi.org/10.48550/arxiv.1703.07019

[6] A. Rahman, S. Elder, F. H. Shezan, V. Frost, J. Stallings, and L. Williams, "Bugs in infrastructure as code," *arXiv (Cornell University)*, Sep. 2018, doi: 10.48550/arxiv.1809.07937. Available: http://arxiv.org/abs/1809.07937

[7] A. Rahman and L. Williams, "Source code properties of defective infrastructure as code scripts," *arXiv (Cornell University)*, Oct. 2018, doi: 10.48550/arxiv.1810.09605. Available: https://doi.org/10.48550/arxiv.1810.09605

[8] M. Artac, T. Borovssak, E. Di Nitto, M. Guerriero, and D. A. Tamburri, "DevOps: Introducing Infrastructure-as-Code," *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pp. 497–498, May 2017, doi: 10.1109/icse-c.2017.162. Available: https://doi.org/10.1109/icse-c.2017.162

[9] M. Shahin, M. A. Babar, and L. Zhu, "The Intersection of Continuous Deployment and Architecting Process," *ESEM '16: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 1–10, Sep. 2016, doi: 10.1145/2961111.2962587. Available: https://doi.org/10.1145/2961111.2962587

[10] H. Yasar and K. Kontostathis, "Where to integrate security practices on DevOps Platform," *International Journal of Secure Software Engineering*, vol. 7, no. 4, pp. 39–50, Oct. 2016, doi: 10.4018/ijsse.2016100103. Available: https://doi.org/10.4018/ijsse.2016100103