BP: Security Concerns and Best Practices for Automation of Software Deployment Processes An Industrial Case Study

Vaishnavi Mohan
Deloitte Analytics Institute
Germany
vaimohan@deloitte.de

Lotfi ben Othmane
Iowa State University
Ames, USA
othmanel@iastate.edu

Andre Kres
IBM
Germany
kres@de.ibm.com

***Abstract*—SecDevOps is a paradigm for integrating the software development and operation processes considering security and compliance requirements. Organizations are reluctant to transform their development and operation processes to SecDevOps because of the expectation of incompatibility between security and DevOps. This paper reports about a study performed at IBM on transformation of five Business Intelligence (BI) projects to SecDevOps. The study revealed that main security concerns for the automation of the deployment process are: separation of duties, enforcement of access controls, manual security tests, audit, security guidelines, management of security issues, and participation of the security team. The majors recommended best practices for a transformation of current processes to SecDevOps are: good documentation and logging, strong collaboration and communication, automation of the process, and enforcement of separation of duties. Based on the study, we believe that separation of duties is the main aspect to be considered when planning to automate deployment processes. The results of the study are being used by IBM BI Unit and may be used by other organizations when planning to migrate to SecDevOps, especially for BI projects.**

I. Introduction

DevOps is a practice that aims at integrating software development and operation processes [1]. It is expected to reduce deployment cycles, improve the quality of software, and shorten the time to patch bugs. This helps to increase the frequency of deployments, which helps to service customers faster [2]. Organizations that practice DevOps can deploy software changes as fast as 500 times per day [3]. CA Technologies found that 1254 out of 1425 surveyed organizations (i.e., 88%) will adopt DevOps by 2020 [4].

Infrastructure as a Service (IaaS) providers, such as IBM Bluemix [5], support the design and operation of DevOps. Current market solutions focus on micro-service architectures, which favor independent/decoupled components [6], [7]. Established organizations usually have separate development and operation processes and use manual processes to transit the software from development to operations. Adopting DevOps for legacy software is challenging [8], given the difficulties in managing dependencies and aspects, such as information access policies [9].

IBM is among the companies currently working on adopting SecDevOps. The company has a set of internal BI applications to provide insights on their product sales. These are long-running, frequently changing applications hosted on legacy systems that need to comply with the quality and security requirements of the company. IBM is in the process of automating the development and deployment processes of the existing BI projects. Automation of the software deployment process is among the main DevOps best practices [10]. The organization is interested to gain insights on the security and compliance aspects that need to be considered when integrating and automating the development and operations processes. Rapidly deployed software changes are more likely to contain vulnerabilities if adequate measures are not considered [3].

This paper reports about a case study that we carried out at IBM to identify the security aspects related to the automation of the deployment process for BI projects. The study aims to address the questions:

1) What are the security concerns that impact the adoption of SecDevOps?
2) What are the best practices for SecDevOps?

This case study focuses on the transformation of development and deployment processes into SecDevOps at the BI Unit of the IBM CIO Europe organization. We selected five current projects for developing and deploying IBM-internal BI applications, listed in Table I. We interviewed nine project stakeholders. The interviews were then transcribed and coded to identify the main themes [11].

There are several empirical studies that aim to study SecDevOps practices in organizations, e.g., [3], [12]. These studies report on the generic opinions of project stakeholders. Our study provides a more objective view on the topic; it reports the opinions of projects stakeholders gained from working on a set of projects that need to be transformed or are transformed to the SecDevOps paradigm. The results of the study are being used by IBM in transforming their legacy BI projects to SecDevOps. They could be used by other organizations, as well.

The paper is organized as follows. First, we give an overview of the development and operation environment

TABLE I
Projects included in the Case Study

NO	Project	Functionality	Environments	Manual-Automated
P1	BMT, Business Metrics Tool	The project, a part of IBM Sales Analytics, develops and maintains an information management tool for IBM Technology Support Services. The information warehouse has information from orders to financing and customer satisfaction data.	Development, ITC, Production	Manual
P2	WW BPDM, World Wide Business Partner Data Mart	The project maintains comprehensive information about IBM's Business Partners. The data marts provide information from business partner contracts to business partner targets.	Development, ITC, Production	Manual
P3	GLIW, Global Logistics Information Warehouse	The project maintains IBM's Global Logistics data to provide analytics capabilities for smarter and more efficient supply chain.	Development, ITC, Production	Manual
P4	PCR Insight	A mobile application enabling IBM partner client's with information on business partner related activities. The application makes use of data from the Business Partner Data Mart (BPDM) Europe.	Development	Manual
P5	Cognos Reporting	The project involves the development of Cognos based reports for end users utilizing data from the data Warehouses and Datamarts.	Development, Production	Automated

and process ($$II). Then, we describe the research method ($$III). Next, we discuss the results of the study, namely the concerns related to automating the deployment processes ($$IV), the security impacts of moving to SecDevOps ($$V), and the best practices for SecDevOps ($$VI). Then, we discuss the threats to validity ($$VII) of the study. We discuss related work ($$VIII) and conclude the paper ($$IX) afterwords.

II. Overview of the current Development and Production Environment and Process

A description of the general architecture of the applications and development and deployment processes follow.

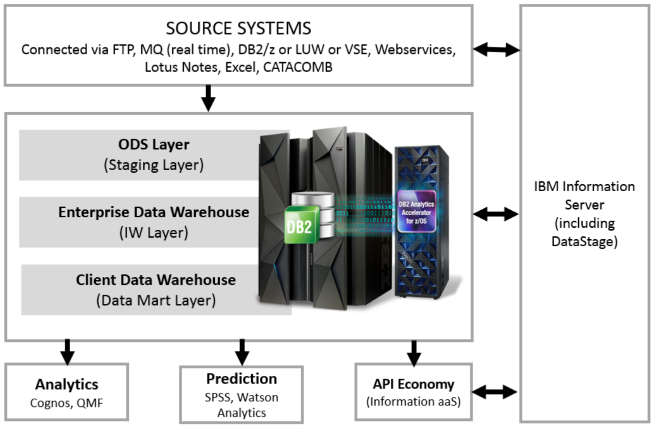
SOURCE SYSTEMS
Connected via FTP, MQ (real time), DB2/z or LUW or VSE, Webservices, Lotus Notes, Excel, CATACOMB
ODS Layer
(Staging Layer)
Enterprise Data Warehouse
(IW Layer)
Client Data Warehouse
(Data Mart Layer)
DB2
DB2 Analytics Accelerator for z/OS
IBM Information Server (including DataStage)
Analytics
Cognos, QMF
Prediction
SPSS, Watson Analytics
API Economy
(Information aaS)

Fig. 1. Technical architecture used by the projects considered in the Case Study.

A. High-level Architecture of the Business Intelligence Applications

Figure 1 visualizes the high-level architecture of the BI applications considered in this study. The applications realize the data lake architecture and are hosted on IBM Z-System servers [13]. In this architecture, several data source systems are connected to the data warehouse using interfacing methods. The Extract-Transform-Load processes are performed using File Transfer Protocol (FTP); IBM MQ, a secure and reliable messaging middleware; IBM DB2, Web services, IBM Lotus Notes; Microsoft Excel; CATACOMB, and IBM InfoSphere DataStage. These technologies play a key role in populating the data lake with cleansed and standardized data. The database management system for the data warehouses residing on IBM Z-Systems is IBM's DB2 for Z/OS and DB2 Analytics Accelerator. Analytics is supported by IBM Cognos Analytics; adhoc querying and reporting are done using IBM DB2 QMF for Workstation. Watson analytics and IBM SPSS predictive analytics provide statistical analysis and reporting, predictive modeling, data mining, and big data analytics. The data in the data warehouse are made available to external businesses and teams through APIs.

B. Development and Deployment Processes for the Selected Projects

The development of BI applications is iterative. Figure 2 shows the development and deployment processes used by the five projects. The projects use three execution environments: Development, Integration Test Center (ITC), and Production. Table I specifies the environments used by each of the projects.

Every application release is assigned a priority based on the priority of the requirements it handles. Emergency fixes have the highest priority and are developed and deployed usually within a day. The deployment of normal priority releases usually takes at least a week.

The installation package and the release letter, are the two main deployment artifacts in the projects. The installation package contains the necessary jobs to install the release which includes the code for the requirements and fixes. The release letter specifies the instructions for the installation of the application in the target environment, including a list of files included in the installation package and instructions on file handling. It also describes storage space requirements in the case of newly developed tables.

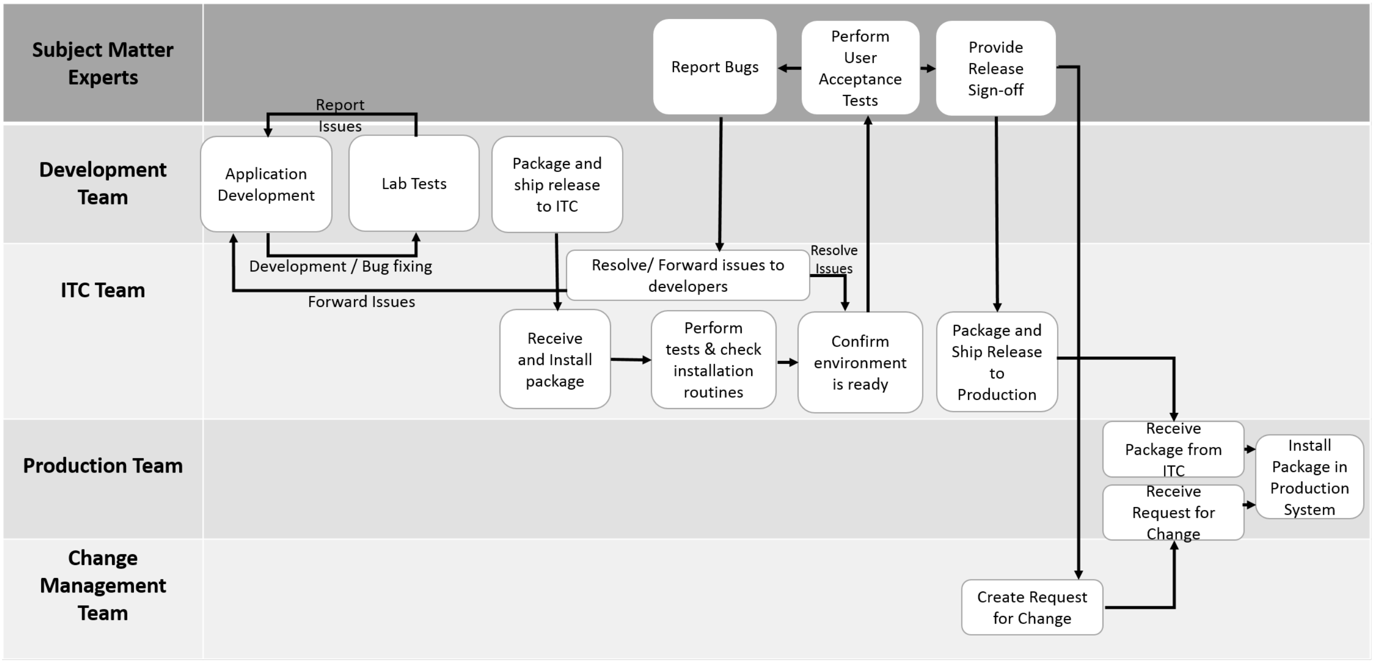
Subject Matter Experts
Development Team
ITC Team
Production Team
Change Management Team
Report Bugs
Perform User Acceptance Tests
Provide Release Sign-off
Report Issues
Application Development
Lab Tests
Package and ship release to ITC
Development / Bug fixing
Forward Issues
Resolve/ Forward issues to developers
Resolve Issues
Receive and Install package
Perform tests & check installation routines
Confirm environment is ready
Package and Ship Release to Production
Receive Package from ITC
Receive Request for Change
Install Package in Production System
Create Request for Change

Fig. 2. Deployment Process.

The projects follow two deployments processes: manual deployment process and automated deployment process. The description of both processes follows.

Manual Deployment Process. Projects P1, P2, P3 and P4 are deployed manually. Figure 2 visualizes the deployment process of Projects P1, P2 and P3. The process is entirely manual, and involves the exchange of work packages through FTP and XMIT [14] or email. The communication takes place mostly over emails and sometimes, through the IBM internal chat client.

Once application components are developed, developers perform unit tests (aka lab tests). Sometimes, the Subject Matter Experts (SMEs) get involved in the lab tests. They verify the conformance of the given application to its requirements and report discovered anomalies to the developers who fix their code accordingly. Once unit tests are approved, the development team packages all components necessary for the release and ships the package to the ITC team.

The ITC team installs the package into the ITC environment and performs initial installation routine checks. Once the environment is ready with the installed application, the SME is notified. The SME performs the tests as an end-user, and provides a sign-off. The sign-off is directed to various teams, including the ITC, Production and the Change Management Teams. The ITC team repackages the installation artifacts and ships them to the Production Team when they receive the sign-off from the SME team. Then, the Production Team proceeds with installing the application into the production environment when they receive the change request from the Change Management Team.

Project P4 is a mobile application that uses a single server. It uses separated environments for development, testing and production. The project is developed and operated by the Development team. The application, on development, and testing in the development environment, is directly manually deployed into production by the developers themselves.

Automated Deployment Process. This deployment process uses Motio tool [15] to automate the deployment of developed applications. Project P5 uses this process. Project P5 deals with report creation using IBM Cognos BI. The developed reports in the Development environment are automatically deployed to Production and tested against production data. An ITC environment exists for this project, but it is rarely used, since testing on production data is more effective.

III. Research Method

The goal of the study is to identify the security aspects that impact the transformation of a manual deployment process into an automated process. Data collection was performed at IBM through qualitative semi-structured interviews with selected stakeholders of the five studied projects. The description of these steps follow.

A. Definition of the Interview Protocol

The interview protocol was constructed using Turner's guidelines [16]. First, a set of interview questions were formulated based on the research goals and information from informal discussions with the project leads. The questionnaire was tested by trial runs with team members for the selected projects. The questions were revised, based on the feedback, until interviewees provided answers relevant

TABLE II
List of the interviewees.

Sub ID	Role	Team	Automated / Manual	Projects
S1	Chief Architect	Overall	Both	
S2	Program Manager	Overall	Both	
S3	Solution Lead	Development	Manual	
S4	Project Lead	Development	Automated	
S5	Business Analyst/IT Specialist	Development	Manual	
S6	IT Specialist	Development	Manual	
S7	Subject Matter Expert/Information Warehouse Administrator	Security	Manual	
S8	IT Specialist	ITC	Manual	
S9	Mainframe Operations Analyst	Operations	Manual	

to the research goal. The open-ended questions enabled participants to provide detailed responses which helped to gain more insights.

B. Selection of the Participants

We selected for the interviews nine participants. Table II lists the selected participants along with their roles and the teams that they belong to. Project leaders involved in the overall development and maintenance process of the studied projects are marked Overall, the developers are marked Development, the specialists at the integration test center responsible for maintaining the test environment are marked ITC, the production environment maintainers are marked Operations and the security specialists are marked Security. The interviewees are selected to openly and honestly share information to support the research [17]. They cover the different aspects of SecDevOps.

C. Execution of the Interviews

The interviews were a mix of face-to-face and telephonic interviews. Due to the geographic distribution of the interviewees we conducted three interviews in-person and six through phone. Each interview lasted about an hour. The participants were informed about the recording of the interviews and were briefed about the research goal before the start of the interview. The interviews were supported by the questions of the interview protocol as a guideline. Secondary questions were often used in the interviews to get further insights. Questions like *'Could you provide an Example?'* and *'Could you tell me more about...'* helped gathering comprehensive information.

D. Transcribing Interviews

We transcribed all the interviews and used the tool oTranscribe[1] to help in the process.

E. Interview Coding

We used the thematic analysis method for the interview coding [11]. Interview Coding uses the interview transcripts as input and outputs codes that identify all the aspects mentioned during the interviews. A code is a word or short phrase identifying the essence of a portion of language-based or visual data. At the end of this step we assigned codes to each of the nine interview transcripts.

TABLE III
The themes identified from interview transcripts.

NO	Themes category	Themes
1	Security concerns for the automation of the deployment process	Separation of duties, enforcement of access control policies, manual security tests, audit, security guidelines, management of security issues, participation of the security team
2	Security impacts of moving to SecDevOps	Application security and separation of duties
3	Best practices for the transformation to SecDevOps	Good documentation and logging, strong collaboration and communication, automation of the process, and enforcement of separation of duties

F. Data Extraction and Classification

The next step was to group similar codes together to form themes. A theme generalizes a set of codes belonging to a given concept as a theme. The process of assigning themes to codes was done for each transcript. Next, we gave the transcripts and codes to the interviewees to ensure that the understanding and interpretation are correct. Then, we merged the codes that were semantically similar across transcripts. Table III lists the themes and associated categories.

G. Analysis of the Results

We classified the codes into three categories: the security concerns for the automation of the deployment process, the security impacts of moving to SecDevOps, and the best practices for the transformation to SecDevOps. We use the term "process automation" when the aspects that we discuss are process related and we use the term "SecDevOps" when the aspects that we discuss could be generalized to SecDevOps paradigm.

IV. The security concerns for the automation of the deployment process

This section discusses the findings with respect to the security state of the projects under the case study. Security is preserved in the projects mainly through separation of duties, access control, performing manual security tests, conducting periodic audits, security guidelines, security issues and their mitigation, and participation of the security team in the DevOps process.

A. Separation of Duties

Separation of Duties ensures that developers cannot change code in the production environment. That is, the teams performing the duties of development, testing and operations are independent. In the studied applications,

[1] otranscribe.com/

access restrictions are applied using Resource Access Control Facility (RACF), a security manager providing access control for z/OS operating systems.

The importance of separation of duties was captured clearly through an example depicted by a participant who works on project P4. They said: *"we are running on one server that is used for all the processes – development, testing and production. Of course it is not good at all. It's really dangerous, because if the developer presses the wrong button, it kills the application that is in production. This is absolutely not good. We had the problem that the developer accidentally killed the demo mode. This has happened twice, and that's when this became really obvious that we needed a separated three environments system"*.

B. Enforcement of Access Control Policies

The main security concept of the Information Warehouses (IWs) is to control usage of data by the users using data views. The users of the data warehouse access the data through views and do not have direct accesses to the tables. Security tables maintain access control information for every user of the IW. The SMEs have access to the tables. During table creation, access is granted to certain RACF user groups [18]. If a user belongs to one of the permitted RACF groups, then, the user is allowed to access the data via the user views. The user IDs and RACF groups are created following the general deployment process, similar to other requirements.

C. Manual Security Tests

The requirements related to access management are identified and documented in the requirements specification during the requirements analysis process. The security objects need to be tested before shipment into the Production environment once the security requirements are developed and deployed on the ITC environment. Two of the participants mentioned that only positive testing for the security objects (e.g., a user ID is granted access to a view) is carried out. One participant mentioned that such security testing is sometimes forgotten. Two participants mentioned the extensive logging of user accesses to the warehouse data for security purposes. One participant also reported that there is a yearly re-validation of user IDs to ensure that all the current users of the applications have valid justifications.

D. Audit

The IW projects undergo Application Systems Control and Auditability (ASCA) – the IBM internal audit – approximately every six months. The audit intends to ensure that the systems are robust and reliable. The IW projects integrated the guidelines necessary for a successful ASCA audit into their standard processes. The ASCA auditors review the software against different review elements and grant certifications for software that comply with all the elements. The certification aims to ensure the integrity of the data and the accuracy of data processing. In addition, it helps preventing and detecting frauds. The elements reviewed during the ASCA audit include: access management and security evaluation, separation of duties evaluation, application system management controls, data security and privacy, and testing. During the ASCA audit, the security of the IW is reviewed, where a presence of every single user of the IWs needs to be justified. Inactive users are to be removed.

E. Security Guidelines

The security guidelines detail the procedure to grant accesses to new IW users. The new user submits their access request via the `One Team` access management tool, an IBM internal tool used for access request management. Once the access request is submitted, it follows a two-step approval process, so the user may be granted access if entitled to. This two-level approval ensures that only the relevant users get access to the requested data.

F. Security Issues and Mitigation

All the participants reported that there were no major security issues with their development and deployment processes. Two participants illustrated two security issues that were identified and immediately rectified. One participant said *"... We had sometimes mild warnings of mild findings. We had some inactive user IDs that were not used anymore."* They refer to inactive user IDs that were detected when analyzing the log files. Another participant reported a security issue related to the user view definition. They stated: *"...if we have a table, that has one million rows, then, the user view usually should also show you one million rows. Sometimes there was a problem in the view definition, and at the end it turned out that the view returned five million rows."* The participant claimed that such testing of user views is usually forgotten despite its importance. The mitigation process involves revoking the access of invalid and unidentified users, or developing a security fix.

G. Participation of Security Team

The security team consists of two IWs administrators who are responsible for the security tables setup and the management of access requests. The security team is involved also in monitoring usage statistics of the data marts in the project environment and reporting and resolving abnormal usage activities. In addition, it is involved in the requirements analysis phase and sometimes involved in the testing phase of the ITC environment.

H. Discussion

In this study we used four projects that use manual deployment processes and one project (P5) that uses automated deployment process–see Table I. Project $P5$ is about developing business reports that use existing data. The BI unit does not consider the project as reference for SecDevOps because the best practices guidelines at IBM do not require that projects for reports-based software to (1) enforce separation of duties, (2) include security features besides authentication, and (3) undergo audit. Thus, we do not identify the security concerns and best practices by comparing project $P5$ with the other projects.

V. security impacts of moving to SecDevOps

The two major security impacts for moving to SecDevOps, as reported by the participants, are application security and preservation of separation of duties. We discuss both impacts in the following.

A. Application Security

Five out of nine participants suggested that the adoption of a DevOps process with automated deployment and testing would make the system more secure. All 5 participants mentioned that the automated deployment would result in faster deployments, which in turn would reduce the duration to deploy security objects and security fixes. Two participants suggested that automated testing for access control would make it more robust. One participant mentioned that manual testing is limited to positive testing, and that this could be improved in the presence of automation. Another participant mentioned that, the testing of user view definitions should be automated because it is usually forgotten during manual deployments.

B. Separation of Duties

DevOps promotes the close collaboration of the teams involved in the development and deployment processes. Five of the participants appealed that separation of duties should be preserved even in an automated process. The major suggested requirements to preserve are:

1) The developer should not design the test cases since the test cases function as gatekeepers for the other environments.
2) A change request needs to be initiated and approved by stakeholders before deployment into production.
3) There should be separate access rights for each of the three environments: Development access, ITC access and Production access. A developer shall not have right access to both the Test and Production environments.

VI. Best Practices for a Transformation to SecDevOps

The participants suggested a set of best practices to guide the transformation towards SecDevOps, which we discuss in the following.

A. Good Documentation and Logging

The automated deployment and testing outcomes need to be documented for easier detection and elimination of anomalies in the process and to serve as evidence for audits. The participants proposed the following practices for effective documentation:

1) Tools like Rational Team Concert and Rational Quality Manager can be used for storing the logs of deployment and testing activities respectively.
2) Security logging of user accesses to data warehouses are essential and should not be disrupted by the automation process.
3) Linked meta-data repositories should be created and maintained to ease retrieval of information relevant to required components, for example, to retrieve the last jobs that manipulated a given table.
4) Document repositories should log the details of process owners at each step of the deployment process, for example, to indicate who approved the last tests set.
5) Documentation should not be shared through a medium which cannot be traced. For example, data should not be shared through only the chat clients. All required information need to be stored in a central repository that can be accessed by all the stakeholders of the process.

B. Strong Collaboration and Communication

The participants suggest that collaboration should be disconnected from the process. For example, the steps of deployment should be recorded automatically to a repository accessible to stakeholders on demand. The participants also recommend that the exchange of unnecessary emails at each step of the process should be reduced since this could possibly save a lot of communication effort. They suggest that the environment owner should be automatically informed in the case of any issues. They also propose that the stakeholders should be automatically notified of successful installations and testing.

C. Automation of Processes

The participants suggest that the automated deployment process should be as follows. The developers invoke the task installation to the ITC environment when they complete the development of their applications. On successful completion, the test cases should be run automatically. A change request for installation could be approved if the results of the tests are conclusive. Then, the package should be shipped to the production environment, where it is installed automatically.

The participants suggested improvements to each step of the existing deployment process to ensure effective automation. First, the dependencies of the functionalities that will need to be deployed need to be identified during the requirement specification step. This is critical to prepare for a safe deployment of the software.

Second, the deployment package should contain the developed artifacts and all required information for the deployment. For example, the developers need to provide calculations for storage space requirements of the table along with table definitions. In addition, the installation packages sent to the production environment need to include a fallback plan to ensure recovery in case of any irregularities during installation. The installation procedures should also define the necessary jobs and batch processes and their scheduling.

Third, the participants claim that the user tests cannot be completely automated. They suggest that the automated process should support two options: automatic invocation of automated tests after successful installation, and notification to the tester that the environment is ready for the manual tests. They also suggest that security tests should ensure that user views retrieved the expected results. In addition, they suggest that installation tests should ensure that the deployed package would not disrupt the existing system or other associated systems. The installation tests should ensure that all the packages that were sent as a part of the deployment package are received and installed successfully. Post-installation tests need to verify that the version of package in the current environment is the same as the expected.

Fourth, service monitoring should be used in the deployment process. It would allow to analyze and report the health of the services being used by the environment, which would ease the debugging process in case any issues in the applications. For example, the feature would enable detecting whether a given problem is caused by application code or associated services.

D. *Enforcement of Separation of Duties*

The participants suggested three separation of duties practices. First, test cases should be designed by a team independent of the developers, since the test cases act as gatekeepers to the other environments. Second, A manual kick-off before installation to production is required as every change to the production environment needs to be driven by a change request. Third, three different sets of access rights should be defined for the developers, testers and users accessing the production environment. This would ensure that people with development access cannot access the production data.

In addition, the participants propose that the developers should define installation procedures along with developing the packages for installation; the ITC team should be involved in defining automated quality steps or gates that need to be successful before the automatic installation of the release happens to the new environment; the SMEs responsible for testing should design automated test cases; and the operations team would be required to fix issues when they arise during installation of the applications to the production environment.

VII. Threats to validity

We discuss now the limitations of the study according to Wohlin's et al. validity taxonomy [19], [20].

Construct validity. To ensure valid relation between the performed study and the goal of the study, we first performed a literature review. Next, we designed interview protocol and tested it with three experts. In each iteration, the questions and the answers were discussed with peers and the questions were adjusted to be more clear and efficient in getting the required information. In addition, we collected the information from nine interviewees who have different roles and are located in different cities. Moreover, the interviewees worked on five projects, which gives confidence in the stability of the collected data. The main limitation of the study is that we only collected the data by interviewing domain experts and we did not cross-validate the results with other source of information.

Internal validity. To ensure causal relationship between the study and its results we told the interviewees at the opening of the interviews the goal of the study. We have participants from five projects covering development, operation, testing, and security aspects. Nevertheless, we do not claim saturation.

Conclusion validity. To ensure ability to draw correct conclusions from the results of the study, we sent each interviewee a short report about the codes that we extracted from the interview that we conducted with them to ensure that we have a common understanding; that is, we performed member checking [21]. In addition, the main author checked the extracted codes and identified themes with the two other researchers to reduce the subjectivity of the results. Nevertheless, we acknowledge that the perspective of the researchers may have impacted the data extraction and interpretation.

External validity. The study was conducted for one application domain, namely BI, at one company, namely IBM Europe. We observe that several of the identified aspects are strongly related to the application domain or application size, such as the control of access to data using the view, or strongly related to IBM such as the use of ASCA audit. The results of the study could be used for transformation of BI projects to SecDevOps, especially within IBM, without high risk.

VIII. Related work

Farroha et al. [2] describe the need for DevOps. They note, however, that fast deployments usually compromise the security and compliance of software. They propose a framework to incorporate and maintain security in systems providing continually updated services. Storms [22] shared the position of the importance of security in DevOps. He advocated that security could be preserved in a DevOps environment, if the right processes and techniques are in place. Vries [23] also promotes the importance of setting security goals during requirements engineering. In addition, they describe in [23] how security processes, like security tests and security scans can make use of DevOps. Schneider [12], introduced the Secure DevOps Maturity Model (SDOMM) which guides applications to attain the target security levels through integrating open source tools in the continuous integration process.

Mohan et al. [9] surveyed the literature from academia and industry to identify the main aspects of SecDevOps. The main aspects that they found are: definition, security best practices, compliance, process automation, tools for SecDevOps, software configuration, team collaboration, availability of activity data and data secrecy.

Rahman and Williams [3] analyzed a selected set of Internet artifacts and surveyed representatives of nine organizations that are using DevOps. They observed that the majority of the practitioners have expressed the potential of common DevOps activities to improve the security of a system. In addition, they observed that organizations that integrate security into DevOps support collaboration between the security team and the development and operations teams.

We note that according to Raynaud [24], applying DevOps principles to security is easy. In addition, it helps improve the relationship between security and development teams and ultimately the success of a product. Balalaie et al. [6] reported on their experiences and lessons learned from adopting DevOps and how this facilitated a smooth migration of security issues.

IX. Conclusion

Automation of deployment process is critical to the transformation of development and operation processes to SecDevOps. This paper reports about the security aspects related to automating the deployment process. The interviewees reported seven concerns to be considered when planning to automate a deployment process. Two of the concerns, management of access control policies and audit, are related to BI domain, which is the application domain of this study. The other five concerns are generic and could apply to contexts other than BI projects at IBM.

The study identified four best practices for a transformation to SecDevOps: documentation and logging, strong collaboration and communication, automation of the deployment process, and enforcement of separation of duties. We find that these recommendations are common sense.

We observe that separation of duties was mentioned in the three aspects of the study: the security concerns for the automation of the deployment process, the security impacts of moving to SecDevOps, and the best practices for the transformation to SecDevOps. We believe that separation of duties is the main aspects to be considered when planning a transformation of development and operation processes to SecDevOps.

References

[1] R. Jabbari, N. bin Ali, K. Petersen, and B. Tanveer, "What is devops?: A systematic mapping study on definitions and practices," in *Proc. of the Scientific Workshop Proceedings of XP2016*, (Edinburgh, UK), 2016.

[2] B. Farroha and D. Farroha, "A framework for managing mission needs, compliance, and trust in the DevOps environment," in *Proc. of the 2014 IEEE Military Communications Conference*, (Baltimore, MD, USA), pp. 288–293, Oct 2014.

[3] A. A. Rahman and L. Williams, "Software security in DevOps: Synthesizing practitioners' perceptions and practices," in *Proc. of the 2016 International Workshop on Continuous Software Evolution and Delivery (CSED)*, (Austin, TX, USA), pp. 70–76, May 2016.

[4] CA Technologies, "Devops: The worst-kept secret to winning in the application economy." Accessed: 2016-11-25.

[5] IBM, "Cloud platform: Cloud infrastructure - IBM bluemix." accessed on October 2017.

[6] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices architecture enables devops: Migration to a cloud-native architecture," *IEEE Software*, vol. 33, pp. 42–52, May 2016.

[7] L. Bass, I. Weber, and L. Zhu, *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, 2015.

[8] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops," *IEEE Software*, vol. 33, no. 3, pp. 94–100, 2016.

[9] V. Mohan and L. Ben Othmane, "Secdevops:is it a marketing buzzword? mapping research on security in devops," in *Proc. of the 11th International Conference on Availability, Reliability and Security (ARES)*, (Salzburg, Austria), pp. 542–547, Sep 2016.

[10] M. Artac, T. Borovssak, E. D. Nitto, M. Guerriero, and D. A. Tamburri, "Devops: Introducing infrastructure-as-code," in *Proc. IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pp. 497–498, May 2017.

[11] J. Saldaña, *The Coding Manual for Qualitative Researchers*. Sage, 2015.

[12] C. Schneider, "Security DevOps - staying secure in agile projects," in *OWASP AppSec Europe*, (Amsterdam, Netherlands), 2015.

[13] IBM, "IBM system Z." http://www-03.ibm.com/systems/z/. Accessed: 2016-11-28.

[14] IBM, "Transfering load modules using XMIT and FTP." https://www.ibm.com/developerworks/community/blogs/cicsabel/entry/transfering_load_modules_between_mainframes_using_xmit_and_ftp20?lang=en. Accessed: 2016-11-28.

[15] Motio, "Motio - automated deployment for ibm cognos." https://www.motio.com/products.do. Accessed: 2016-11-28.

[16] D. Turner, "Qualitative interview design: A practical guide for novice investigators," *The Qualitative Report*, vol. 15, no. 3, pp. 754–760, 2010.

[17] J. W. Creswell and V. Clark, "Designing and conducting mixed methods research," 2007.

[18] Wikipedia, "Resource access control facility." https://en.wikipedia.org/wiki/Resource_Access_Control_Facility. Accessed: 2016-11-28.

[19] C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in Software Engineering*. Berlin Heidelberg: Springer-Verlag, 2012.

[20] D. S. Cruzes and L. ben Othmane, *Empirical Research for Software Security: Foundations and Experience*, ch. Threats to Validity in Software Security Empirical Research, pp. 275–300. Taylor & Francis Group, LLC, 2017.

[21] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on Software Engineering*, vol. 25, pp. 557–572, Jul 1999.

[22] A. Storms, "How security can be the next force multiplier in devops," in *RSAConference*, (San Francisco, USA), 2015.

[23] S. de Vries, "Continuous security testing in a DevOps world," in *OWASP AppSec Europe*, (Cambridge, UK), 2014.

[24] K. Carter, "Francois Raynaud on DevSecOps," *IEEE Software*, vol. 34, no. 5, pp. 93–96, 2017.