SecurityCompass

The CISO's Guide to DevSecOps

Table of Contents

Acknowledgments 1

Foreword 2

Methodology 3

Demographic Distribution of Secondary Sources 3

Demographic Distribution of Surveys 5

Survey roles 5

Survey industries 7

Pt. I: The Importance of DevSecOps 9

Pt. II: Key DevSecOps Challenges and Barriers to Success 10

Lack of Assurance 10

Lack of Quality: 12

Organizational Barriers: 13

Lack of Skills: 14

Insufficient Guidance: 15

Incorrect Assumptions: 16

Pt. III: Overcoming DevSecOps Challenges 18

Policy-to-Execution Platform Defined 18

Creating an Adequate Governance Model 19

Mapping to Business Needs 19

Fostering a Security Culture 20

Creating a Secure Development Pipeline

Using Automation 22

Manual vs. Automated Secure Development Tools 23

Pt. IV: In Conclusion 24

Acknowledgments

We sincerely thank all those who collaborated with us and contributed to this research:

Andrew Laffin, Senior Software Engineer at Valeo Radar Systems Inc.

Alex Smolen, Engineering Manager - Infrastructure and Security at Clever Inc.

Frank Kim, Founder at ThinkSec

Ayhan Tek, Chief Security Architect HBC

Edward Amoroso, CEO at TAG Cyber

Jeremiah Grossman, CEO at Bit Discovery

Stefan Streichsbier, Chief Technical Officer at Numisec

Sahba Kazerooni, CISO at Aviva Canada

Kashif Ali, Information Security at Questrade

Lead Research Authors: Altaz Valani, Clara Christopher

Executive Sponsors: Rohit Sethi, Mike Kologinski

Editors: David Clark, Nathanael Mohammed

Graphic Designer: Vernon Villanueva

Foreword

Note from Research Director

It is clear, as we examine the industry, that there are many challenges in software security. As security professionals, we have a responsibility to respond to this challenge. Our vision at Security Compass is to build a world where technology can be implicitly trusted. One of the gaps we see in achieving that goal is the inability of software teams to properly execute in alignment with business priorities. Today, this discussion is centered around DevSecOps. We decided to conduct research to find out what activities organizations are engaged in, what challenges they are facing, and what they perceive as next steps in the future.

We want to acknowledge a number of other communities that are doing an outstanding job with helping us better understand and manage software security. Among them are OWASP, CERT, SANS, IEEE, SAFECode, SEI, and our friends at ISACA. To them, we tip our hat and thank you for your tremendous effort.

In our opinion, we still have a long way to go in helping organizations manage their software security. Our intent with this research is to share our knowledge and perspective. As with any research, there is always the risk of introducing bias even though we may not be fully aware of it. To keep us honest, we have used numerous sources and individuals to validate our research. One of the challenges in this industry is that obtaining data can be difficult. This is why we decided not to focus solely on any single method of data gathering. Instead, we chose to combine interviews, surveys, books, and examinations of current research literature to provide a holistic picture.

I want to thank those who participated in making this research a reality. You generously gave your time and expertise, which benefits all of us. For those of you interested in participating with us in future research at Security Compass, please reach out. You will find contact details at the end of this report. We would love to engage with you. Our desire is that you gain valuable insights from this report. You may agree or disagree with us, but please continue to engage in the conversation. We need the ongoing diversity of perspectives with all members of the software security community. With that, I want to thank you for taking the time to read the report. I wish you continued success as you try to scale software security in a DevSecOps world.

Sincerely,

Altaz Valani
Research Director, Security Compass

Methodology

Since we are trying to understand the DevSecOps space a little better, the nature of this research is observational. We are trying to discover insights in an open-ended way so that we can eventually derive models and frameworks that can help us better understand the challenges associated with DevSecOps and be able to respond effectively. In that sense, this research is deemed exploratory. We are not yet at a stage where we have globally accepted empirical models for software security. The very nature of this problem is extremely complex. What we are trying to do here is to focus on a very specific area of that vast problem space - namely, how DevSecOps can improve software security. Historically, a lot of discussion in software security has been focused at the project level. We emphasized code scanning, penetration testing, exploratory functional testing, and so on. Today, that discussion has shifted to the program level. We are now interested in scaling up those project level security initiatives. The challenges that emerge from this are rooted in coordination, training, program management, risk, compliance, and so on. With this in mind, our research will focus on role based perspectives. We feel this is important because, in order to achieve alignment, we need to better understand these different perspectives from the project level to the program level and all the way up to the portfolio level.

Demographic Distribution of Secondary Sources

In addition to the survey data that we collected, we also reviewed recent journal articles and conducted interviews. Most of these secondary sources are from recent years. This was a deliberate attempt to understand where the most recent and pressing needs are within the DevSecOps discussion. Where articles were used, we chose to emphasize peer reviewed articles as a further check against quality for our research.

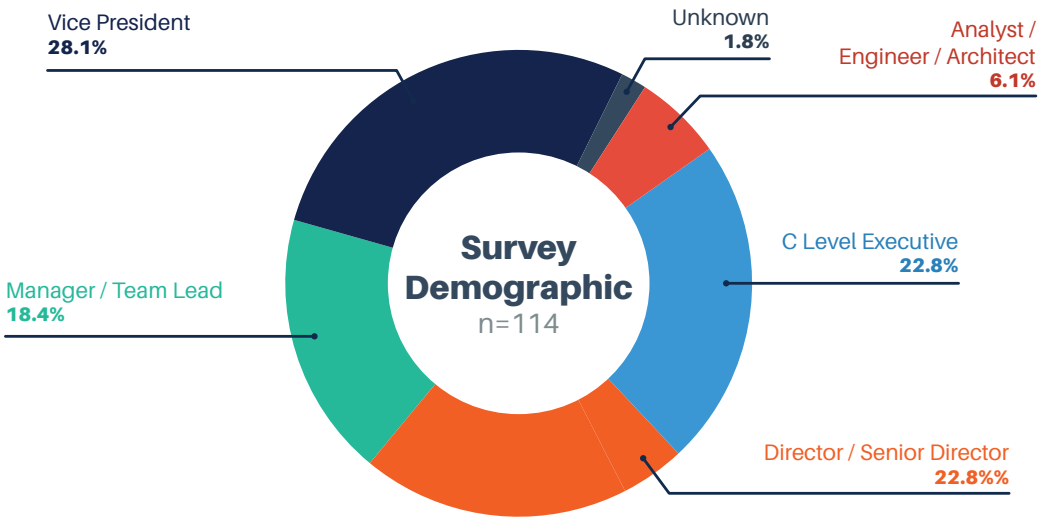
Demographic Distribution of Surveys

As mentioned earlier because we are focused in this research on DevSecOps and alignment, it is important to understand the perspectives of different roles from the project level to the program level to the portfolio level. Not only that, but we felt it was important to obtain data from a number of different industries so the data would not be skewed.

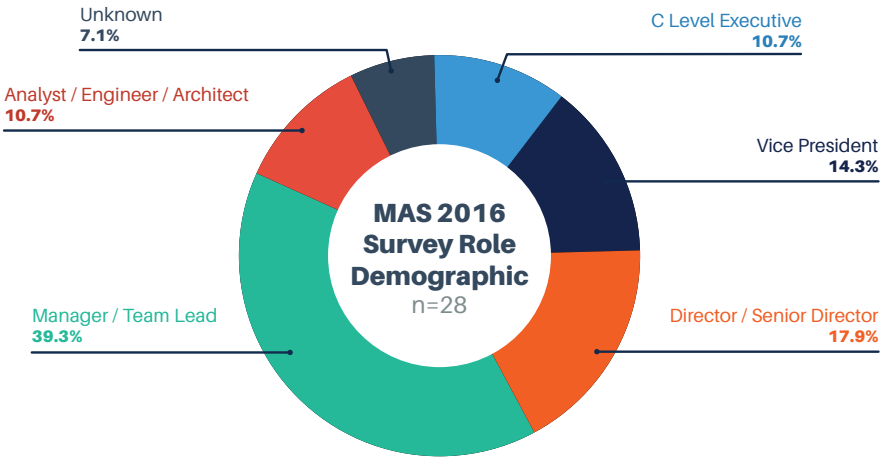
Our surveys were opportunistic surveys. We did not conduct a random sample, but rather, obtained data from our customers and from conference attendees. We found this to be a highly effective way of obtaining data that is often times difficult to extract when using a random sample.

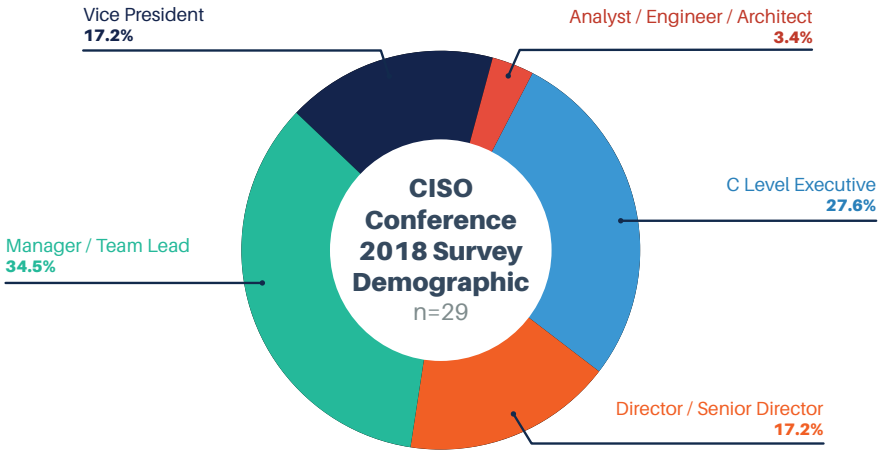
SURVEY ROLES

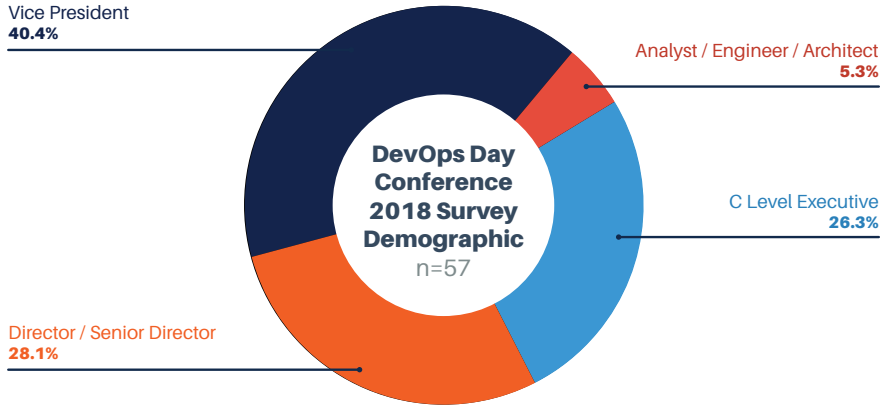
We utilized three surveys for this research. Overall, we were looking for a reasonable distribution across project, program, and portfolio level roles. Below is the combined distribution for all three surveys:



Each data set was further examined to ensure that we did not have the same role overemphasized across all data sets. Below is the distribution of roles across each of the surveys.

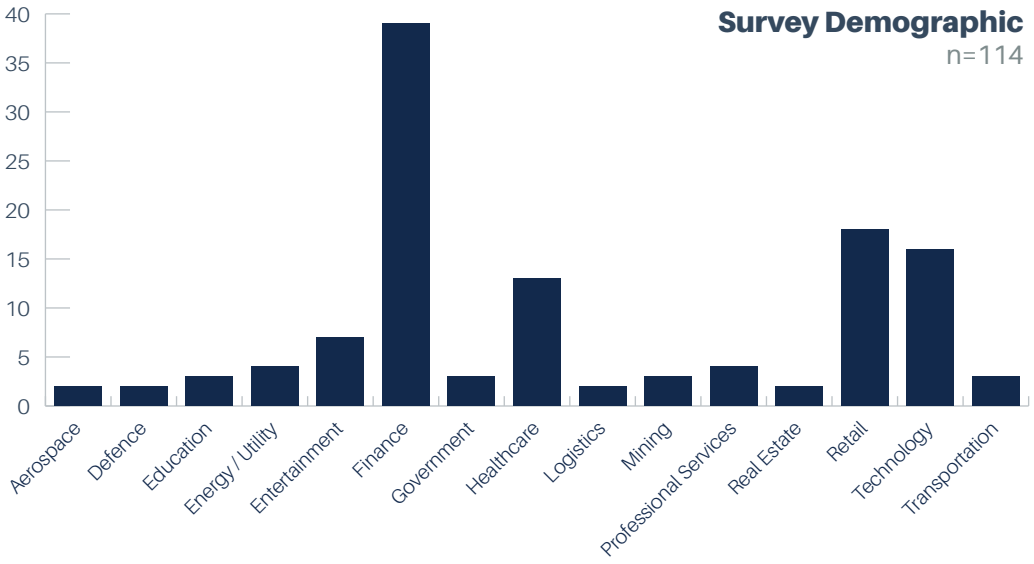




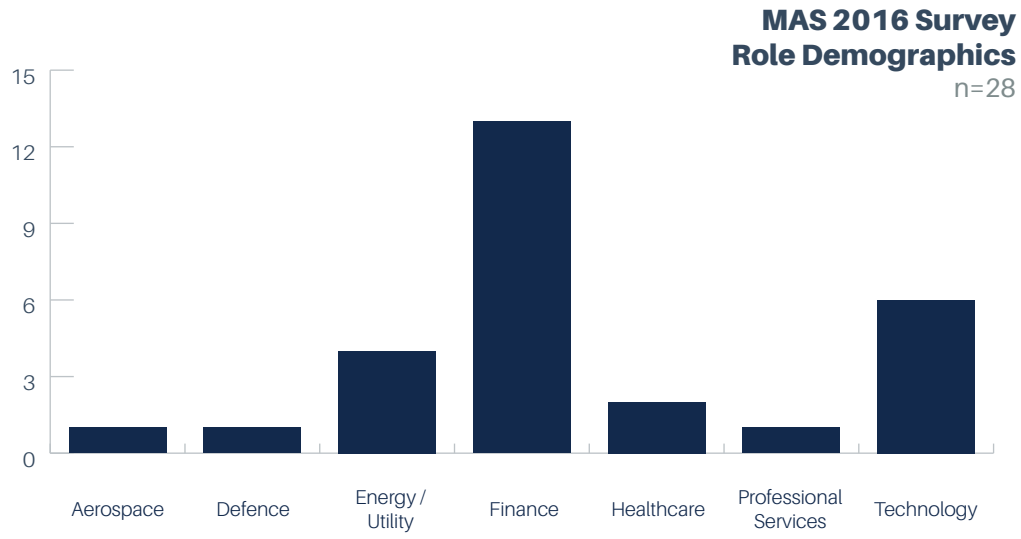


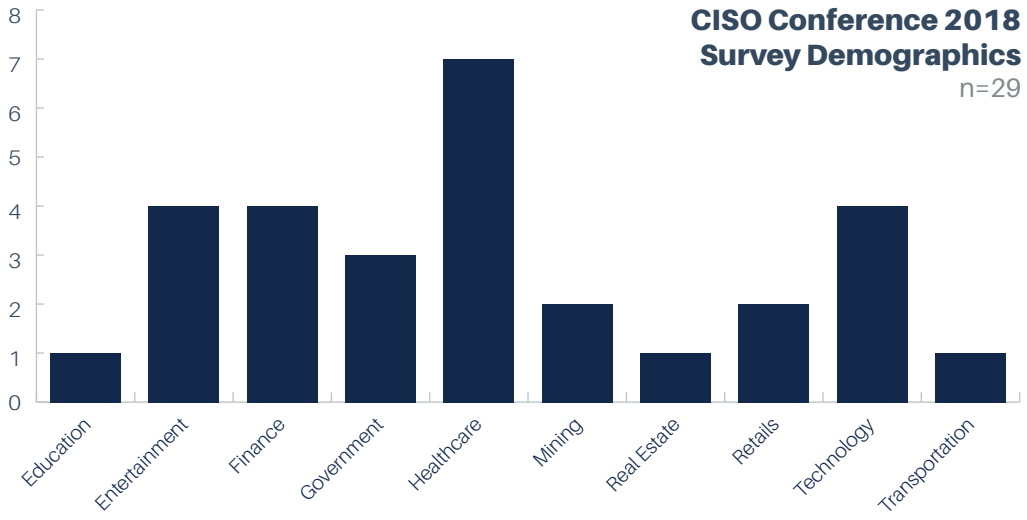
SURVEY INDUSTRIES

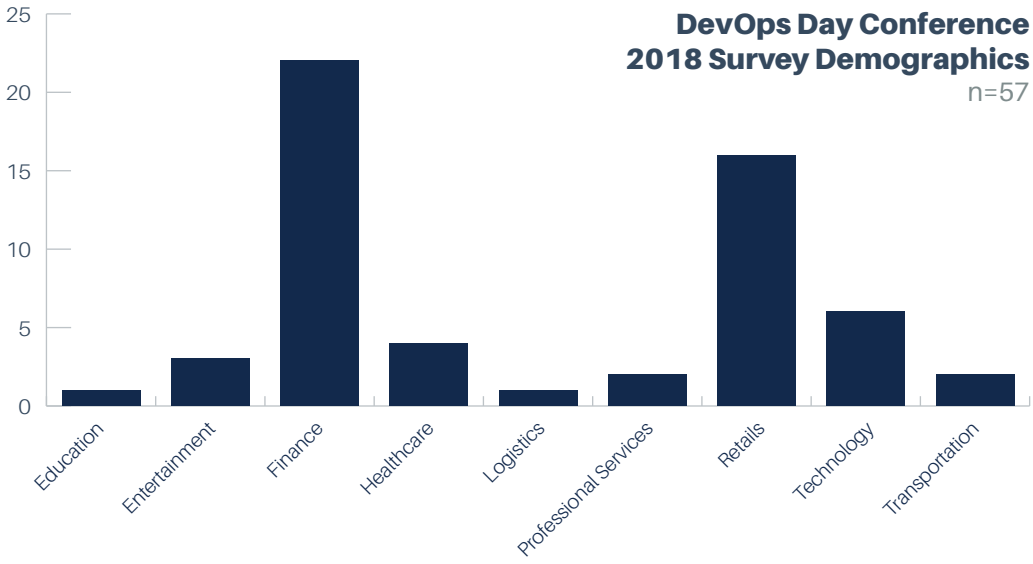
A number of different industries were represented in our surveys. The largest groups were Finance, Retail, Technology, and Healthcare.



The following charts show how various industries were represented within the individual surveys:




CISO Conference 2018
Survey Demographics
n=29
8
7
6
5
4
3
2
1
0
Education
Entertainment
Finance
Government
Healthcare
Mining
Real Estate
Retails
Technology
Transportation


DevOps Day Conference
2018 Survey Demographics
n=57
25
20
15
10
5
0
Education
Entertainment
Finance
Healthcare
Logistics
Professional Services
Retails
Technology
Transportation

Pt. I: The Importance of DevSecOps

In transitioning to a modern development environment, organizations face many challenges related to software security. As the number of security-related incidents continues to increase, security professionals face a growing pressure to respond. However, they cannot handle the overwhelming demands on their own. Security and compliance teams do not communicate with DevOps teams, yet this line of communication is essential for ensuring that security is adequately addressed in an agile environment. As such, many organizations recognize the need for a paradigm shift regarding the way security activities are managed.

The ideal outcome of such a shift would have software teams that act in alignment with business goals, understanding that security practices are integral to business success. A DevSecOps environment is one wherein security is embedded into the DevOps process in a standardized way. Such an environment hinges on cross-functional teamwork, bringing together the roles and responsibilities of technical teams, business teams, and security professionals, to operate more efficiently. As organizations strive toward a DevSecOps environment, program-related efforts start to take priority over project-related efforts. Many such programs include Security Champions, application security program planning and management, as well as the implementation of automated security processes, such as policy-to-execution platforms. These programs all contribute to cross-functional team alignment and maximum organizational efficiency surrounding security.

In our research process, we set out to explore two key questions:

The first question that guided our research was: ***'what are the current challenges in implementing a DevSecOps program?'*** To answer this, we identified the gaps between the way security practices are typically conducted versus the way security practices must be conducted in organizations, in order to meet environmental demands.

The second question we asked was, ***'which components of a DevSecOps program address these challenges?'*** To answer this question, we identified all technical and cultural loose ends that needed to be addressed in order to effect a fundamental, sustainable change in the way security is managed. We also identified which specific tools and services can be leveraged to facilitate program-level DevSecOps incentives.

In this report, we use qualitative primary and secondary research, exploring our own surveys, as well as 3rd party surveys, journal articles, and interviews with industry

analysts, to generate an idea of the challenges organizations face today. In conducting this research, we started with the assumptions that organizations want to move faster, that business and technical teams need to be aligned in order to move faster, and that DevSecOps is an important initiative for those organizations seeking to achieve such an alignment.

Pt. II: Key DevSecOps Challenges and Barriers to Success

Though we have not yet reached a point where we have globally accepted, empirical software security models at our dispense, we were able to identify notable gaps in DevSecOps discussions. To start, we identified the two main pressures driving software security forward: these were market pressures and regulatory pressures. The state of current market pressures is such that software release cycles are faster than ever, creating an urgent need to keep security practices up to speed. Survey data from very large organizations across various roles confirms that this is a problem, most acutely felt by project and program teams.

The state of current regulatory pressures is such that legal security and compliance mandates are becoming more stringent, imposing new penalties on organizations that do not abide. In the face of these pressures, particular security gaps have become manifest. That is, discrepancies have been identified between the current state of security in organizations versus the needed state of security. These gaps and their related challenges are reviewed below.

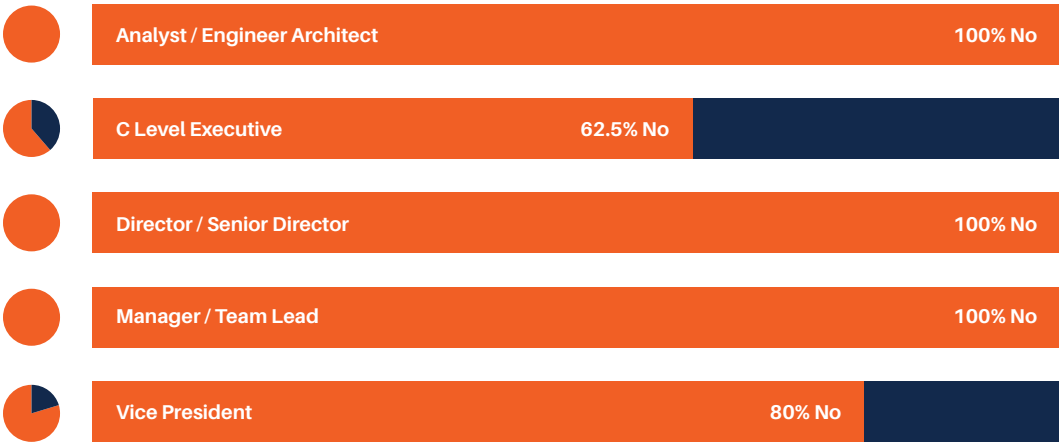
Lack of Assurance

Assurance is defined as the extent to which you can ensure that security is being adequately addressed in your organization. This can be accomplished, for instance, through reports, training, incident response plans, or disaster recovery plans. Our secondary research has shown that, in general, industry assurance models are lacking. That is, there are no standardized procedures for ensuring that your organization is adequately practising security. Of the existing assurance models, they are mainly high-level constructs, like ISO 27001. So, there are adequate models related to governance, but only a few that are related to the specifics of secure application development. The lack of acceptable assurance models has had a

trickle-down effect, leaving a gap where business and project assurance is needed. Only ~50% of CEOs believe that they are well-prepared for a cyber attack[1], and project-level teams are skeptical about their current cybersecurity posture, more so than their senior executives are.

Are you 100% confident in your current cybersecurity posture?

n=27 *(CISO Conference 2018 Survey)*



Whereas 100% of managers, directors, analysts, engineers, and architects reported that they were not entirely confident in their cybersecurity posture, only 62.5% of C level executives and 80% of VPs agreed (CISO Conference 2018 Survey). These studies reflect the general belief that there is a lack of security assurance at the lower levels of business. In a 2017 study ran by Svendsen, it was found that the vast majority, 46%, of organizations re-evaluated risk "only when it fits," while 7.7% said that they did not re-evaluate risk at all. The reported reasons for infrequent risk re-evaluation were lack of knowledge (46.2%), followed by lack of time (30.8%), and lack of budget (23.1%).[2]

1 KPMG. (2018). Growing Pains - 2018 Global CEO Outlook.

2 Svendsen, H. (2017). Security Risk Assessment in Software Development Projects.

Unfortunately, however, O'Neill states, "industry and government continue to increase dependence on software" that is "critical" to the nation's "infrastructure and defense industrial base."[3] Hence, a successful DevSecOps program would need to provide clear communication across roles and different levels of business, so that security is managed effectively. It would also need to offer a way to concretely demonstrate resilience and risk reduction in business. With new, more stringent assurance standards, like the PCI SSF, on the horizon, organizations will need to better prepare in order to protect themselves from legal penalties. Not only will they need a way to ensure that they are thoroughly compliant with current standards, but they will need to do so efficiently, adopting automated methods, if necessary.

Lack of Quality:

Many efforts to improve software quality focus on process improvements and better integration. However, we don't often see security being characterized as a constituent of quality improvement. Additionally, quality assurance teams struggle to provide the necessary assurance to business that the software being released is secure. Though code scanners are a commonly used security tool, they tend to produce false negatives– or failures to alert users when there are, in fact, security defects in code. A 2016 case study revealed that scanners, on average, missed over 50% of security defects.[4] Yet, a 2017 Security Compass survey revealed that static analysis was one of the top three security activities relied upon by large organizations. Lacking or outdated security documentation further compounds the problem because there is no clear source of truth on the security posture of applications.[5] As organizations' applications become increasingly complex and

3 O'Neill, D. (2017). In Search of a Modern Software Life Cycle Secure DevOps Foundations for Large-Scale Software Systems. CrossTalk.

4 Ye, Tao et al. "An Empirical Study on Detecting and Fixing Buffer Overflow Bugs", 2016.

5 Bartsch, S. (2011). Practitioners' perspectives on security in agile development. 2011 6th International Conference on Availability, Reliability and Security.

distributed– to provide greater scalability and functional performance– the number of pathways through an application also increases, creating greater potential for security defects. In the future, DevSecOps programs will need to devise methods to address security at scale in complex application environments. DevSecOps programs will also need to find an efficient, program-based way to rigorously uphold security documentation to improve release quality, treating security as a first-order attribute in quality discussions.

Organizational Barriers:

Organizational barriers present a major challenge when it comes to establishing a DevSecOps program. These barriers include a lack of stakeholder collaboration, difficulty integrating security into existing Agile and DevOps processes, and lack of accountability surrounding security tasks.

Firstly, there's a lack of consistent terminology and understanding about security-related matters amongst stakeholders. As Ramadan points out, "such loopholes may be hard to detect, since traceability mechanisms for security requirements across the different phases are usually not available."[6] That is, "...vulnerabilities may arise from misunderstandings between...stakeholders, in particular due to the divergent use of terminology."

Secondly, organizations generally experience difficulty integrating security into existing Agile and DevOps processes. In fact, more than 25% of surveyed companies expressed that they were willing to adopt DevOps but hesitant to do so due to security and compliance concerns.[7]

Many organizations report a lack of accountability regarding security tasks. In order for a DevSecOps program to function, security must be a priority for everyone. When it came to assessing the priority levels of different members in organizations, analysts, engineers, and architects felt most strongly that achieving software security in a DevOps environment was important. Directors, Vice Presidents, and C Level Executives all rated achieving software security as a lower priority relatively. Thus, in order to improve security accountability in organizations, there needs to be a common set of metrics or traceability mechanisms for stakeholders to collaborate on, as well as a way to seamlessly integrate 'shifting left' into the development process.

---

6 Ramadan, Q., Salnitri, M., Strüber, D., Jürjens, J., & Giorgini, P. (2018). Integrating BPMN-and UML-based Security Engineering via Model Transformation.

7 Mohan, V., &; Ben Othmane, L. (2016). SecDevOps: Is it a marketing buzzword? Mapping research on security in DevOps. 2016 11th International Conference on Availability, Reliability and Security.

LACK OF SKILLS:

When it comes to security practices, there is a lack of skills amongst developers and customers. In fact, customers don't know where to begin asking questions related to security. As Bartsch explains, "non-technical customers often cannot comprehend the technological basis of each security measure."[8] To add, "acquirers complain that they don't know how to ask for secure code from vendors."[9] As a result, if organizations are reliant on customers providing the security requirements, the list will very likely be incomplete.

An even more imminent issue is that developers lack security skills. As O'Neill states, "best practices are insufficient," and standard development education doesn't have enough emphasis on security.[10] And, as Kuper points out, "current remediation techniques are ineffective," with too large of a time window between vulnerability discovery and patching.[11] In the CISO Conference 2018 Survey, analysts, engineers, architects, C level executives, and vice presidents all listed security skills and awareness as the top challenges faced in their application security programs. This challenge stood out above budget challenges, team collaboration challenges, and challenges related to fitting security into the Agile or DevOps process.

A successful DevSecOps program will need to promote secure coding training while finding a way to deliver relevant and easily-understood security requirements for customers to help them better understand security and avoid high remediation costs.

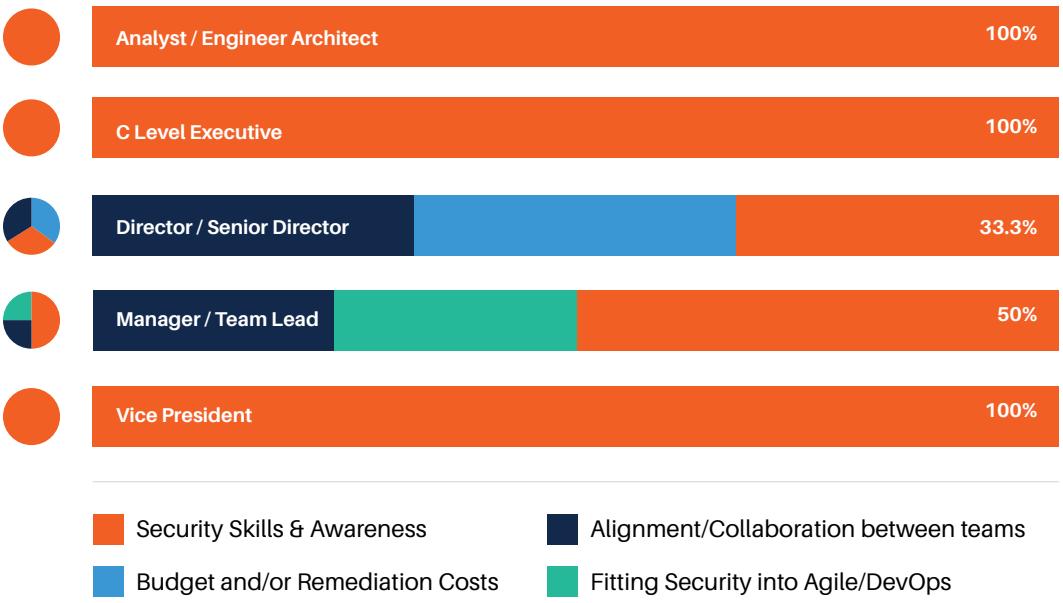
8 Bartsch, S. (2011). Practitioners' perspectives on security in agile development. 2011 6th International Conference on Availability, Reliability and Security.

9 O'Neill, D. (2017). In Search of a Modern Software Life Cycle Secure DevOps Foundations for Large-Scale Software Systems. CrossTalk.

10 O'Neill, D. (2017). In Search of a Modern Software Life Cycle Secure DevOps Foundations for Large-Scale Software Systems. CrossTalk.

11 Kuper, P., & Gannon, T. (2005). The state of security. Security & Privacy, IEEE.

What are the top challenges you face with your Application Security Program?
n=9 *(CISO Conference 2018 Survey)*



INSUFFICIENT GUIDANCE:

When it comes to implementing a DevSecOps program in organizations, there is the challenge of insufficient guidance due to a general lack of security resources, standards, and research. Organizations have generic security guidelines, but nothing to guide their specific implementation of security measures. By extension, there is a general lack of security standards. In some industries, no security standards exist. As Senior Software Engineer at Valeo Radar Systems Inc, Andrew Laffin explains, "industry and government have not yet coalesced around [security] standards."

Even more concerning is the lack of a rigorous body of research to draw from. Jaatun explains that security research is far from adhering to an established scientific approach, as we see in other scientific domains. "The area," he says, "suffers from a lack of credible empirical evaluation." He goes on to say that "there is little evidence [showing] how to implement security practices in the software industry, much less in an agile context."[12]

12 Jaatun, M. G., Cruzes, D. S., &; Luna, J. (2017). DevOps for Better Software Security in the Cloud.

INCORRECT ASSUMPTIONS:

The challenge of incorrect assumptions boils down to a false sense of security in organizations. It's often the case that organizations assume they are sufficiently secure simply because they haven't yet experienced an attack. As Jaatun states, "a major problem in software security is that it is impossible to know all attacks that the system will be exposed [to]."[13] In fact, he says, "uncovered vulnerabilities remain unresolved, often for many years," thus breeding a "false sense that software security is not a big problem," and resulting in the lower prioritization of vulnerabilities compared to other software defects.

Alternatively, some organizations assume that filling out the compliance checkboxes means that they are secure, when, actually, compliance is the starting point. When organizations' employees only take security training "because they have to," not needing to understand it because it lacks ostensible relevance to their job, "it ends up becoming a checkbox [they] have to tick," says Jim Bird, CTO of Bids Trading Technologies Ltd. Actually, being compliant does not necessarily mean that applications are secure and organizations are immune to attack. In this case, use of a policy-to-execution platform can account for all known vulnerabilities before development even starts, reducing potential for incorrect assumptions about the security of applications.

There's also the issue of completely bypassing security to the point that it becomes second nature. When users experience no consequences as a result of this behavior, they assume that it is acceptable. As Pfleeger explains, "paradoxically, security systems have conditioned many individuals to respond to security cues by ignoring or bypassing them whenever possible."[14] That is, the security systems have encouraged bad, rather than good, security habits.

13 Jaatun, M. G., Cruzes, D. S., &; Luna, J. (2017). DevOps for Better Software Security in the Cloud.

14 Pfleeger, S. L., Sasse, M. A., &; Furnham, A. (2014). From Weakest Link to Security Hero: Transforming Staff Security Behavior. Journal of Homeland Security and Emergency Management.

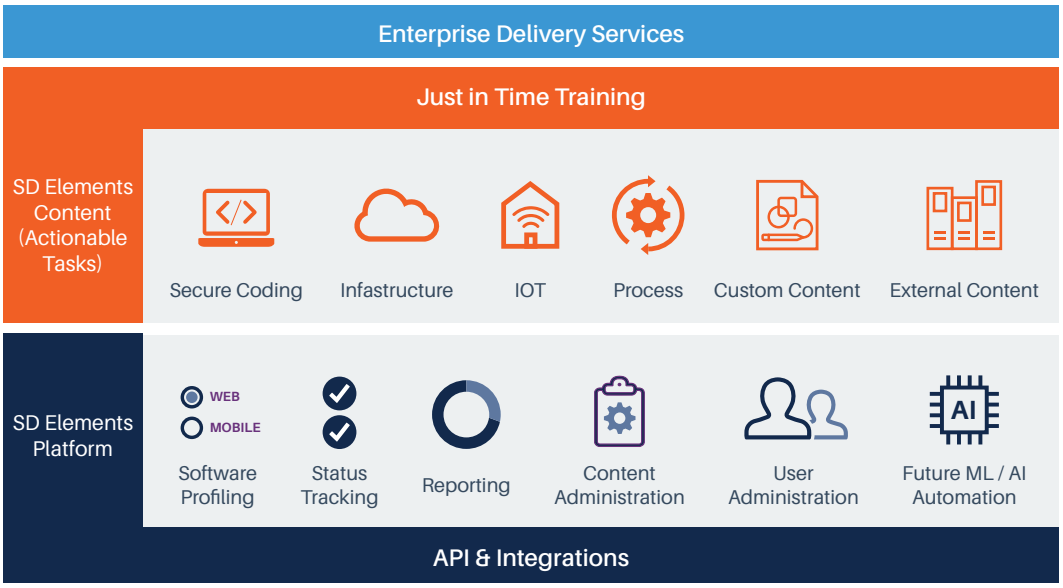
Pt. III: Overcoming DevSecOps Challenges

To address the challenges described so far, we suggest creating a DevSecOps for Application Security Program that focuses on filling the identified gaps. The areas of focus include the establishment of a governance model and a mapping to business needs, the development of a collaborative security culture, and the creation of a secure development pipeline with highly automated processes, including a policy-to-execution platform. We define and elaborate on these program aspects below.

Policy-to-Execution Platform Defined

A policy-to-execution platform is a technology that translates security policy into actionable tasks that developers can complete to ensure their applications are secure by design. Security and risk teams produce policy as their output. These policies are then given to development teams, who are responsible for interpreting the policies and turning them into procedures. Every policy, however, is open to nuances and interpretations, leaving a degree of ambiguity between policy and execution. A policy-to-execution platform fills this gap. Using such a platform, policy teams can define and communicate risk policies for their applications based on regulations, industry standards, and internal policies. Once applications are onboarded to the platform and the risk policy is set, translation is done for development teams, giving them specific actionable guidance about how to build controls that comply with the policy. This means less time is spent trying to interpret what the policy means and more time is spent working to improve security. Of all existing security approaches available to organizations today, policy-to-execution platforms are the most scalable and comprehensive option.

SD Elements Solution



Creating an Adequate Governance Model

Creating an adequate governance model means having the right roles, processes, and controls in place so that every level of the organization can practice security with confidence. Executing this requires thorough program planning and management. Michener points out that: to start, security tools should be run on the code to maintain quality and validate that any changes made are reasonable.[15] By extension, developers should log their actions to ensure that traceability and accountability are upheld. Furthermore, the development environment should be able to track all changes for forensic purposes, in the case of a data breach. To support the adherence to this new governance model, organizations need to clarify who will be designated to these responsibilities.

Mapping to Business Needs

Above all, a DevSecOps program should align with business needs. This alignment typically centers around digital transformation, which entails risk management and compliance. As Founder of ThinkSec, Frank Kim states, “the whole reason DevSecOps exists [is] to drive business value faster.” He goes on to say that “incentives need to be in place so that each team has a reason to care about DevSecOps as it relates to the overall business goals.”

15 Michener, J. R., &; Clager, A. T. (2016). Mitigating an Oxymoron: Compliance in a DevOps Environments. 2016 IEEE 40th Annual Computer Software and Applications Conference.

While a major priority of DevSecOps is to drive business value, many organizations have vastly different priorities and perspectives on which are the most crucial business drivers. In fact, a recently published research report revealed that 27% of business executives believed security investments had a negative return on investment (ROI). IBM Security provides a background context for this negative mentality, claiming that ROI on security is "the trickiest metric."

Nevertheless, the ROI on security can be approximated. The budget for security usually falls somewhere under an organization's IT budget, which is controlled by the Chief Information Officer (CIO). The CIO's main concerns are that IT spend helps to achieve business outcomes. Given that security investment has the capacity to affect business outcomes, there is a business case for it, and this usually centers around loss prevention. On one end, organizations rely heavily on technology, and if these systems go down as a result of a cyberattack, it may result in a significant loss of revenue. Additionally, having security practices in place reduces the cost of compliance audits, the penalties for which are only becoming more stringent. Even some degree of security implementation can ameliorate matters in the event of a breach, since the demonstration that security best practices were followed in good faith can reduce the penalties. On the other end, developers who face stalled release cycles as a result of last minute security tasks will also benefit from a standardized security system that embeds security controls into software, earlier in the software development lifecycle. Lastly, an important consideration is which security tools are part of the investment, as some security tools are significantly more reliable and efficient than others.

Fostering a Security Culture

When it comes to changing organizational processes, people are one of the greatest challenges. One of the major people-related challenges is encouraging different departments, who have never previously collaborated, to start collaborating more. Kim suggests having a DevSecOps evangelist to encourage these new collaborations. Such an individual can explain the importance of security because, as Alex Smolen claims, "if you can't do this, you'll have a hard time because people won't collaborate with you." He goes on to say that, "security is an important part of the business and so everyone from the CEO down should understand this is a part of their job." Jim Bird suggests building a culture of continuous improvement through friendly competition, claiming that "if you set healthy challenges for people to try and create systems that cannot be broken by pen testers, you end up making this part of their responsibility… it sets the benchmark to go after real problems, not theoretical problems or checklist issues."

Another part of building a security culture involves training developers in security. As Pfleeger outlines in her article, failure to adapt to new work demands is usually a result of ambiguity in what needs to be done, vague target goals, and too much demand for change at once.[16] Hence, to sidestep these roadblocks, the security training must strategically avoid them. One solution is to develop security training that acknowledges the long-term focus required to build a talent pipeline. According to Andrew Laffin, a senior software engineer at Valeo Radar Systems Inc., "when you hire somebody," you should expect "to spend 2-3 years training them. Building out good intake training programs is a good way to address the talent gap." This long-term mindset ensures that developers aren't overloaded with too much work at once and that any ambiguities in their work tasks can be clarified, given the reasonable timeframe within which they can learn and experiment.

One approach taken to foster a security culture in organizations is the development of a Security Champions program. Research indicates that existing computer-based security-awareness programs cannot effect the change needed.[17] Security Champions programs take security-awareness one step further, designating one member from a development team to be the 'Security Champion.' This developer acts as the security conscience, leading all security activities on the development side. They help to build a relationship between the security and development teams, while facilitating all necessary communications. They head security-related improvements on their own development teams, and according to the 2017 OWASP Summit, most survey respondents hold Security Champions responsible for security updates, training, threat modeling, risk reporting, and mentoring. The designated Security Champions are trained in security and must meet a specific set of criteria in order to earn this role. Champions are provided with instructor-led training and appropriate course materials to become adequately trained.[18] There are also incentives to become a Security Champion, which may include annual raises, potential for career growth, and other recognitions.[19]

16 Pfleeger, S. L., Sasse, M. A., &; Furnham, A. (2014). From Weakest Link to Security Hero: Transforming Staff Security Behavior. Journal of Homeland Security and Emergency Management.

17 Gartner Report, Designing a Security Champion Report, 2018

18 Gartner, 2017. DevOps Security Champions Help Organizations Gain Leverage Without Training Everyone

19 Gartner, 2017. DevOps Security Champions Help Organizations Gain Leverage Without Training Everyone

Creating a Secure Development Pipeline Using Automation

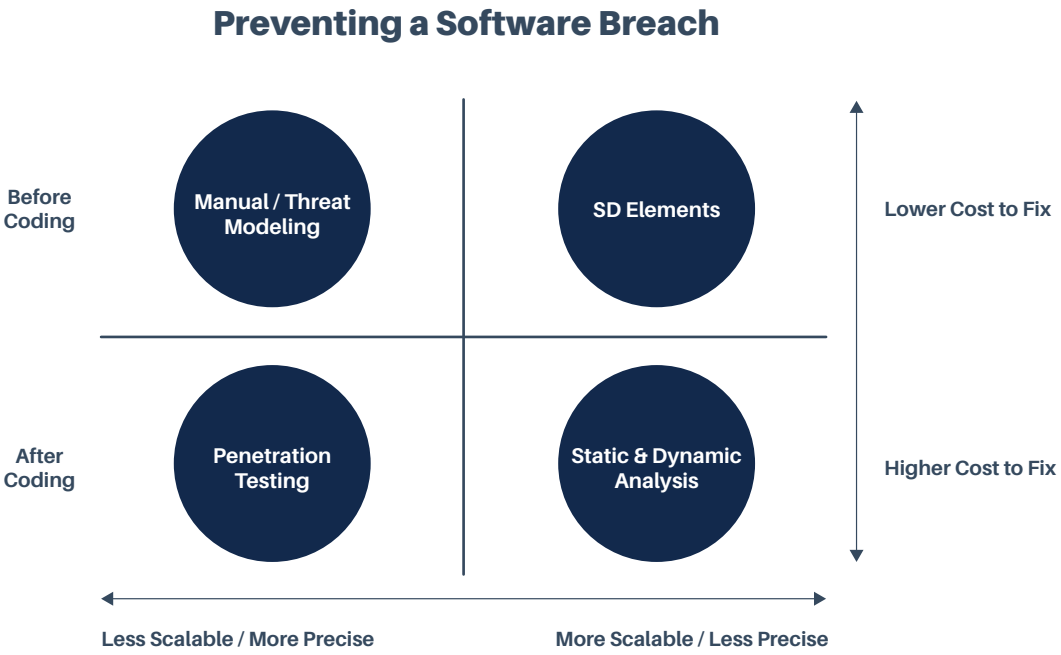
Scaling out means building the critical CI/CD pipeline to operate continuously in order to achieve consistency and quality. As indicated by our Security Compass 2017 survey, 66% of C Level Executives, 60% of Directors, and 50% of Vice Presidents felt that standard tools not working well was a challenge to rolling out an application security program. Industry researchers suggest incorporating several CI/CD elements across the software lifecycle. Some of these include continuous planning, security requirements analysis, architectural risk analysis, static analysis, dynamic analysis, continuous deployment, configuration management, and production support, to name just a few.

As Smolen states, "automation is a really important concept for turning your desires into actual cause/effect behavior in the real world." As a starting point, it's important to build compliance automation. Compliance is the minimum threshold organizations should strive to meet. And, once compliance is met, the goal should be to continue to improve it. Farroha suggests automatic reporting for compliance violations, whereby access is terminated when a certain threshold is surpassed and alarms are set off when a new policy is not accepted.[20] The automation aspect is important because, as Smolen states, "as security teams mature, they should be looking for ways to take policies and intentions" and to transform those "into what runs regularly."

20 Farroha, B. S., &; Farroha, D. L. (2014). A framework for managing mission needs, compliance, and trust in the DevOps environment. 2014 IEEE Military Communications Conference.

MANUAL VS. AUTOMATED SECURE DEVELOPMENT TOOLS

In general, there are 4 main approaches to secure development. The manual approaches include person-driven threat modeling and traditional penetration testing. The automated approaches include policy-to-execution platforms, like SD Elements, Static Analysis Security Testing (SAST), and Dynamic Analysis Security Testing (DAST). Comprehensive approaches include manual threat modeling and penetration testing. These approaches, however, are time-consuming and slow to scale. The most scalable options are policy-to-execution platforms, SAST, and DAST. Static and dynamic analysis, however, are often inaccurate and thus costly in the face of security defects. Of all approaches, policy-to-execution platforms are the most scalable and comprehensive in security coverage. The diagram below outlines all approaches.



Pt. IV: In Conclusion

The question of how a DevSecOps program can improve software security practices in organizations is unprecedented and highly complex. The rapidly changing dynamics of development environments are creating an urgent need for organizational change. This is part of the reason why we see the emergence of DevOps environments, which are inherently cross-functional and more collaborative. As applications grow in number and complexity, the potential for vulnerabilities amplifies, and security becomes more crucial than ever. At the same time, business requires the speed, agility, and continuous improvement that DevOps practices can offer. As such, it's important to consider the possible solutions that a DevSecOps program could offer. Through our research, it was evident that security understanding generally needed to be improved amongst employees, especially developers, and that this understanding needed to be improved in a systematic, efficient way, to keep up-to-speed with development release cycles. At the core of the DevSecOps solution, therefore, is access to security support and learning that's offered in a highly automated fashion. As organizations transition to DevOps environments, it will be important that program-level efforts include more comprehensive governance models that map to business needs, collaborative security cultures, and secure development pipelines which leverage policy-to-execution platforms where necessary.

Security Compass is a company focused on helping companies lower their cybersecurity risk. We offer SD Elements, Advisory and Pen Testing services, and Security Training. Our mission is to build a world where software can be trusted. To find out more about Security Compass or about our platform SD Elements, please contact us here.

SecurityCompass

Security Compass believes in a world where people can trust technology, and it is our mission to help customers proactively manage cybersecurity risk, without slowing down business. Our holistic program offerings are tailored to your organization's needs. Through advisory services, training, and SD Elements, our award-winning policy-to-execution platform, we set you up with all of the resources and tools your organization needs to develop secure software. Security Compass serves some of the world's largest enterprises, including 16 of the largest financial institutions in North America, South America, and Asia, as well as 4 of the largest tech companies in the world.

OFFICES

GLOBAL HEADQUARTERS
1 Yonge Street
Suite 1801
Toronto, Ontario
Canada M5E 1W7

TORONTO
390 Queens Quay W
2nd Floor
Toronto, Ontario
Canada M5V 3A6

CALIFORNIA
1001 Bayhill Drive
2nd Floor
San Bruno, California
USA 94066

NEW JERSEY
621 Shrewsbury Avenue
Suite 215
Shrewsbury, New Jersey
USA 07702

INDIA
#4.07
4th Floor, Statesman House
Barakhamba Road, New Delhi
India 110001

1.888.777.2211
info@securitycompass.com
www.securitycompass.com

@SECURITYCOMPASS
SECURITY COMPASS