Advances in Science, Technology and Engineering Systems Journal Vol. 6, No. 1, 1157-1163 (2021)

www.astesj.com

Special Issue on Multidisciplinary Sciences and Engineering

ASTESJ

ISSN: 2415-6698

An Operational Responsibility and Task Monitoring Method: A Data Breach Case Study

Saliha Assoul[1], Anass Rabii[*, 2], Ounsa Roudiès[2]

[1]*ENSMR, Siweb, E3S, Mohammed V University, Rabat, 10010, Morocco*

[2]*EMI, Siweb, E3S, Mohammed V University, Rabat, 10010, Morocco*

ARTICLE INFO

Article history:
Received: 25 December, 2020
Accepted: 09 February, 2021
Online: 25 February, 2021

Keywords:
Information Security
Security Breach
Supervision
Monitoring Method
SysML
Capital One

ABSTRACT

As a result of digitalization, services become highly dependent on information systems thus increasing the criticality of security management. However, with system complexity and the involvement of more human resources, it becomes more arduous to monitor and track tasks and responsibilities. This creates a lack of visibility hindering decision making. To support operational monitoring, we propose a method composed of i) a core of security concepts from International Standard Organization (ISO) standards ii) a graphical modeling language iii) a guiding process and iv) a tool that provides verification through formal Object Constraint Language (OCL) queries. Applying this method to the case of the Capital One data breach showcases incident prevention through task supervision. The resulting work product is a formal comprehensive map of assets, actors, tasks and responsibilities. The SysML formalism allows different actors to extract information from the map using OCL queries. This allows for regular task and responsibility verification thus closing any window of attack possible.

1. Introduction

With the advent of digitalization era, the widespread dependence on information systems (IS) rose as well. This new age was kick started by technological developments especially in IoT, Big Data and Cloud, paving the way for newer services and spreading IS use to more diverse fields (Industry-science research alliance). However, this dependence translated into an increase in criticality levels of assets and tasks. Fields such as health, energy, or e-Government, which were already dealing with critical information or physical assets, are facing greater challenges in security management. As a response, practices of security management had been thoroughly addressed by corporations, standardizing organizations and academia. The quest for security approaches and methods to apply systematically to specific situations fueled the creation of best practices, standards and maturity models. The popular ISO 27000 standard series stood out for the creation and management of Information Security Management Systems (ISMS) (ISO 27000). ISO provides baseline requirements in ISO 27001, best practices in ISO 27002, security risk management in ISO 27005 or even specialized recommendation for network security in ISO 27033. Additionally, a plethora of security maturity models were created between 2007 and 2018 to support continuous improvement of security in diverse fields. However, our systematic review [1] of the literature showed that even if many standards and approaches are available, implementation remains problematic. The lack of implementation feedback further makes the study of a real case such as the Capital One Financial Corporation breach [2] more appealing to discuss. We postulate that this issue is due to two major aspects: system complexity and human behavior.

Dealing with increased complexity created between interacting systems, we find it necessary to adhere to a high level of formalism to ensure rigor and precision to tackle one of the four dimensions of human factors in information security: responsibility [3]. Taking root in the system engineering paradigm, our solution is based on the standard system formalization language SysML. We propose a scalable formal representation of security responsibilities in accordance with complex system composition that supports traceability and management. This formalism provided by extended SysML profiles, enables the use of formal verification languages such as OCL to support supervision. We also provide a guiding procedure based on continuous improvement for iterative applications as well as a supporting tool. This method would provide security managers with proper knowledge to track fundamental causes, such as asset vulnerabilities, unaccomplished tasks and accountabilities. In our previous studies, we described

[*]Corresponding Author: Anass Rabii, anassrabii@gmail.com

https://dx.doi.org/10.25046/aj0601130

our generic framework dedicated to information security management, extracted security concepts for SysML profiles [4] as well their application in security requirement management [5]. In this paper, we examine the responsibility issue and illustrate it using a case study.

We present and discuss the case of a data breach that occurred in Capital One in order to analyze the consequences of an informal specification of responsibilities and therefore prove the relevance of the problem we tackle. Next, we apply our method to the case to highlight its advantages. Through the Capital One case study we will address the following research questions:

- RQ1: How did Capital One track the chain of responsibility for their data breach?
- RQ2: How did Capital One security managers verify the implementation of security controls?
- RQ3: How to model tasks and responsibilities in a multi-scale information system?
- RQ4: How to formalize operational monitoring in a multi-scale information system?

In the following sections, we detail the context of our study highlighting the problem that our method aims to solve and how it fits within the existing body of work. Next, we present the components of our proposed method. Lastly, we present the Capital One Case followed by the application of our solution to highlight its applicability and its ability to prevent similar cases.

2. Related Works & Problematic

Information system complexity introduced heterogeneous components with a high connectivity level. In the complex system paradigm [6,7], systems can be decomposed into smaller autonomous collaborating sub-systems. This entails that security breaches and by consequence security management become pervasive. This means that measures taken to address security matters must be propagated to other collaborating systems. Similarly, this complexity makes responsibility harder to pinpoint and necessary tasks less apparent. On the other hand, the transition from theoretical and generic standard recommendations or requirements to practical application by human resources reveals a substantial gap. In 1996, including human behavior was considered as a novel conceptual stance within security management [8]. Since then, both academia and standardizing organizations have provided a plethora of references for security management that take human behavior as an important variable.

On one hand, standardizing organizations such as NIST and ISO provide thorough and well recognized references like the NIST Cybersecurity Framework, ISO 27001 and ISO 21827. These standards have either stood the test of time or have been periodically updated. We consider these standards as an important body of knowledge and procedures for our studies. We position ourselves as an extension of these standards providing the method to guide implementation that is out of their scope. On the other hand, academia has also produced countless information security maturity models such as SOASMM for SOA architecture [9], MMISS-SME [10] for small and medium enterprises, CCSMM for American governmental entities [11] or ISMM-PCI [12] for the payment card industry. Through our systematic literature review [1], we became aware of the lack of implementation and validation results for academic security maturity models. We also perceived that the academic shift towards specialization is also related to the implementation issue, hence why our solution intervenes at the implementation phase.

This implementation issue was due to the impact of human behavior on information security as asserted by both academia and standardizing organizations. According to research [13], human resources are considered the weakest link in securing information systems. In studying the impact of habits in following security policy [14], the authors highlight the effect of individual beliefs, thoughts, actions, attitudes, awareness, and training among others on policy compliance. ISO 27001 and 21827 standards also emphasize the importance of awareness building and skill development for all interested parties. In addition, as system complexity also entails more stakeholders, responsibilities become pervasive and contain multiple levels of accountability. In this context, the problem, firstly, is that actors have difficulties proficiently managing their responsibilities within their own scope of action. While the standard recommendations are clear, organizations have difficulties down-scaling them into actionable tasks and responsibilities [15]. Secondly, managers don’t have a multi-scale visibility over the system to insure monitoring adapted to scalability. In fact, we currently rely on individual manager capability for tracking tasks using natural language management tools and renowned methods such as responsibility assignment matrices [16].

As the Design Science [17] method entails, the next step is the production of artifacts to put this knowledge to use. Our method aims to bridge this gap by addressing operational monitoring for security management with a scalable vision, focusing on human behavior. In their study of non-malicious security violations, the authors [18] clear up that while employees are goal centered, job performance is the end goal rather than security. Sharing this view, we perceive that providing managers with the adequate tool to operate is the way forward. Also, studies on the social impact on information security address the influence of external systems in their culture models [19] demonstrating the need for the multi-scale visibility our solution provides. Finally, formalizing human involvement in applying security processes as well as supporting diversity of security approaches through genericity are essential. In order to achieve this, we rely on the prevalent system modeling language SysML. As a matter of fact, several studies use SysML as their basis for security risk assessment [20], secure system design [21] for model transformations [22]. We share the same system engineering vision for security management as these studies to address a different aspect of security that is operational management. Other modeling languages such as KAOS [23] offer the possibility for responsibility modeling but lacks the formalism necessary for supervision. Whereas, our solution enables a formal and complete representation of security related information. As a result, it will help test and verification using OCL, allowing for day to day monitoring, the cornerstone of Agile's [24] success in software engineering.

3. The operational responsibility and task monitoring method

As information systems' complexity blurs the lines of responsibility, we provide a generic method for responsibility, task and vulnerability traceability. By definition, a method is composed of a set of concepts, a language, a procedure and a tool to support it all. In the following sub-sections, we will detail each component as well as their provenance.

3.1. Core concept

In order to populate our modeling language, we need to introduce information security concepts. We can rely on existing references such as standards or prominent security maturity models in order to extract these concepts. In fact, we analyzed and compared the main security concepts used in the security maturity models we found through our systematic literature review [1]. We have found that these models are highly connected to the ISO 27001 and ISO 27002 standards [4]. Seeing that the ISO standards are aligned, we compared the security concepts used in ISO 21827 to prominent academic and governmental security maturity models. We concluded that the ISO standards are comprehensive sources for information security concepts. The concepts that we put to use in this monitoring method are listed in Table 1.

Table 1: Core of Information security concept list

Concept Name	Semantic Meaning	Example
System	Smart building, Energy grid, Network	Smart building, Energy grid, Network
Asset	Anything of value to the organization	Software, Building, Server, Information, Process
Human Resource	An actor implicated with the system	Client, Chief of security, System Owner
Task	A set of actions assigned to a human resource that must be executed	Log data analysis, Configuration update
Responsibility	The obligation to oversee, take care of a specific asset	Security responsibility, Maintenance responsibility, Access responsibility
Vulnerability	The state of exposure to the possibility of being attacked, damaged or tampered with	ragility, Open port, Plain text storage, Weak password

3.2. Language

In order to provide a graphical modeling language tailored to our concepts, we chose to extend SysML, the standard system modeling language. It supports specification, analysis, modeling, verification and validation of complex systems. It is defined as an extension of the well-known UML language used in software engineering. Another upside for choosing UML is the Object Constraint Language (OCL). OCL is a formal textual language that allows constraint and object query expressions. It is a key point to the testing needed for monitoring. In addition, we can extend the SysML basic concepts by adding our security core concepts by the mean of profiles. This leads to define a new block diagram that includes security concepts. It allows us to define newer associations between these concepts for the necessary interactions. This ensures that we profit fully from the richness of the predefined SysML block diagram concepts. Our profile consists of the security concepts in Table 1 coupled with five new associations added to define new relationships between them. In Table 2 we specify the composition of our profile as well as the properties required for traceability.

Table 2: Security Responsibility Profile Elements

Concept Name	Relationship	Properties
System	Systems contain one or many Assets. Systems can interact with one another.	Owner Purpose
Interact With	Association linking two systems	Influence StartDate EndDate
Asset	Assets collaborate with one another. Assets Belong to one or many systems. Assets have one or many responsibles. Assets have one or many vulnerabilities.	ID Description IsActive LastActiveDate
Belong To	Association between assets and systems	Criticality level
Collaborate With	Association between assets	Nature Description CollabDuration CollabStartDate
Vulnerability	A Vulnerability belongs to one or many assets	Existence likelihood Description Recommendation
Human Resource	A human resource can be responsible for one or many assets. A human resource is a specialization of an asset. A human resource can execute one or many tasks.	Organizational unit
Responsible for	Association linking a human resource to one or many assets.	RespDuration RespStart LastResp RespNature

Execute	Association linking a task to a human resource.	StartDate EndDate
Task	A task can be executed by a human resource. An influence from an external system can impact a task.	ID Description IsExecuted Deadline Nature Criticality level

3.3. Procedure

The third part of our operational monitoring method is a procedure to guide step by step the usage of the modeling language. Drawing from the existing base of knowledge, continuous improvement is a central aspect of the ISO security standards. This translates into a cyclical security management procedure [5]. Using a PDCA cycle [25], operational monitoring would take place during the "Do" and "Check" phases. However, with the facilities enabled by the formal language, we can have a higher frequency of verification.

This is reflected in more frequent cycles of verification and testing. In Figure 1, we zoom into the "Do" and "Check" phase of the overall security management in order to highlight the usage of our method in operational monitoring.

First Step: The system definition resembles that of a general security management approach. It allows us to determine the scope of the system we plan to assess and aim to protect. This also clarifies the borders of the system properly defining the flow of interactions with other systems. The resulting work product is a mapped out system through a security block diagram. This diagram would present assets including their characteristics and collaborations, human resources involved and systems with which the system in question interacts.

Second Step: The organizations must devise an action plan that aligns with their internal goals and security requirements. That action plan is then adapted depending on existing resources and human resources capabilities.

Third Step: The action plan is down-scaled into actionable assignments. These duties and responsibilities are then allocated to the different actors within the system bounds with specific deadlines. This is where, traditionally, managers would use classic management tools defining tasks using User Stories that can be translated into model elements: "As a *Responsibility* I must *Task* by *Deadline*". This information is used to complete the previously mentioned security diagram.

Fourth Step: Human resources execute the tasks allocated to them. Task execution duration is task dependent and supervisions needs to be adapted accordingly.

Fifth Step: Supervisors and managers can design OCL tests to monitor task execution or do bulk monitoring. These tests can immediately yield undone tasks, return human resource work load or other tasks where manual verification would be arduous.

Sixth Step: These managers can determine, through multiple pre-designed tests, the current operational posture for their responsibilities within the system. The cycle then restarts, devising a new action plan based on previous results and future goals.

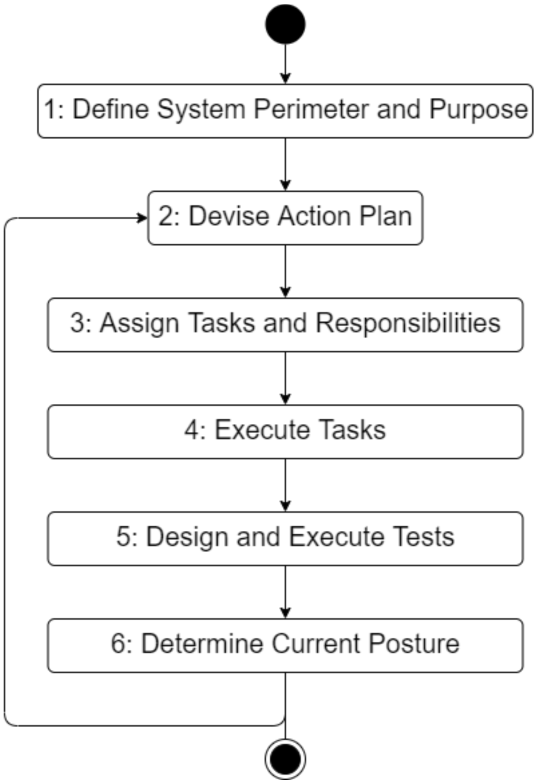
1: Define System Perimeter and Purpose
2: Devise Action Plan
3: Assign Tasks and Responsibilities
4: Execute Tasks
5: Design and Execute Tests
6: Determine Current Posture

Figure 1: Operational monitoring procedure steps

3.4. Tool

In order to produce a functioning graphic editor for our language to support our method, we set out to find an existing well renowned modeling tool to extend. In Table 3, we set out to find a free and open source modeling solution.

Table 3: Renowned Modeling Tool Comparison

Modeling Tool	Author	OpenSource	Free
Modelio	Modelio Corp.	Yes	No
Entreprise Architect	Sparx Systems	No	No
MagicDraw	NoMagic	No	No
Eclipse Papyrus	Eclipse Foundation	Yes	Yes
Eclipse Sirius	Eclipse Foundation	Yes	Yes

Both Sirius and Papyrus are adequate choices, the scale tipped in favor of Papyrus for the availability of support and documentation. After this selection, the profile for the security SysML extension is created in a separate project. It is meant to act as a new extension point to be added to the profile list. This gives us, through the option of palette configuration, the possibility to add the profile elements. We can also define their graphical syntax i.e. new shapes if needed, we elected to leave the original block shape with their respective stereotype. The resulting palette is shown in Figure 2. As it is for tests through OCL, Eclipse Papyrus supports the addition and execution of any OCL queries written in all Papyrus projects. By right clicking within the project, Eclipse allows the addition of OCL files, their execution and validation through the drop-down menu in Figure 3.

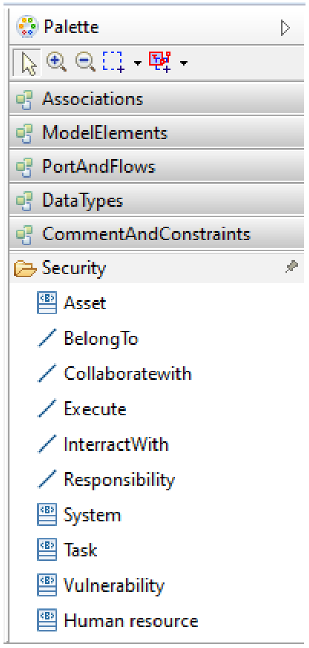

Figure 2: Security Responsibility Palette

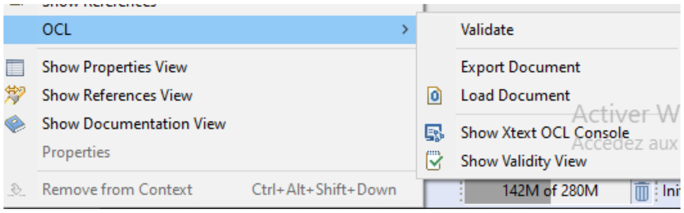

Figure 3: Dropdown Menu for OCL

4. Capital One Breach Case Study

In this section, we follow the operational responsibility and task monitoring method in analyzing the Capital One data breach. At first, we analyze the impact of Capital One's monitoring practices. Secondly, we use the data breach as a case study to apply the steps of this method. Finally, we highlight the lessons learned from this case study.

4.1. Methodological Consideration

Because of the rarity of details concerning information security management, analyzing the detailed study [2] would yield interesting insights. Their study aligns with our vision since their initial hypothesis: Renowned standards do not provide enough guidance for organizations for implementation nor incident management. Also, organizations aren't capable of implementing and maintaining security controls. One of the recommendations made was to find a way to manage the time frame between control implementation, evaluation and audit. This is precisely where our study intervenes analyzing task and responsibility operational monitoring. As per the typology provided in [26] and [27], our units of analysis in this "Descriptive Case Study" are the tasks or responsibilities of a single human resource. Our conceptual framework is that of our proposed solution, detailed in section 3. 1., as we will be studying the different human resources and assets intervening in Capital One's systems as well as the different responsibilities and tasks. We will address them through the research questions previously presented in the introduction:

- RQ1: How did Capital One track the chain of responsibility in their data breach?
- RQ2: How did Capital One security managers verify the implementation of security controls?
- RQ3: How to model tasks and responsibilities in a multi-scale information system?
- RQ4: How to formalize operational monitoring in a multi-scale information system?

4.2. Procedure

Capital One is ranked eighth largest bank overall with a revenue of around 28 billion dollars in 2018. Technology implementation and consistent progress are considered driving ideologies within Capital One. As it is the case in banking, they abide by several security standards such as the New York Stock Exchange corporate governance rules, as well as being one of the participants in supplementing the NIST security standards. Capital One are considered pioneers in migrating their data centers to a cloud environment, environment from which the breach stemmed later. In spite of all their efforts, a breach disclosing the personal data of around 106 million individuals was discovered in July the 19th, 2019. According to the investigations that followed, an employee for the cloud storage service provider Amazon Web Service (AWS) created a scanning tool that allowed the culprit to recognize servers with firewall misconfigurations allowing access to buckets of data. This permitted the later execution of a set of scripts to retrieve access credentials then copy the now available data.

4.3. Analysis

Following the steps of our procedure described in section 3.3, we address primarily the failure in the implementation of technical security controls from Capital One's side. In the ***First step***, we describe the contents of the Capital One Security Management System. They have invested in hiring a renowned Chief Information Security Officer (CISO), talented security engineers as well as on other security related investments for tool development.

Secondly, the bank's action plan allegedly revolves around implementing information security standards such as the NYSE requirement or NIST security controls. This meant that the CISO had the duty of choosing the security controls he deems adequate to protect his system. He then had to specify the time and resources allocated for each security control as well as defining the evaluation and audit periodicity. This is also when he's supposed to decide the means by which the controls are to be satisfied. First of all, a Web Application Firewall (WAF) needs to be implemented and configured to block any entry from malicious proxy servers or from TOR exit nodes. Secondly, set up proper periodic vulnerability scans for the WAF. Lastly, revoke administrative account access credentials in case of internal changes. In the ***third step***, the CISO should assign these duties to the security engineer and the security manager. In the case as we understand it, there was time window in which the attack occurred and that is between task execution in ***step four*** and the monitoring test that would have identified the failed NIST controls in ***step five***.

Research Question 1: In an article by The Wall Street Journal, Capital One attributes the problem to an error in its own infrastructure. However, in a second article in the same journal, interviewing Capital One employees reported that multiple issues are at the root of this breach. Prior to the incident, there have been complaints that employees have raised concerns that they required more software to spot breaches. This delay is caused by the difference in skillset between IT professionals and governance professionals. As systems become more interconnected, all involved parties must acquire multidisciplinary skills so that governance professionals understand the necessary requirements in terms of technology to be able to provide them within a smaller timeframe. This could also be linked to failures to stay within budget lines despite hefty investments. This shows that Capital One failed to trace back the source of their infrastructural issues that is managerial and financial at its core.

Research Question 2: A second issue voiced by employees was low morale due to an increase in number of employee terminations. This is also due to budget issues that later became detrimental to routine security control implementation. Employees also reported an internal practice giving liberties to programmers the freedom to code in whatever language they choose if it would help complicate breaking into their systems in turn complicating pen-testing. These two incidents reflect that Capital One also lacked visibility as well as control over the tasks executed inside its scope. The Capital One security managers knew the necessary security controls to implement, however, they lacked regular verification of their implementation. This is what created the timeframe the culprit needed to execute his attack.

4.4. Lessons Learned

In the previous subsection, we clarified that Capital One had issues regarding task and responsibility tracking. Now, we will address these issues through discussing the second set of research questions.

Research Question 3: In our previous analysis, the first issue was the lack of multi-scale visibility coupled with lack of understanding between engineers in the security management system and their higher-ups in top management. This an aspect that our solution addresses via a complex system vision, where multiple systems interact with one another and therefore influencing one another. Using our method, in the first step, we can model Capital One as two systems "Governance System" and "Security Management System" each containing their respective human resources. As is shown in Figure 4, we can model the budget that the first system provides to the next or how the financial issues can influence security management. In this case, any employee dismissed must be removed from the model. Carrying on, we add to each human resource their set of tasks to which we assign deadlines and the attribute *Isexecuted* to be set to "True" when it's done. Figure 5 shows the different security tasks and their responsible. Each task is the implementation of a necessary NIST security control [2]. In the case of the layoffs and human resource removal, the tasks would remain and it would be apparent that they need to be reassigned in order to maintain security controls. Verification for human resource task load and tasks with no *responsible* association can be done through OCL tests. In addition, each human resource can find her own tasks by executing a script. This script would return all tasks that are assigned to the human resource with a *responsible* association.

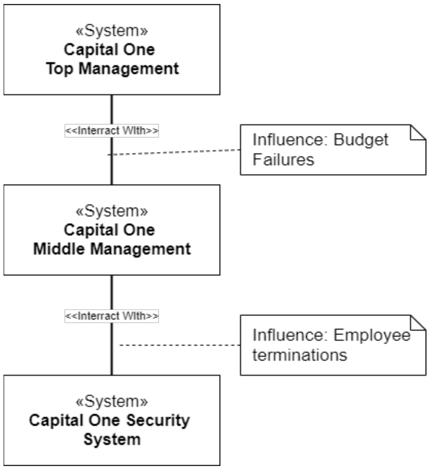

Figure 4: System Influence Diagram

Research Question 4: Following task modeling and responsibility assignment, it is the managers' duties to design tests for task supervision. This would normally be achieved by individually verifying each task accomplishment manually. Through the formal modeling of each task and responsibility, we now have the possibility of designing tests where we can list all tasks that have the attribute *IsExecuted* set to "False" and compare the current date to the deadlines. This would also link us immediately to the person responsible for this task execution. Since the day to day supervision tests don't have to be re-written, this makes monitoring less time consuming and therefore more efficient opening the possibility for day to day supervision. In the case of Capital One, this would have resulted in the revocation of the culprit's access credentials earlier as well as reassigning other employee's tasks in time. In turn, this would have closed any timeframe required to conduct the attack.

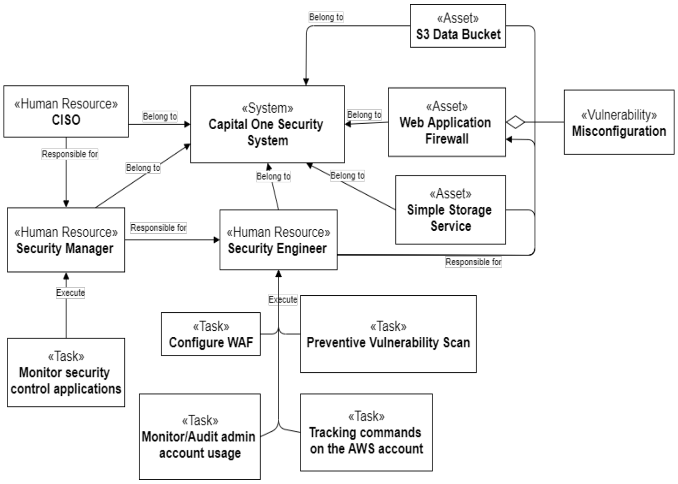

Figure 5: Operational Traceability diagram for Capital One Case

5. Conclusion

This paper presents a method for operational security monitoring applied to a security breach case. It aims to bridge the gap identified between security control identification and implementation. We have presented the components of our method that provide a much-needed formalism in security management. We have built this method using prominent modeling and security management standards in the base of knowledge. We have chosen to extend the SysML modeling standard with security monitoring concepts. This allows the creation of specialized block diagrams that model the actors and assets within a system as well as the tasks and responsibilities. The resulting diagram is a map capable of capturing complex system composition and interaction. Furthermore, this formalism provides the ability to use OCL for proper monitoring and supervision even for complex systems. In fact, the information depicted is of different use to different actors. OCL queries allow managers to regularly verify task execution and manage task loads. They also allow security engineers to see which new tasks were assigned to them or whether undone tasks still remain. Moreover, we have highlighted through our procedure the issues usually faced during the "Do" and "Check" phases of the Deming wheel. This allowed us to showcase the importance and applicability of our method for security management onto a high profile case. The described case study provided a high level of detail that allowed us to properly study the applicability and utility of our proposed method. Our method does not circumvent the complexity of the systems studied, it retains all the necessary information for operational monitoring. It aims to provide the tools that different actors need to carry out their roles in security management.

For future studies, we intend to include this study within a bigger whole designed for security management to reach "security by design". We also plan to further improve the modeling tool facilitating tests and creating a plugin to be featured in Eclipse Marketplace for public usage.

References

[1] A. Rabii, S. Assoul, K. Ouazzani Touhami, O. Roudies, "Information and cyber security maturity models: a systematic literature review," Information and Computer Security, **28**(4), 627–644, 2020, doi:10.1108/ICS-03-2019-0039.

[2] N. Novaes Neto, S.E. Madnick, A. Moraes G. de Paula, N. Malara Borges, "A Case Study of the Capital One Data Breach," SSRN Electronic Journal, (January), 0–24, 2020, doi:10.2139/ssrn.3542567.

[3] A. Alhogail, "Design and validation of information security culture framework," Computers in Human Behavior, **49**, 567–575, 2015, doi:10.1016/j.chb.2015.03.054.

[4] R. Anass, A. Saliha, R. Ounsa, "A Concept &amp; Compliance Study of Security Maturity Models with ISO 21827," in Proceedings of the 22nd International Conference on Enterprise Information Systems, SCITEPRESS - Science and Technology Publications: 385–392, 2020, doi:10.5220/0009569703850392.

[5] A. Rabii, S. Assoul, O. Roudies, "Security requirements elicitation: A smart health case," Proceedings of the World Conference on Smart Trends in Systems, Security and Sustainability, WS4 2020, 776–781, 2020, doi:10.1109/WorldS450073.2020.9210330.

[6] F. Lahboube, O. Roudies, N. Souissi, "Extending supervision metamodel to complex system context," in 2015 Third World Conference on Complex Systems (WCCS), IEEE: 1–7, 2015, doi:10.1109/ICoCS.2015.7483242.

[7] Y. Bar-Yam, "General Features of Complex Systems," Knowledge Management, Organisational Intelligence and Learning and Complexity, **I**(1), 1–10, 1997.

[8] J. Backhouse, G. Dhillon, "Structures of responsibility and security of information systems," European Journal of Information Systems, **5**(1), 2–9, 1996, doi:10.1057/ejis.1996.7.

[9] M. Kassou, L. Kjiri, "SOASMM: A novel service oriented architecture Security Maturity Model," in 2012 International Conference on Multimedia Computing and Systems, IEEE: 912–918, 2012, doi:10.1109/ICMCS.2012.6320279.

[10] MMISS-SME Practical Development: Maturity Model for Information Systems Security Management in SMEs, in Proceedings of the 5th International Workshop on Security in Information Systems, SciTePress - Science and and Technology Publications: 233–244, 2007, doi:10.5220/0002430402330244.

[11] G.B. White, "The community cyber security maturity model," in 2011 IEEE International Conference on Technologies for Homeland Security (HST), IEEE: 173–178, 2011, doi:10.1109/THS.2011.6107866.

[12] S. Yulianto, C. Lim, B. Soewito, "Information security maturity model: A best practice driven approach to PCI DSS compliance," in 2016 IEEE Region 10 Symposium (TENSYMP), IEEE: 65–70, 2016, doi:10.1109/TENCONSpring.2016.7519379.

[13] K.D. Mitnick and William L. Simon, The Art of Deception: Controlling the Human Element of Security, John Wiley & Sons, Inc., 2003, doi:10.5555/861316.

[14] J.P. Jeretta Horn Nord,Alex Koohang,Kevin Floyd, "IMPACT OF HABITS ON INFORMATION SECURITY POLICY COMPLIANCE," Issues in Information Systems, **21**(3), 217–226, 2020, doi:https://doi.org/10.48009/3_iis_2020_217-226.

[15] A. Longras, T. Pereira, P. Carneiro, P. Pinto, "On the Track of ISO/IEC 27001:2013 Implementation Difficulties in Portuguese Organizations," in 2018 International Conference on Intelligent Systems (IS), IEEE: 886–890, 2018, doi:10.1109/IS.2018.8710558.

[16] K.H. Rose, "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)-Fifth Edition," Project Management Journal, **44**(3), e1–e1, 2013, doi:10.1002/pmj.21345.

[17] R.B. Fuller, "A comprehensive anticipatory design science," **34**, 357–361, 1957.

[18] K.H. Guo, Y. Yuan, N.P. Archer, C.E. Connelly, "Understanding Nonmalicious Security Violations in the Workplace: A Composite Behavior Model," Journal of Management Information Systems, **28**(2), 203–236, 2011, doi:10.2753/MIS0742-1222280208.

[19] K. Huang, K. Pearlson, "For What Technology Can't Fix: Building a Model of Organizational Cybersecurity Culture," 2019, doi:10.24251/HICSS.2019.769.

[20] S. Ouchani, O.A. Mohamed, M. Debbabi, "A Security Risk Assessment Framework for SysML Activity Diagrams," in 2013 IEEE 7th International Conference on Software Security and Reliability, IEEE: 227–236, 2013, doi:10.1109/SERE.2013.11.

[21] L.A. Yves Roudier, "SysML-Sec - A Model Driven Approach for Designing Safe and Secure Systems," in Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development, SCITEPRESS - Science and Technology Publications: 655–664, 2015, doi:10.5220/0005402006550664.

[22] F. Lugou, L.W. Li, L. Apvrille, R. Ameur-Boulifa, "SysML Models and Model Transformation for Security," in Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development, SCITEPRESS - Science and and Technology Publications: 331–338, 2016, doi:10.5220/0005748703310338.

[23] A. Dardenne, A. van Lamsweerde, S. Fickas, "Goal-directed requirements acquisition," Science of Computer Programming, **20**(1–2), 3–50, 1993, doi:10.1016/0167-6423(93)90021-G.

[24] J.A. Highsmith, Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, Dorset House Publishing Co., Inc., 2000, doi:10.5555/323922.

[25] W.E. Deming, Out of the Crisis, The MIT Press, 2018, doi:10.7551/mitpress/11457.001.0001.

[26] T. Hollweck, "Robert K. Yin. (2014). Case Study Research Design and Methods (5th ed.). Thousand Oaks, CA: Sage. 282 pages.," The Canadian Journal of Program Evaluation, 2016, doi:10.3138/cjpe.30.1.108.

[27] P. Baxter, S. Jack, "The Qualitative Report The Qualitative Report Qualitative Case Study Methodology: Study Design and Qualitative Case Study Methodology: Study Design and Implementation for Novice Researchers Implementation for Novice Researchers," Number 4 Article, **13**(4), 12–13, 2008.