Software Security in DevOps: Synthesizing Practitioners' Perceptions and Practices

Akond Ashfaque Ur Rahman
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
+1 860-712-8137
aarahman@ncsu.edu

Laurie Williams
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
+1 919-513-4151
williams@csc.ncsu.edu

ABSTRACT
In organizations that use DevOps practices, software changes can be deployed as fast as 500 times or more per day. Without adequate involvement of the security team, rapidly deployed software changes are more likely to contain vulnerabilities due to lack of adequate reviews. *The goal of this paper is to aid software practitioners in integrating security and DevOps by summarizing experiences in utilizing security practices in a DevOps environment.* We analyzed a selected set of Internet artifacts and surveyed representatives of nine organizations that are using DevOps to systematically explore experiences in utilizing security practices. We observe that the majority of the software practitioners have expressed the potential of common DevOps activities, such as automated monitoring, to improve the security of a system. Furthermore, organizations that integrate DevOps and security utilize additional security activities, such as security requirements analysis and performing security configurations. Additionally, these teams also have established collaboration between the security team and the development and operations teams.

Categories and Subject Descriptors. D.2.0 [Software Engineering]: Design-Methodologies

Keywords. DevOps; security; software practices; survey

1. INTRODUCTION
DevOps is a software process that emphasizes collaboration within and between different teams involved in software development [2]. According to a study from CA Technologies [22], 88% of 1425 organization executives stated that they have adopted DevOps, or are planning to adopt DevOps in the next five years. According to Puppet Labs' 2015 State of DevOps Report [10], organizations that have adopted DevOps experienced 60 times fewer failures and deploy 30 times more frequently than organizations that have not adopted DevOps. Despite the popularity and perceived benefits, software security aspects of DevOps remain a concern for organizations that want to adopt DevOps [22]. In organizations that use DevOps practices, developers can commit and deploy their software changes at a rapid rate using an automated pipeline [6]. For example, in Facebook developers can deploy software changes up to 500 times a day [4]. At such a rapid rate, if the security team operates in isolation without close collaboration with the development and operations teams, then the rapidly deployed software changes might not undergo the adequate security reviews, potentially leading to vulnerable software. Bringing security principles within the DevOps process can help the organization in achieving better quality of software by integrating security checks into the phases of development, testing, and deployment.

The goal of this paper is to aid software practitioners in integrating security and DevOps by summarizing experiences in utilizing security practices in a DevOps environment.

According to Moore [16], organizations often prefer to learn through the experiences of other organizations that belong to the same industry. Thus, organizations that are considering adopting DevOps can also benefit from a study that identifies the names of organizations that have adopted DevOps and are using software practices to integrate security.

We state the following research questions:

RQ1: *Perception.* How do software practitioners perceive the integration of DevOps and security? What DevOps related activities contribute to those perceptions?

RQ2: *Security Practices*. What security practices are used by organizations that integrate security into DevOps?

We answer these research questions by first selecting and analyzing a set of 66 Internet artifacts, such as blog posts, conference presentations, and video presentations. We then identified the perceptions of DevOps towards the system's security and DevOps related activities that contribute towards these perceptions. We also identified a set of software practices used to integrate security in DevOps. Leveraging findings from our analysis of Internet artifacts, we created a survey to further investigate perceptions of DevOps towards the system's security, and the activities that contribute to such perceptions. The survey was administered to representatives of nine organizations that have adopted DevOps. Software practitioners have interchangeably used the two terms 'activity' and 'security practice', but we differentiate between the two terms 'activity', and 'security practice' to facilitate the discussion in the paper. In our paper, an activity focuses on achieving a small, well-defined goal that has a tangible output. For example, 'automation of testing' is an activity. A security practice is a collection of activities that can be grouped based on existing similarities within those activities. For example, 'use of automation activities' is the practice that contains the activity of 'automation of testing'.

We summarize the contributions of this paper as follows:

- A list of DevOps activities that seem to have a positive and negative impact on the security of a system;
- A list of security practices and an analysis of how they are used in organizations that have adopted DevOps to integrate security; and
- An analysis that quantifies the levels of collaboration amongst the development teams, operations teams, and security teams within organizations that are using DevOps.

The rest of the paper is organized as follows: Section 2 provides necessary background and prior work that are related to our study. In Section 3, we describe our methodology. In Section 4, we provide findings from our study. We describe the limitations of our paper in Section 5. In Section 6, we discuss several observations from our study.

2. BACKGROUND AND RELATED WORK

In this section, we provide background information and prior academic work relevant to our study.

2.1 Background

Smeds et al. [21] defined DevOps as a collection of software engineering activities such as, continuous planning, and continuous deployment that are supported by cultural facilitators such as sharing of responsibility and goals, and technical facilitators such as automated build process, and automated configuration management. According to Dyck et al. [2], DevOps is the software process that emphasizes collaboration within and between different teams involved in software development. For the rest of the paper, we refer to organizations that use DevOps to deliver software and services as *DevOps organizations*.

Software practitioners have stressed the importance of integrating security in DevOps. As a result, the term *DevSecOps* has gained popularity recently. In April 2012 Turnbull [23] introduced the concept of collaboration between security teams and every other team inside the organization.

In this paper, we use the term DevSecOps to refer to the concept of integrating security principles through increased collaboration between the development teams, operations teams, and security teams of a DevOps organization.

To facilitate our discussion in the paper we differentiate between the two terms 'activity', and 'security practice'. In this paper, a DevOps activity focuses on achieving a small, well-defined goal that has a tangible output. A security practice is a collection of activities that can be grouped based on existing similarities within those activities.

2.2 Related Work

Prior studies have discussed software practices used in DevOps and continuous deployment, security aspects of Agile methodologies, and security initiatives proposed for organizations. Feitelson et al. [4] in their work reported development and deployment practices conducted in Facebook, and stated how software changes related to end-user privacy is handled differently by limiting the deployment rate. Smeds et al. [21] described barriers of adopting DevOps amongst organizations, but did not consider the security aspects of DevOps. Rahman et al. [18] investigated the usage frequency of 11 software practices used amongst software companies that implement continuous deployment. In our study, we have listed the DevOps related activities that can contribute positively or negatively for software security.

In the Building Security In Maturity Model (BSIMM) McGraw et al. [14], studied the security initiatives of 78 organizations and listed their activities related to software security. Epstein et al. [3] identified 13 software practices that are detrimental to software security for organizations that use service-oriented architecture and provide software services. Bartsch [1] explored the perceptions of software security amongst Agile practitioners and observed the importance of appropriate involvement of customers, and continuous improvement, to implement software security amongst Agile practitioners. Our paper focuses on investigating the perceptions of software practitioners towards DevOps with respect to the system's security, and reporting the practices that have been used to integrate security for DevOps.

3. RESEARCH METHODOLOGY

We describe the major steps of our study in this section. Our first step was to identify and analyze Internet artifacts. We then conducted a survey[1] with nine DevOps organizations to further investigate perceptions and security practices that they use to integrate security.

3.1 Analysis of Internet Artifacts

We used the Google search engine to identify Internet artifacts, such as blog posts, conference presentations, and video presentations. Initially we used the search string "security in DevOps" to identify the necessary set of Internet artifacts in the study. We then extended our list of search strings based on two observations:

- From the top 50 search results for the search string "security in DevOps" we observe that security in DevOps is also referred as "DevSecOps", "SecDevOps", "SecOps", and "RuggedOps"
- Rahman et al. [18] have attributed continuous deployment and continuous delivery to be related to DevOps

Considering the above two observations, we utilized seven search strings to identify the necessary Internet artifacts for the study: "security in DevOps", "DevSecOps", "SecDevOps", "SecOps", "RuggedOps", "Security in Continuous Delivery", and "Security in Continuous Deployment".

Next, we collected the top 50 search results for each of the seven search strings. We excluded an Internet artifact from the study if all of three conditions were true for the artifact:

- the artifact did not discuss the benefits of DevOps activities for the system's security;
- the artifact did not discuss the negative effects of DevOps activities for the system's security; and
- the artifact did not discuss what software practices can be used to integrate security in DevOps.

From the identified Internet artifacts we investigated what DevOps activities have software practitioners' considered beneficial and negative to the security of the system. To identify these DevOps activities we separated the activities that are considered as beneficial and detrimental to the system's security in two separate lists. If multiple activities were mentioned in different artifacts then we include that activity only once, and keep track of how many Internet artifacts in which the activities were mentioned.

We also investigated what security practices have software practitioners stated to integrate security in DevOps. To identify these security practices, we read the Internet artifacts and listed each practice. If different terminologies were used for the same

[1] http://goo.gl/forms/hH1PuRmg7a

practice, then that particular practice is included once for our study. For example, the two terms 'automation of monitoring', and 'automation of testing' fall under the same practice 'use of automation activities'.

In some Internet artifacts, software practitioners have reported different activities that constitute a certain practice. To find such activities we read the artifacts, and document each new activity. If different terminologies are used to mention the same activity, then we list that activity only once. For example, if 'use of automated linters', and 'use of automated code review' were mentioned in same or different Internet artifacts then we include the activity 'automation of code review' once.

3.2 Survey

We surveyed one representative of each of nine organizations that utilize DevOps practices to investigate their perception of DevOps activities with respect to system's security. We also queried the survey respondents about their use of security practices. Please recall that in our paper, an activity focuses on achieving a small, well-defined goal that has a tangible output, and a security practice is a collection of activities that can be grouped based on existing similarities within those activities.

We designed the survey based on our findings from our analysis of Internet artifacts. In the survey, we asked the participants to identify the DevOps activities that they think can help positively towards software security. We asked them to select from a list of activities that we obtained from our analysis of Internet artifacts in form of multiple check boxes. We also provided the option for a free form text to mention additional activities not included in the list. From our analysis of Internet artifacts, the count of DevOps-related activities contributing negatively to system's security are small when compared to that of the DevOps activities that contribute positively. As a result, we only used a free form text to identify the activities that can contribute negatively to the system's security.

In the survey, we asked the survey participants which of the identified security practices they use, along with necessary activities. The survey participants were also provided the option to mention any additional activities that they use to integrate security.

As DevOps necessitates increased collaboration between teams [21], in the survey we asked three questions that assess collaboration between:

- development and operations teams;
- development and security teams; and
- security and operations teams.

We used a Likert Scale [11] from one to five, where five indicated the highest level of collaboration, and one indicated the lowest. If the DevOps organization did not have any one of the above-mentioned teams, we asked the survey participant to assign zero. Later in our analysis we refer to the collaboration ratings of five, four, three, two, and one respectively as 'highest', 'high', 'moderate', 'low', and 'lowest'.

4. RESULTS

In our study, we collected 350 Internet artifacts that included blog posts, conference presentations, and video presentations. After eliminating artifacts through the use of our exclusion criteria, we included 66 Internet artifacts in our study. The complete list of Internet artifacts used in the study is available online[2]. In our study, 55 of the 66 Internet artifacts were blog posts, five were conference presentations, five were Slideshare presentations, and one was a video presentation.

In our study, we surveyed one software practitioner from each of the nine DevOps organizations. These organizations were: CA Technologies, Cisco Systems, CoolBlue, Facebook, Google, LexisNexis, Mozilla Firefox, Netflix, and SAS.

We organize the following two sub-sections to present results related to RQ1 and RQ2, respectively, based upon our analysis of Internet artifacts and survey.

4.1 RQ1: Perception

In this sub-section, we present the activities that contribute to software practitioners' perceptions of DevOps to the system's security based upon an analysis of Internet artifacts and the survey.

4.1.1 Analysis of Internet Artifacts

From our analysis of the Internet artifacts, we found that in 32 of the 66 artifacts software practitioners discussed the perceptions of DevOps towards a system's security. Software practitioners stated five DevOps activities that can improve a system's security. We present these activities in Table 1. The column 'References' presents the count of Internet artifacts in which the author referred the corresponding activity. As shown in Table 1, use of automated monitoring was the most referred activity, followed by use of automated pipeline. We observe the concept of automation getting mentioned as four of the five top activities in 27 artifacts.

Table 1: How DevOps positively impacts software security?

Activity	References
Use of automated monitoring	13
Use of automated pipeline to deploy software	8
Automatic deployment of software	3
Automatic testing of software changes	3
Delivering software in small increments	3

We also found three DevOps activities that artifact authors felt caused a negative impact on a system's security, as presented in Table 2. In Table 2, the column 'References' presents the count of Internet artifacts that referred to the corresponding activity.

Table 2: Why DevOps can be detrimental to software security?

Activity	References
Use of immature automated deployment tools	2
Use of inappropriate software metrics	2
Inadequate monitoring of collaboration	1

4.1.2 Analysis of Survey

Table 3 presents DevOps activities mentioned by the survey respondents that can improve a system's security. In Table 3, we present two columns: 'Yes', and 'No'. 'Yes' represents the count of survey respondents who consider a specific activity to be contributing positively to system's security, and 'No' represents the count of survey respondents that do not consider that specific

[2] http://www.realsearchgroup.org/realsearch/devsecops-ref/

activity to be positively contributing towards system's security. As shown in Table 3, automated monitoring is the most referenced activity that can makes DevOps beneficial to a system's security. Apart from automated monitoring, six or more of the nine survey respondents have considered automated deployment, and automated pipeline to be activities that can contribute positively to a system's security.

Table 3: DevOps activities that are beneficial for security

Activity	Yes	No
Use of automated monitoring	8	1
Use of automated pipeline to deploy software	7	2
Automatic deployment of software	6	3
Automatic testing of software changes	5	4
Delivering software in small increments	5	4

Our survey respondents stated different DevOps activities that can contribute negatively to a system's security. According to five survey respondents, the fast deployment of software can contribute negatively for DevOps with respect to a system's security. These software practitioners argued that to ensure rapid deployment of software, DevOps organizations might overlook crucial security techniques, for example performing security tests or performing penetration tests.

Increased collaboration amongst different teams is another activity that can contribute negatively to system's security, according to two of the nine survey respondents. Collaboration between the development teams, and operation teams implies that some members of both teams might get unrestricted access to different parts of the system, enabling them to intentionally or un-intentionally harm system's security. One survey respondent identified the use of third party libraries to deploy software products, to be an activity that is detrimental to system's security. Often these third party libraries are not thoroughly tested to identify security vulnerabilities and might lead to rapidly deploying vulnerable software.

4.2 RQ2: Security Practices

Similar to Section 4.1, we answer RQ2 using our analysis of Internet artifacts, and analysis of survey that are presented respectively in Section 4.2.1, and 4.2.2.

4.2.1 Analysis of Internet Artifacts

We found 34 Internet artifacts that discuss the use of security practices for integrating security in DevOps organizations. From our analysis of these 34 Internet artifacts, we observe four security practices: use of automation activities; collaboration amongst different departments of the software organization; providing security training for development team members; and use of non-automated security activities. We describe these practices briefly in the following four sub-sections.

4.2.1.1 Use of Automation Activities

Automation of all activities related to software development is one of the common software practices used in DevOps culture. In this paper we refer to this security practice as 'use of automation activities'. This security practice includes five automation activities. We list the automation activities that were mentioned in the studied Internet artifacts with definitions, as following:

- *Automation of Code Review*: Code review is the activity of presenting source code changes for comment, approval, and improvisation [7]. Automation of code review is the activity of performing code review, and giving appropriate feedback to the software developers of interest, using open source and commercial static analysis tools [13].
- *Automation of Monitoring*: Automation of monitoring refers to the activity of gathering, reporting, and storing system-related information such as CPU usage and memory usage for further analysis using automated tools [7].
- *Automation of Software defined Firewall*: Automation of software defined firewall is the activity to maintain consistent policies to manage the settings of the organization's firewall using automated tools [15].
- *Automation of Software Licensing*: Software licensing is the activity of enabling users to purchase, install, and use software in accordance with a set of conditions set by the software vendor [5]. We define automation of software licensing as the activity that ensures users are purchasing, installing, and using the software as per the conditions set by the software vendor of interest, using automated tools.
- *Automation of Testing*: Automation of testing refers to the activity of automatically performing testing tasks, such as test case management, test monitoring and control, and test data generation, for different types of tests namely, functional testing, integration testing, and unit testing [9].

Several Internet artifacts have referenced these automation activities, as shown in Table 4. According to Table 4, automated monitoring is the most referred automation activity amongst the studied Internet artifacts. Apart from automated monitoring, automated testing and automated code review, are other automation activities that are referred in more than 10 Internet artifacts.

Table 4: List of automation activities

Name	References
Automation of monitoring	20
Automation of testing	13
Automation of code review	11
Automation of software licensing	5
Automation of software defined firewall	3

4.2.1.2 Collaboration

The practice of actively collaborating with other teams is another practice that software practitioners have stated in the Internet artifacts. Increased collaboration between development teams, security teams, and operation teams have been mentioned in 16 Internet artifacts. In 13 Internet artifacts, authors reported that instead of operating in silos, the security team could adopt existing DevOps automation activities inside the organization, and customize existing security tools in a way that ensures the feedback cycle between the security team, with the other teams is short. In another Internet artifact, an author stated developers could learn from the security team and build or customize the necessary security tools by themselves.

4.2.1.3 Providing Security Training

Software practitioners referred security training for development team members to integrate security in DevOps organizations. This practice was mentioned in three Internet artifacts. Completing relevant online coursework, attending developer boot camps, and in-house security awareness meetings are the three activities that the authors mentioned on implementing this practice.

4.2.1.4 Use of Non-Automated Security Activities

We identified 10 security activities from the Internet artifacts of interest. We describe these security activities with definitions as following:

- *Design Review*: Design review is the activity of reviewing the design of the entire software as well as different modules of the software to identify potential security flaws that might be exposed at latter stages of software development [12].
- *Input Validation*: Input validation is the activity of performing data validation, and rejecting non-conformant data that are both entering and exiting the software of interest [20].
- *Isolation of Untrusted Inputs*: Isolation of untrusted inputs is the activity of identifying, and performing security measures on resources that are not verified as secure by the system vendor for example, third party library used to develop the software [19].
- *Performing Compliance Requirements*: Performing compliance requirements is the activity that continuously checks if the software of interest satisfies the federal regulations set by the government as per the domain of interest such as healthcare, and trade organizations [8].
- *Performing Security Configurations*: Performing security configurations is the activity of identifying potential resources that contain configuration information related to the software, and securing them using security tests [12].
- *Performing Security Policies*: Performing security policies is the activity of ensuring all software related information is only accessible to entities with appropriate level of authorization [19].
- *Security Requirements Analysis*: Security requirements analysis is the activity of identifying a set of capabilities that must be possessed by the software to satisfy a set of specifications that ensures prevention of intentional or unintentional unauthorized access to the software [12].
- *Performing Manual Security Tests*: Performing manual security tests is the activity that aims to reduce software risk by applying two tasks: ensuring that the software's functionality is properly implemented, and executing risk-based security testing via simulating an attacker [17]. Security tests such as penetration testing can be performed in an automated or non-automated fashion to systematically compromise different parts of the software. We do not include performing manual security tests as part of the 'use of automation activities' practice as this activity is non-automated.
- *Risk Analysis*: Risk analysis is the activity of creating design specifications relevant to security and later on testing those design specifications [12].
- *Threat Modeling*: Threat modeling is the activity of identifying, describing, and categorizing threats along with the actors or agents who are associated with those threats [12].

Several Internet artifacts have referenced these security activities as shown in the 'References' column of Table 5. According to Table 5, performing security requirements analysis was the most referenced security activity amongst the 10 non-automated security activities of interest.

Table 5: Security activities referred in Internet artifacts

Name	References
Security requirements analysis	6
Performing security configurations	5
Performing security policies	5
Performing manual security tests	5
Performing compliance requirements	4
Design review	3
Input validation	3
Isolation of untrusted inputs	3
Threat modeling	3
Risk analysis	2

4.2.2 Analysis of Survey

Here we discuss or findings from analyzing the survey. First we discuss the usage of four security practices amongst the nine organizations. Then we provide the details of practice usage in terms of activities.

Our survey analysis states that all of the nine DevOps organizations use three of the four identified security practices. Two of the nine DevOps organizations do not use the practice of providing security training for development team members.

4.2.2.1 Use of Automation Activities

Table 6 presents the use of automation activities amongst the nine DevOps organizations. From Table 6, we observe automation of monitoring, and automation of testing are the two most frequently used automation activities amongst the nine DevOps organizations. None of DevOps organization reported any additional automation activity that was not included in the survey.

Table 6: Use of automation activities

Name	Yes	No
Automation of monitoring	8	1
Automation of testing	8	1
Automation of code review	7	2
Automation of software defined firewall	6	3
Automation of software licensing	4	5

4.2.2.2 Collaboration

All of the nine survey respondents have reported to have separate development and security teams in their organizations. Eight of the nine representatives have reported to have a separate operations team. In Table 7, we present our findings for three types of collaborations namely, 'Dev&Ops', 'Dev&Sec', and 'Sec&Ops'. For each of the nine organizations, Dev&Ops presents the level of collaboration between development and operations teams, Dev&Sec presents the level of collaboration between development and security teams, and Sec&Ops presents the level of collaboration between security and operations teams. Each cell in the table presents the count of DevOps organizations for a certain type of collaboration. For example, for the collaboration type 'Dev&Ops', five of the nine DevOps organizations reported their collaboration as 'High'. A survey respondent of one DevOps organization reported not having a separate operations team, and responded with a zero for

collaboration types Dev&Ops and Sec&Ops. We exclude that survey response in Table 7.

Table 7: Level of collaboration

	Lowest	Low	Moderate	High	Highest
Dev&Ops	1	0	1	5	1
Dev&Sec	0	2	4	1	2
Sec&Ops	0	0	4	3	1

4.2.2.3 Providing Security Training

According to our survey results, seven of the nine DevOps organizations provided security training for their development team members.

4.2.2.4 Use of Non-Automated Security Activities

Table 8 presents the frequency of security activity usage amongst the nine DevOps organizations of interest. As shown in Table 8, performing security policies, and performing manual security tests are the two most frequently used security activities. From Section 4.2.1.4, our analysis of Internet artifacts identified use of security requirements analysis as the most frequently referenced security activity amongst the Internet artifacts of interest. Five of the nine DevOps organizations have reported to use this security activity. None of DevOps organization reported any additional security activity that was not included in the survey.

Table 8: Use of security activities

Name	Yes	No
Performing security policies	9	0
Performing manual security tests	8	1
Input validation	7	2
Performing compliance requirements	7	2
Performing security configurations	7	2
Risk analysis	7	2
Isolation of untrusted inputs	6	3
Threat modeling	6	3
Design review	5	4
Security requirements analysis	5	4

5. LIMITATIONS

In our study, we cannot claim that the set of Internet artifacts is complete as we used seven search strings to collect the necessary Internet artifacts. We do not claim that the identified security practices for integrating security in DevOps is complete. Since the number of surveyed organizations is small, we cannot strongly claim our findings are generalizable. We did not study if there is any relationship between the use of automation activities, and quality of software deployed by the nine DevOps organizations of interest. We also did not discuss whether level of collaboration between different teams had an impact on use of the four security practices, the five automation activities, or the ten security activities. We leave the scope of pursuing these limitations as research guidelines for future work.

6. DISCUSSION

We use this section to discuss our findings from our analysis of Internet artifacts and the conducted survey.

- According to our analysis of Internet artifacts, use of automated monitoring was the most frequently mentioned activities that software practitioners perceive beneficial to system's security. The majority of the survey respondents also echoed this observation; eight of the nine survey respondents stated automated monitoring to be one of the DevOps activities that are beneficial to system's security. Velasquez et al. [24] in their study identified automated monitoring to be one of the 'top' activities amongst DevOps organizations. Other DevOps activities such as use of automated pipeline, and automated testing, were identified as benefactors to system's security in Internet artifacts, as well as by five or more of the nine survey respondents.

> **Observation 1** - Commonly used DevOps activities, such as automated monitoring, automated testing, and automated deployment of software can be helpful to a system's security.

- In one Internet artifact, collaboration was mentioned as an activity that negatively impacts system's security. This view was also echoed by two of the survey respondents. When teams collaborate too closely, individuals might inappropriately get access to system resources, and accidentally or deliberately change system properties and hurt system's security.

> **Observation 2** – Unrestricted collaboration might lead to inappropriate access to system resources, which may hurt a system's security.

- We observe that software practitioners have mixed opinions about automated deployment of software with respect to system's security. The analysis of Internet artifacts and the survey indicated that, software practitioners find automated deployment to be beneficial to system's security. However, both the analysis of artifacts and survey revealed a security concern related to automated deployment if the DevOps organization uses improper automated deployment tools, or overlooks the need of security practices in the desire to deploy software rapidly. From these two seemingly opposite observations we state that the system's security might benefit from automated deployment with adequate supervision from the security team, and use of proper deployment tools.

> **Observation 3** – Supervised collaboration with security team might help in making automated deployment beneficial to system's security as a DevOps activity.

- From the survey results found in Table 7, we observe seven or more of the nine DevOps organizations having 'Moderate' or higher levels of collaboration between the security team and the other two teams: development, and operations. This finding echoes our findings from our analysis of Internet artifacts from which we identified collaboration between all the teams, as a security practice to integrate security in DevOps.

Observation 4 – Security teams actively collaborate with development and operations teams in established DevOps organizations.

- We identified 10 non-automated security activities from our analysis of Internet artifacts. From Table 8 we observe that five or more of the nine DevOps organizations are using all the 10 non-automated security activities. We observe a certain level of consensus between the stated non-automated security activities in Internet artifacts, and the security activities that are actually in use within DevOps organizations.

Observation 5 – Security awareness is prevalent amongst established DevOps organizations, considering their use of security activities, such as performing security policies, performing manual security tests, and performing security configurations.

7. ACKNOWLEDGEMENTS

The National Security Agency (NSA) Science of Security Lablet at the North Carolina State University funded this work. We thank all the software practitioners who participated in our survey. We also thank the Realsearch research group for providing helpful feedback on this paper.

8. REFERENCES

[1] Bartsch, S. 2011. Practitioners' Perspectives on Security in Agile Development, in *Proc. of the 6th International Conference on Availability, Reliability and Security (ARES), Vienna, Austria*, pages 479-484, August, 2011

[2] Dyck, A., Penners, R., and Licthter, H. 2015. Towards Definitions for Release Engineering and DevOps, in *Proc. of the 3rd International Workshop on Release Engineering, Florence, Italy*, pages 3-3, May, 2015

[3] Epstein, J., Matsumoto, S., and McGraw, G. 2006. Software Security and SOA: Danger, Will Robinson! in *IEEE Security & Privacy*, vol.4, no. 1, pages 80-83, January, 2006

[4] Feitelson, D., Frachtenburg, E., and Beck, K. 2013. Development and Deployment at Facebook, in *IEEE Internet Computing*, vol. 17, no. 4, pp. 8-17, July –August, 2013

[5] Ferrante, D. 2006. Software Licensing Models: What's Out There? in IT Professional, vol. 8, no. 6, November, 2006

[6] Humble, J., and Farley, D. 2011. Continuous Delivery, 1st Ed. Addison-Wesley, Boston, MA, 2011

[7] IEEE Standards Association. IEEE SA – 24765 – 2010 – Systems and Software Engineering – Vocabulary: 2010. https://standards.ieee.org/findstds/standard/24765-2010.html. Accessed: 2016-01-24

[8] Innovation. S. Regulatory Compliance Demystified: An Introduction to Compliance for Developers: 2006. Available: https://msdn.microsoft.com/en-us/library/aa480484.aspx. Accessed: 2016-01-24

[9] ISO/IEC/IEEE. ISO/IEC/IEEE 29119:2013 Software and Systems Engineering-Software Testing-Part 1: Concepts and Definitions: 2013. http://www.iso.org/iso/catalogue_detail.htm?csnumber=45142. Accessed: 2016-01-24

[10] Labs, P., and Revolution, IT. 2015 State of DevOps Report | Puppet Labs: 2015. Available: https://puppetlabs.com/sites/default/files/2015-state-of-devops-report.pdf. Accessed: 2016-01-24

[11] Likert, R. 1932. A Technique for the Measurement of Attitudes, in *Archives of Psychology*, vol. 22, no. 140, pages 5-55, June, 1932

[12] McGraw, G. 2006. Software Security: Building Security In, Addison-Wesley Professional, Cambridge, MA, 2006

[13] McGraw, G. 2008. Automated Code Review Tools for Security. *Computer*, vol.41, no.12, pp.108-111, December, 2008

[14] McGraw, G., Migues, S., and West J. BSIMM 6: Building Security in Maturity Model: 2015. https://www.bsimm.com/download/. Accessed: 2016-01-24

[15] Microsoft. Microsoft Secure Development Lifecycle Guidance: 2012. https://www.microsoft.com/en-us/download/details.aspx?id=29884. Accessed: 2016-01-24

[16] Moore, G. 2002. Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers, Revised Ed., Collins Business Essentials, New York City, NY, 2002

[17] Potter, B. and McGraw, G. 2004. Software Security Testing, in *IEEE Security & Privacy*, vol.2, no.5, pages 81-85, September, 2004

[18] Rahman, A., Helms, E., Williams, L., and Parnin, C. 2015. Synthesizing Continuous Deployment Practices Used in Software Development, in *Proceedings of the 13th Agile Conference (AGILE 2015), Washington D.C., USA*, pages 1-10, August, 2015

[19] Shostack, A. 2014. Threat Modeling: Designing for Security, John Wiley & Sons Inc., Indianapolis, IN, 2014

[20] Simpson, S. 2014. SAFECode Whitepaper: Fundamental Practices for Secure Software Development, in *ISSE 2014 Securing Electronic Business Processes*, vol. 1, no.1, pages 1-32, October, 2014

[21] Smeds, J., Nybom, K., and Porres, I. 2015. DevOps: A Definition and Perceived Adoption Impediments, in *Proceedings of 16th International Conference on Agile Processes in Software Engineering, and Extreme Programming, Helsinki, Finland*, pages 166-177, May, 2015

[22] Technologies, CA. DevOps: The Worst Kept Secret to Winning in the Application Economy: 2014. http://www.ca.com/us/~/media/Files/whitepapers/devops-the-worst-kept-secret-to-winning-in-the-application-economy.pdf. Accessed: 2016-01-24

[23] Turnbull, J. DevOps & Security: 2012. http://www.slideshare.net/jamtur01/security-loves-devops-devopsdays-austin-2012. Accessed: 2016-01-24

[24] Velasquez, N., Kim, G., Kersten, N., and Humble, J. 2014 State of DevOps Report: 2014. https://puppetlabs.com/sites/default/files/2014-state-of-devops-report.pdf. Accessed: 2016-01-24