Abstract— DevOps ensures collaboration of all stakeholders in the software development process based on lean and agile principles by incorporating user’s feedback in a faster manner. Adaptation of DevOps guarantees customer satisfaction and increased business value with reduced time for bagging the feedback and adjusting the deliverables. As the business is moving to cyber space, security concerns need to be given priority and DevOps requires security to be embedded in it. This seems to be the mission critical issue in many organizations as it requires breaking down of the barriers of operations and security team and review of many security policies in place. This introduced a challenge of finding the best way to implant security in DevOps environment and how it can be automated so that DevOps can still perform Continuous Integration (CI) and Continuous Delivery (CD). This paper introduces a complete migration framework from DevOps to DevSecOps explaining the strategy, steps, tools and support functions.

Keywords— DevSecOps migration, DevOps framework, security in DevOps, migration levels, support functions for migration, migration strategy, tools

I. INTRODUCTION

Software development process is the decisive force for determining the quality and speed of software development. The traditional waterfall model follows a series of phases to be executed as an interleaved chain of actions, each phase providing a documentation as input to the next phase and testing and deployment of the product done at the last phase. This traditional model give way to continuous delivery based DevOps where each individual deliverables are tested and deployed. Later the developers sensed the need of infusing security in to this continuous development and deployment process and started thinking of Security in DevOps or DevSecOps.

Now a days companies are moving from software as a product to software as a service as the software is centrally hosted mostly on cloud and accessed through a browser. This facilitated the software provider to deliver improved versions without delivering the whole software again. This transformation in software delivery give rise to the concept of Continuous Integration (CI) and Continuous Delivery(CD) which reduced the cycle time and incorporated faster feedback. This was the starting point of the evolution of the concept called DevOps (Development and Operations). Cloud adaption and software defined environments also contributed to the requirement of daily deliverables of software solutions and supported the introduction of DevOps concept.

This scenario required better collaboration and communication of development and operations team in a software development environment. DevOps implemented a lean development methodology by integrating the different processes such as development, delivery and operations. This whole concept ended up in shifting the software development process from a distributed autonomous groups to cross-functional groups delivering continuous results.[1] DevOps combined software development with other information technology operations. Introduction of DevOps shortened the systems development life cycle by delivering desired results and updating frequently in agile manner aligning closely with business objectives.

Security in DevOps has been a challenge since customary security techniques are unable to maintain agility and speed. Introduction of DevOps facilitated rapid software deployment and service by automating the software development and deployment lifecycle. DevOps primarily biased towards availability of CIA triads; so need to think of integration of confidentiality and integrity also to the DevOps framework.[2]

This style of fast deployment of software components will make security to suffer if security testing is not integrated with. Organizations considers security as a barrier for fast development and deployment of software solutions. They does not want security to slow down the development process and always tries to shift it to last phase[3]. Industries so identified the need of information security professionals becoming an active participant in DevOps, preserving the concept of teamwork, coordination, agility and shared responsibility[4]. For that organisations must integrate modernized security methods to achieve optimised software development process. Practitioners also agree to the fact that DevSecOps is not just adding security to the CI and CD; but building security to the software to deliver compliances[5]. This integration becomes crucial when the security needs to be aligned with the compliances of different regulatory standards.

To facilitate this; developers, information security professionals and operation team must work in harmony maintaining agility and speed of DevOps. This requires the information security tools to be integrated at multiple points into DevOps lifecycle[6]. Traditional application security testing offerings were not designed for speed and transparency. The majority of DevOps developers keep low priority for security of the applications. Organizations also concern about compliance issues in DevOps culture. These reasons created an intense need of adopting security into the
DevOps making DevOps and security a single entity of existence.

In this paper we are addressing DevSecOps as a way of integrating security in DevOps where the absorption of security is spread across the overall process of software development. DevSecOps advanced workflow will integrate security practices at every stage of entire lifecycle. This paper talks about a complete migration framework from DevOps to DevSecOps including the strategy, procedure, support functions and tools.

The remainder of this paper is as follows. Section 2 discusses the previous researches relevant to security in DevOps and its adaptation to software development. Section 3 explains the migration framework and Section 4 talks about the contributions of the research in the software development community. Finally the paper is concluded with the indication of future research direction.

Fig. 1. Security Adaption Approaches

II. LITERATURE REVIEW

Earlier software development and delivery process followed a way of promising a delivery of the entire product in a time line specified by the users. This many times end up in false promises and extended delivery deadlines compromising the quality, time and efficiency. DevOps improved this process by introducing joined, agile, lean and transparent process in development and deployment [7]. In DevOps, organizations decides continuous delivery of up gradations after incorporating customer feedback[1].

Industries follow three approaches for adopting security in DevOps environment. (Fig 1)

1. DevOpsSec
2. SecDevOps
3. DevSecOps

DevOpsSec: It implies that security is integrated at the end of the process. If security is added at the last the software is developed in a fundamentally insecure way and then the development team starts thinking about the security[8]. But security is not embedded into every aspect of the lifecycle. The team will complete development, deployment and operation, and then review security. This is a traditional way but if it is completed in small increments and accomplished rapidly then it is a good improvement to the security assessment. However, it lacks complete security, optimum speed and may give a way to vulnerabilities. It requires the security team to be the primary testers of the software and make sure that this does not slow down the process[3].

SecDevOps: It implies security integration before any development or operations. This approach is difficult to adopt practically, it focuses on incorporating security requirements in the beginning of DevOps process. Development requirements should include security requirements. These security requirements cannot be changed after starting the actual development. Automated tests are created based on security requirements and added to process to ensure that security throughout the cycle. But it lacks updated security checks and adaption.

DevSecOps: In this approach, security is incorporated into the overall process. Equip existing development life cycle with highly automated tools enabling developers to take small actionable step quickly and enhance security. This method also emphasizes on improving collective teamwork of security, development and operation professionals. Actionable feedback mechanism must be integrated throughout the life cycle to continuously update security of developed software. This approach is gaining wide acceptance and many organizations are shifting from DevOps to DevSecOps.
Security at the end of development lifecycle was slowing the process and was causing flaws and delay in delivery. So, there was always a need to automate security-testing processes. It was necessary to make sure that the security team in the development process work in tune with the DevOps environment and not in isolation. DevSecOps believes in delivering security as a code. DevSecOps Centre of Excellence (EGT LAB) defines the best practices of DevSecOps as establishing a security culture, coaching, security first design, automation, build-deploy-test often and DevSecOps friendly acquisition[9].

Increased predictability, consistency and auditability contributed to security friendliness in DevOps; at the same time faster production, more deployments, less test cases and no audit aggravated the security issues in DevOps[7]. DevSecOps delivers many advantages to the software development process by embedding security at the very basic level of development. DevSecOps promotes threat hunting, immutable infrastructure, security auditing and monitoring, cost reduction by ensuring customer value, culture of openness and transparency and measuring ability [10].

DevSecOps make sure the integration of security in the development by shifting security to the left. This provides a great advantage to the development process in terms of cost by avoiding lots of rebuild on the software[1]. The DevSecOps migration should also consider the integration of security at different layers such as infrastructure, platform and application [2]. Software practitioners suggest that DevSecOps should focus on software quality and security, compliance requirement and risk avoidance in the process of software development and deployment[11].

DevSecOps practitioners suggest some key points that eases the migration as adapting security with operations not vice versa, remember 100% security is not attainable, compensate it with security applications, identify and remove known vulnerabilities, train developers on security coding[12]. Sanitizing the input to block common attacks like buffer overflows, SQL injection and cross site scripting is one best practice that can be followed to include security in DevSecOps[4].

The best practices suggested for the migration include good documentation, strong collaboration, process automation and enforcement of separation of duties[13]. Stemming NIST framework along with the DevOps delivery pipeline is one method adopted for DevSecOps implementation in healthcare sector [14].

One challenge this collaboration of security faces is the mandatory information sharing happens between the teams which may have countless security implications. So there
should be a proper policy defined for this information sharing when the organisations are opting for DevSecOps[15]. The limitless collaboration as part of security integration may also result in unauthorized access and need to be addressed properly [16]. The major challenge over here is the slow maturation of the security tools which will enable the integration of security in DevOps environment [17].

Integration of security with other workgroups sometimes creates conflicts in the group dynamics. One methodology introduced for integrating security in DevSecOps without conflicts is agency theory. It defines development team as the agent and customer as principal and the conflicts happen in the goal attainment. It suggests goal abstraction and information transparency as the remedies for this problem[18].

A DevSecOps maturity model is defined with the stages Burp, Crawl, Walk, Run and Fly which can be used to define the DevSecOps posture of any organization[5]. The different aspects of DevSecOps like the definition, security best practices, compliances, process automation, tools, software configuration, team collaboration, availability of auditing data and information security is analysed to find the momentum of this concept in the software development methodology [15].

As the concept of DevSecOps is still in its preliminary stage, organizations are struggling to find a model to benchmark their actions towards integrating security in DevOps. There is a requirement of a comprehensive model which can be used as a checkpoint when the organizations migrate to DevSecOps approach. This research tries to integrate all the different aspects of security migration in to a single framework and provides a consolidated technique to fix the problem that organizations undergo when migrating to DevSecOps. This framework is unique in the aspects that it comprehensively covers all the different components of information system development and their contributions to the DevSecOps migration.

III. CONCEPTUAL FRAMEWORK- MIGRATION FROM DEVOPS TO DEVSECOPS

Migration from DevOps to DevSecOps requires complete strategy to profess a secure culture and hassle free smooth transitions. Unplanned and unthoughtful migration may not achieve the main purpose of DevSecOps. Before integration into DevOps, security teams should undergo a series of homework including risk assessment, threat modelling and defining security baselines. This helps the security team to flawlessly integrate security in to the automated CI and CD phases.

The migration framework explained bellow talks about the following components.

A. Levels of Migration

B. Migration Strategy

C. Migration Procedure

D. Support Functions

E. Tools

Fig. 2 shows the complete migration framework including all the above mentioned components and their interrelationships. It provides an instrument to identify the tools used in every stage of integration and the support functions at each level that needs to be aligned with the migration.

A. Levels of Migration

DevSecOps migration strategy is focused on four levels: Governance, People, Process and Technology [19][20].

Governance

Governance is the base level from which the migration should start from by ensuring the top management support for the security integration to DevOps. This may necessitate a complete remodelling of the governance and compliance framework with proper delegation of authority to the operational support.

People

People, considered to be the weakest link of security plays an important role in DevSecOps migration too. They are responsible for removing the silos by communicating effectively in the entire process of DevSecOps migration[10]. They need to innovate and adapt quickly to the new environment [2] to make sure that the inclusion of Security into DevOps does not ruin the speedy delivery concept of DevOps. Proper communication, feedback mechanism and employee training is effectively included in this migration framework as migration procedures and support functions.

Process

Aligning the software development process with security is the important concern in DevSecOps migration[2]. There should be a proper integration of security process to the development and deployment process flow adding incident response, clear documentation and version control[1][20]. It should include a policy to move security to the earliest, for standard enforcement, security testing, release and deployment[7]. Establishing a concrete DevSecOps practice is necessary to make sure that the new security process does not affect the agile behaviour of DevOps.

Technology

At this level the focus will be to identify the extent to which technology can enable the migration process. The migration framework identifies the repetitive tasks for establishing security and applies automation to the utmost level it can. Host hardening, CI/CD patching etc. are some of the examples of the technology adaption for DevSecOps migration[10][2]. Technology should aid the reduction of manual intervention in security testing so that the migration to DevSecOps will be smooth without any hurdles and delays.

B. Migration Strategy

The strategy for the migration to DevSecOps describes the approach organizations follows while migrating which ensures all the aspects of migration has been taken care of. We suggest set of migration strategy which extensively covers all the phases of migration starting from understanding the requirements to continuously monitoring and vulnerability management.

- Strategy 1: Understand DevSecOps and it’s extend with depth.
- Strategy 2: Identify requirements of DevSecOps.
- Strategy 3: Gather required tool set and infrastructure changes
• Strategy 4: Harden the core infrastructure and integrate secure development tools
• Strategy 5: Adopt secure development processes with enhanced security automation.
• Strategy 6: Integrate security controls and compliance automation tools and practices in operations
• Strategy 7: Continuously monitor and analyse output at each process and integrate feedback mechanism for entire lifecycle to continuously update security posture.
• Strategy 8: Set vulnerability management program, threat intelligence mechanism and red team for security testing at regular interval as well as in ad-hoc manner.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Contribution in Security Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aqua Security [21]</td>
<td>Accelerates application delivery by early detection of bugs. This helps in “shift left “security in to DevOps by reducing the attack surface.</td>
</tr>
<tr>
<td>CA Technologies [22]</td>
<td>Provides Identity and Access Management system and uses a trusted platform for integrating security across applications in to development process.</td>
</tr>
<tr>
<td>CodeAI [23]</td>
<td>Does deep code analysis to uncover bugs as early as possible and facilitate the analysis of code as they are building.</td>
</tr>
<tr>
<td>Synopsys [24]</td>
<td>Helps to integrate security into every aspect of development by injecting security and quality analysis in to the development workflow.</td>
</tr>
<tr>
<td>Checkmarx [25]</td>
<td>DevOps introduces a new risk known as software exposure to the product as we try to deploy it frequently. This tool identifies this risk at earlier stages by integrating secure delivery process.</td>
</tr>
<tr>
<td>Chef Automate[26]</td>
<td>Analytical tool which enables cross team collaboration aiding compliances and auditable history.</td>
</tr>
<tr>
<td>CloudPassage [27]</td>
<td>Tool which provides complete security visibility and continuous compliances for cloud infrastructure.</td>
</tr>
<tr>
<td>CollabNet VersionOne[28]</td>
<td>A platform developed for enabling the collaboration of different stakeholders in the agile development methodology.</td>
</tr>
<tr>
<td>Contrast [29]</td>
<td>A tool that detects and fixes vulnerabilities for known attacks and targeted attacks.</td>
</tr>
<tr>
<td>CyberArk [30]</td>
<td>Tool to provide access security solutions to protect workstations against credential theft in a public cloud environment.</td>
</tr>
<tr>
<td>Datical [31]</td>
<td>Tool for database deployment and change management automation, which ties with CI/CD of DevSecOps. It helps the developers to find the implementation issues before actual deployment and accelerates the development pipeline.</td>
</tr>
<tr>
<td>DBmaestro [32]</td>
<td>Tool for accelerating database deployment process for DevOps to achieve faster, error free implementation and security policy compliances and auditing.</td>
</tr>
<tr>
<td>IBM UrbanCode [33]</td>
<td>Tool aids in automating application deployment through DevOps process. It offers rapid feedback, continuous deployment, audit trials and versioning.</td>
</tr>
<tr>
<td>IBM AppScan [34]</td>
<td>Entire suit for application scanning and provides automated security risk ratings, which helps in security assessments, vulnerability prioritization etc.</td>
</tr>
<tr>
<td>IBM QRadar [35]</td>
<td>SIEM tool to help the security teams to reduce the incidents and prioritize the detected threats by analysing the events for insights.</td>
</tr>
<tr>
<td>Imperva [36]</td>
<td>Tool set for application security and data security provides Content Delivery Network (CDN) solution for quick deployment and faster feedback.</td>
</tr>
<tr>
<td>JFrog Xray [37]</td>
<td>Tool for continuous security and artefact analysis for detecting vulnerabilities, compliance issues and quality assurance to aid the CI/CD pipeline.</td>
</tr>
<tr>
<td>Nosprawl [38]</td>
<td>Tool for delivering continuous security deployments, which helps to identify the vulnerabilities in the earlier stages of development and promises SecDevOps.</td>
</tr>
<tr>
<td>Parasoft [39]</td>
<td>Tools for automated software testing solutions which provide integrated API testing for early and error free deployment of deliverables in DevOps.</td>
</tr>
<tr>
<td>Qualys [40]</td>
<td>A single cloud platform for all IT security and compliances activities which includes features for continuous assessment, identification of Zero day vulnerabilities, automatic patch management etc.</td>
</tr>
<tr>
<td>Redgate [41]</td>
<td>Devops Database deployment tool for automatic database deployment, performance monitoring, standardized team based development and data protection.</td>
</tr>
<tr>
<td>Rogue Wave [42]</td>
<td>Tool for reverse debugging and dynamic analysis to deploy secure enterprise applications.</td>
</tr>
<tr>
<td>Sonatype Nexus [43]</td>
<td>Tool to build and manage artifacts across software supply chain for better coordination and faster deployment.</td>
</tr>
<tr>
<td>Sumo Logic [44]</td>
<td>Continuous intelligent platform for DevSecOps to collaborate, develop, operate and secure the applications on cloud environment.</td>
</tr>
<tr>
<td>WhiteHat Security [45]</td>
<td>Application security platform for DevSecOps implementation to enable business to protect data, ensure compliances and manage risks.</td>
</tr>
<tr>
<td>Continuum Security [46]</td>
<td>Automatic threat modelling tool with integrated access control to enable collaboration between security and engineering team in DevSecOps implementation.</td>
</tr>
<tr>
<td>ThreatModeler [47]</td>
<td>An automated threat modelling solution that secures enterprise software development life cycle by identifying the threats on entire attack surface to make proactive security decisions and minimize overall risk.</td>
</tr>
<tr>
<td>Dome9 [49]</td>
<td>Cloud infrastructure security for SaaS to minimize the attack surface and protect against vulnerabilities, identify theft and data loss.</td>
</tr>
<tr>
<td>Immune.io [50]</td>
<td>Tool to protect web apps against application layer attacks.</td>
</tr>
<tr>
<td>WhiteSource [51]</td>
<td>Open source security and licence management software which integrates the building process with security checks and licensing.</td>
</tr>
</tbody>
</table>
C. Migration Procedure

After identifying the levels and understanding the migration strategy, this framework defines the migration procedure from DevOps to DevSecOps which involves the following steps.

- Step I: Initial assessment of entire DevOps process
- Step II: Imbibe Threat Modelling to secure network
- Step III: Harden the development environment
- Step IV: Integrate security tools in development environment and then proceed for coding.
- Step V: Update security definitions and integrate tools to automate code review
- Step VI: Integrate extra layer for compliance and penetration testing by Red Teams
- Step VII: Secure configuration and deployment procedures and automate the process.
- Step VIII: Establish monitoring, intrusion detection mechanism
- Step IX: Setup rollback mechanism in case of incident and app update tools
- Step X: Automate log and incident analysis using tools
- Step XI: Setup repose process, escalation ladder and remediation process using enhanced tools
- Step XII: Integrate tools to predict upcoming security requirement
- Step XIII: Establish feedback mechanism in the entire lifecycle to speed up changes
- Step XIV: Train employees while establishing new procedures and adopting new tools
- Step XV: Threat Intelligence and Vulnerability management program will update security tools
- Step XVI: Continuously optimise the process to maintain speed and agility with security.

D. Support Functions

The other functions of organization need to support the migration frame at different migration points to make sure that a smooth transition takes place in the defined scope. The other functions that supports migration are identified as follows.

- Inventory Management
- Configuration and Patch Management
- Vulnerability Management Program
- Threat Intelligence
- Change Management
- Employee Training
- Event/Incident Management

This model expresses that the involvement of these support functions are inevitable and spreads across the entire life cycle of the product development and deployment. The success factor of this migration primarily depends on the extent of support these functions can provide to the required time frame.

E. Tools

We also identified a list of tools that can aid the migration process. The list is not an extensive list of all tools and can include more when new tools are introduced. Table I shows a list of tools with their features and their role in DevSecOps.

IV. CONTRIBUTIONS OF THE MODEL

This model enforces the very basic concept of “Security by design” by adapting the best migration practices and tools. This research brings forth a framework for the successful migration of DevOps to DevSecOps comprising of all the aspects of migration. This framework is a naïve idea in DevSecOps migration and is the true contribution of our research. Organizations who are thinking of migrating to DevSecOps can use it as a tool for benchmarking their activities with other counterparts.

V. LIMITATIONS AND FUTURE RESEARCH DIRECTIONS

As it is a theoretical framework, no experiments are conducted to validate the framework. There is no models in research offering this kind of migration strategy. The research limitation is this lack of benchmarking model with which this migration framework can be combined.

This research can be extended by identifying the attributes on which the framework can be evaluated and assessing it with the help of real project data.

VI. CONCLUSION

The industry still not reached to that saturation point of DevOps and its adaptation. This is the time where we can align security with DevOps without much pain. The framework explained in this model will aid the organizations to smoothly move to a security intense model of DevOps. The tools identified in the study can assist the organization to automate the process of security integration without making a delay which is the inherent part of security.

REFERENCES


[17] “DevSecOps The road to faster, better and stronger software .”, George V Hulme, Devops.com


[19] M. G. Moore and T. Lip, “DevSecOps Embedded Security Within the Hyper Agile Speed of DevOps What is Dev Sec Ops ? A transformational shift which incorporates secure culture , practices , and tools to drive visibility ,”

[20] “Introduction to DevSecOps Best Practices for Adoption.”, Continio


