Towards Automating the Security Compliance Value Chain

Smita Ghaisas, Manish Motwani, Balaji Balasubramaniam, Anjali Gajendragadkar, Rahul Kelkar, Harrick Vin
Tata Consultancy Services Ltd.
Tata Research Development and Design Centre (TRDDC), Pune, India
{smita.ghaisas, manish.motwani, balaji.balasubramaniam, anjali.sg, rahul.kelkar, harrick.vin}@tcs.com

ABSTRACT
Information security is of paramount importance in this digital era. While businesses strive to adopt industry-accepted system-hardening standards such as benchmarks recommended by the Center for Internet Security (CIS) to combat threats, they are confronted with an additional challenge of ever-evolving regulations that address security concerns. These create additional requirements, which must be incorporated into software systems. In this paper, we present a generic approach towards automating different activities of the Security Compliance Value Chain (SCVC) in organizations. We discuss the approach in the context of the Payment Card Industry Data Security Standard (PCI-DSS) regulations. Specifically, we present automation of (1) interpretation of PCI-DSS regulations to infer system requirements, (2) traceability of the inferred system requirements to CIS security controls (3) implementation of appropriate security controls, and finally, (4) verification and reporting of compliance.

Categories and Subject Descriptors
K.4.1 [**Computers and Society**]: Public Policy Issues – *regulation,* Organizational Impacts – *Automation, computer-supported collaborative work.*

K.5.2 [**Legal Aspects of Computing**]: Governmental Issues – *regulation*

General Terms
Regulatory compliance, Security Benchmarks, Security Compliance Value Chain, PCI-DSS, CIS, Automated interpretation.

Keywords
Rule act, Rule intent, Rule Model.

1. INTRODUCTION
Intensifying regulatory pressure continues to be the main factor that drives spending on security world over [1, 3]. Due to their highly specialized diction, regulations are hard to interpret. Moreover, they evolve and create requirements that pose increasing demands on security practices in organizations [5]. Businesses routinely hire Qualified Security Assessors (QSAs) to identify and interpret the regulations that are applicable to their systems [2]. Organizations such as the Center for Internet Security (CIS) [15] produce benchmarks for improving the security effectiveness of businesses. The benchmarks when implemented, guarantee a widely accepted level of system-hardening; however, they do so without any specific reference to the individual regulations in a given security act. Implementing recommended security benchmarks therefore, does not necessarily promise compliance with a given set of regulations.

We interacted with 20 in-house legal, domain, technical experts and program managers whose experience varies from 14 to 20 years, in order to find out the activities performed for ensuring regulatory compliance. We learnt that a regulatory compliance lifecycle consists of following activities: The Chief Compliance Officer (CCO) gets notified of the new or changed regulations. The Legal experts then interpret the regulations and assign them to the respective business units. The Business Unit Owners (BUO) assign the changes pertaining to their line of business and jurisdiction to the respective Business and IT Analysts. Business and IT Analysts perform a detailed impact analysis to identify the impact of regulatory changes on existing applications, policies, procedures, etc. Compliance managers implement the changes with the help of respective project teams. Finally, the compliance is reported and continuously monitored. BUO/Program Managers send a consolidated report of the highlights of the Impact Assessment to the CCO. The CCO furnishes responses and periodic reports on compliance to the regulator.

We derive a Security Compliance Value Chain (SCVC) from the activities in regulatory compliance lifecycle. The SCVC consists of the following four crucial activities: (1) **Interpret** regulations in terms of implementation-specific validations. (2)**Trace** regulations to artifacts (such as requirements and test cases) and/or to security controls in order to identify the impact of regulations on systems. (3) Accordingly, **Implement** the changes/updates in the system. (4) **Verify** and **Report** the compliance of the systems. In this paper, we present a novel method to automate these four activities of SCVC. The method utilizes the Rule Model from our previous work [16]. To the best of our knowledge, a method that achieves an end-to-end automation of SCVC activities in an integrated manner has not been reported so far.

2. METHOD AND TOOLSET FOR AUTOMATION
Figure 1 shows the sequence of activities in the SCVC. We discuss the method in the context of the PCI-DSS regulations and CIS security controls. Regulations are rules that are composed of several (intended) constraints. We term these constraints 'Rule intents' [16]. We represent each regulation as a composition of Rule intents. The groups of frequently co-occurring Rule intents are inspected to label the 'Rule acts'. Analogous to the Speech acts [21], from Linguistics that express apology, gratitude, and request; the Rule acts explicate system validations such as *access*

ESEC/FSE'15, August 30 – September 4, 2015, Bergamo, Italy
© 2015 ACM. 978-1-4503-3675-8/15/08...$15.00
http://dx.doi.org/10.1145/2786805.2804435

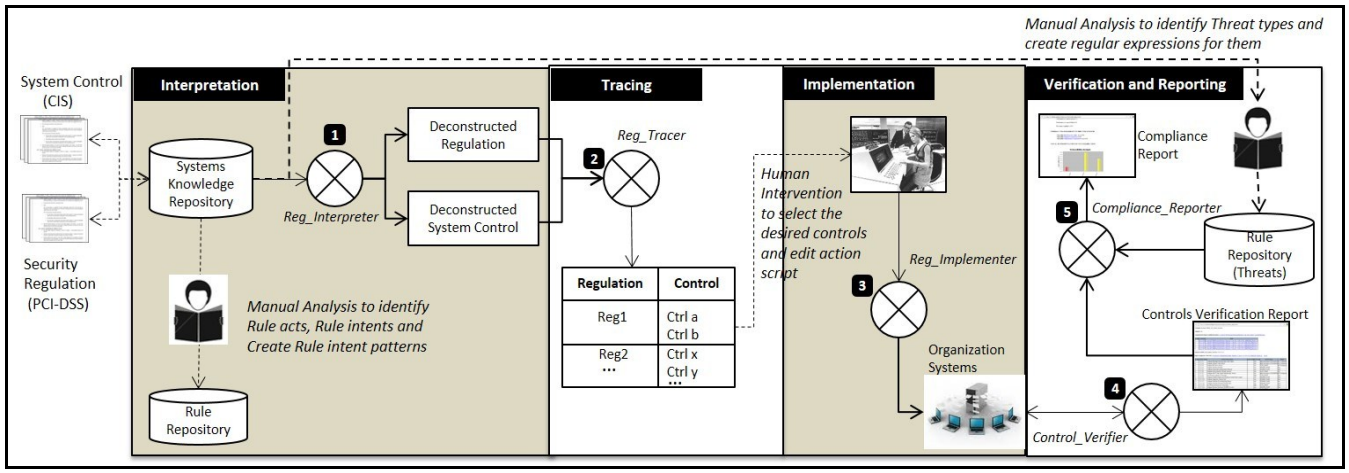

Figure 1. Automation of Security Compliance Value Chain.

control, *data validation*, and *conditional execution*. The labeling is a one-time activity. We have created a Rule Repository that contains Rule acts, Rule intents and associated Rule intent patterns [16].

2.1 Dataset

This sub-section describes the datasets of PCI-DSS regulations and CIS security controls, and the details of their analysis.

2.1.1 PCI-DSS Regulations

We considered PCI-DSS regulations from PCI-DSS version 3.0 [14]. There are 12 PCI-DSS requirements. For each PCI-DSS requirement, there are different clauses and corresponding to each clause, there exists a security assessment (testing) procedure and guidance. Henceforth, by regulation, we mean a clause along with its security assessment procedure. We have a total of 209 such regulations. These can be classified as Organizational, Technological and Third party [2]. Our analysis of the regulations reveals that out of 209, 189 regulations are of type *Technological*. We interpret all the 209 regulations; however, we trace, implement, and verify and report 189 regulations of the type *Technological*. The regulations from the other two categories are not amenable to automation; they need a manual intervention.

We selected 100 out of 209 regulations using random sampling to form a training dataset. We analyzed this dataset to identify instances of Rule Model elements. This is a one-time activity and took approximately 1.5 person days. At the end of the analysis, we identified 17 new Rule intents and 139 Rule intent patterns were created. We identified 7 new Rule acts. The new Rule acts, Rule intents and Rule intent patterns were added to the existing Rule Repository [16]. For example, the new Rule act **Hardware Configuration** (Rule act that specifies hardware details for security mechanisms) is composed of Rule intents - *system, network device, storage device, server, activity,* and, *condition.* The ground truth for interpretation was prepared as discussed in our previous work [16].

2.1.2 CIS Security Controls

We analyzed 434 CIS security controls for Microsoft Windows Server 2008 R2 [15]. Henceforth by controls we mean CIS security controls. The analysis was aimed at identifying the controls (classified in terms of control categories) that are associated with 189 regulations of type *Technological*; and create an 'answer sheet' for traceability. This involved examining 82,026 candidate trace links. This again is a one-time activity and took approximately 5 person days.

2.2 Interpreting Regulations

This is the first activity of the SCVC. We use the *Reg_Interpreter* module which includes (1) a component of *Rule Extractor* (to automatically identify the Rule intents from a given regulation using Rule intent patterns) and (2) *Rule classifier* (to classify a regulation into Rule acts) from our previous work [16]; to interpret regulations and in terms of the implementation specifics they imply. This involves applying NLP-based techniques to identify the parts of speech (POS) [17] structure of a textual regulation. The *Reg_interpreter* module matches the Rule intent patterns from the Rule Repository against the POS tags and phrases of the textual regulation to identify Rule intents. Using the *Rule classifier* module, it then detects frequently co-occurring clusters of Rule intents and identifies the clusters in terms of the Rule acts. A single regulation may get classified in terms of one or more Rule acts. For example, a regulation "*8.1.4- Remove/disable inactive user accounts at least every 90 days.*" is found to contain **Rule intents**: activity (*remove/disable*), object (*inactive user accounts*), and threshold (*at least every 90 days*); and gets classified into **Rule acts**: **Data Validation** which is composed of *activity* and *object* and, **Conditional execution,** which is composed of *threshold* and *activity*. This kind of interpretation can help Business and IT Analysts identify relevant regulations and the implementation specifics they necessitate for the software systems. Table 1 presents the results of the automated interpretation of regulations. We use precision and recall to evaluate our interpretations. We are able to achieve an average precision and a recall of 80.86% and 83.20% respectively.

2.3 Tracing Regulations to Security Controls

This is the second activity of the SCVC. We identify Rule acts from the source (regulations) and target (controls) datasets using Rule Model [16]. Using these as topics we trace regulations to controls. We benchmark this method with traceability results obtained using the Latent Dirichlet Allocation (LDA) technique [18], a widely used topic modeling technique.

We represent regulations and controls in terms of Rule Model elements using the *Reg_Interpreter* module, as described in sub-section 2.2. The *Reg_Tracer* module then uses Rule acts as Topics (similar to the LDA topics) and computes the similarity scores between regulations and controls. The computation of similarity

Table 1. Regulation Interpretation Results

Rule act	#ReT	#ReTL	#ReL	Precision (%)	Recall (%)
Deadline	29	20	22	68.97	90.91
Data Validation	31	27	33	87.10	81.82
Software	55	26	31	47.27	83.87
Conditional Execution	56	44	45	78.57	97.78
Access Control	19	18	21	94.74	85.71
User Responsibility	54	43	43	79.63	100.00
Vulnerability Management	17	17	22	100.00	77.27
Hardware	16	14	20	87.50	70.00
System Setting	36	27	32	75.00	84.38
System Environment	25	21	28	84.00	75.00
Network Protocol	15	13	19	86.67	68.42
	Average			80.86	83.20

#ReT: # regulations retrieved for a given Rule act
#ReTL: # regulations that are retrieved and relevant for a given Rule act
#ReL: # regulations that are relevant for a given Rule act

scores takes into account the number of common Rule acts and Rule intent instances identified from both; a given regulation and a control. The precision at recall 100% obtained using LDA technique is found to be in range 0.2-2.7 whereas that using Rule Model is in the range 0.9-7.1. We consider this measure because in the case of regulations, recall assumes greater importance; false negatives are riskier.

Analysts inspect the controls traced by the *Reg_Tracer* and select the controls that need to be implemented. An automatically generated script template facilitates implementation of the selected controls. Software developers need to specify the values of hostnames and system variables in the script template.

2.4 Implementing Controls for Compliance

This is the third activity of SCVC. The *Reg_implementer* module takes as input, the script generated and edited by software developers to implement the controls. It uses a Systems Knowledge Repository (SKR) to identify the relevant executable commands using which the values of the system variables can be set or verified. Currently, this repository contains information related to 3 compliance sources namely PCI-DSS, SoX [23] and CIS. The commands and system variables are specific to a given technology type, a security benchmark, and a control. For e.g., to implement the CIS (compliance source) recommended control for setting the account lockout duration to 15 or greater, the *Reg_Implementer* module uses 'gpresult/Z' (command) and 'LockoutDuration' (system variable) for Windows 2008R2 (technology) from the SKR. The SKR is created and maintained for consistency by the security team in our organization. It harnesses the expert knowledge and experience in implementing controls to achieve certifiable organizational compliance.

2.5 Compliance Verification and Reporting

This is the fourth activity of the SCVC. The *Control_verifier* module verifies whether all the selected controls are correctly configured in systems. It uses the SKR to identify the relevant commands and check the values of system variables. The output is a Control Verification Report generated for each system. For each control tested, the report shows either (a) pass, or (b) fail, or (c) not configured. The controls that fail or are not configured are non-conformities while the ones that pass are termed as conformities.

To identify the potential threats that may result from non-compliance, we analyzed the information in the guidance column corresponding to each clause of PCI-DSS requirement and the rationale for the CIS security controls From the analysis, we identified different Threat types. Corresponding to each Threat type, Threat patterns (similar to Rule intent patterns) were created. The Rule Repository was augmented with the Threat types and corresponding Threat patterns. A total of 13 Threat types and 291 Threat patterns were created. This again is a one-time activity and took approximately 3 person days. Table 2 shows few examples of Threat types (TY).

Table 2. Threat types and Threat patterns in Rule Repository

S.No	Threat type	#Threat patterns
1	System Malfunctioning	34
2	System Availability Failure	20
3	Misused User Privilege	18
4	Data Protection Failure	32
5	Unwanted Code Execution	24

The *Compliance_Reporter* module takes as input the Control Verification Report generated by the *Control_Verifier* module and uses Threat patterns to identify the potential threats (from regulations and controls) due to non-conformities. The output generated is a Security Compliance Dashboard that displays for each system, (a) potential threats corresponding to non-conformities, (b) percentage compliance and, (c) overall threat criticality scores for a system. The Threat criticality score of a system is calculated by:

$$\mathbf{ThreatCriticalityScore_{TY} = W_{TY} \times (\#Threat\_Patterns\_Detected_{TY}) / (\#Total\_Threat\_Patterns_{TY})}$$

Here, **W** is the configurable criticality weight (range 0-1) determined based on organization's priority for each TY.

3. RELATED WORK

Several approaches address different activities of the SCVC. The approaches that address **Interpretation** involve modeling regulations using formal or semi-formal languages. The existing models are abstract and do not represent regulations in terms of implementation specifics they imply [19]. Besides, the modeling activities are largely manual in nature [3, 5, 8, 9, 20, 22]. Massey, et al. [24] highlight that legalese is extremely complex and presents a readability challenge to requirements engineers seeking to analyze it. There exist several domain-specific and domain-agnostic models and methods to establish **Traceability** between regulations and software artifacts [6, 7, 13, 19]. However, most of these approaches are manual in nature, and they do not address the issues of automating the traceability process. Those that are automated do not claim generalizability.. For **Implementation**, the existing compliance solutions are hard-coded and ad-hoc [10]. Such solutions are specific to benchmarks and therefore difficult to be extended generically for security compliance. Moreover, when the regulations and the benchmarks change and evolve, the solutions are rendered unreliable from a compliance point of view. For **Verification** and **Reporting**, there are automated approaches that facilitate the detection and recommendation for

prevention of non-compliance [3, 11, 12]. But they require manual effort for representing the regulations using formal methods and identifying the relevant controls to be validated. The market offerings use hard-coded check repositories that require users to identify the relevant controls to be validated. In [4], authors describe a comparative evaluation of 32 regulatory compliance management (RCM) solution frameworks. The RCM Framework Alignment criteria correspond to the four activities of SCVC. Their evaluation report shows that none of the existing frameworks address all 4 activities of SCVC.

4. CONCLUSION AND FUTURE WORK

In this paper, we identify SCVC from regulatory compliance lifecycle and, automate the activities of SCVC. For automating the interpretation of regulations, we use the Rule Model from our previous work. We are able to achieve an average precision and recall of 80.86% and 83.20% respectively. The obscure diction used in the regulations is an obvious major challenge to any NLP-based approach including ours. We continue to enhance the training inputs to our toolset both;-quantitatively:- (1) in the form of an increasingly comprehensive corpus of vocabularies (represented as Rule intent patterns), and qualitatively:- (2) by taking into account the permutations of Rule intents in a regulation, and (3) bringing in a pragmatics-based approach that takes into account the unstated ramifications of regulations. We further use the Rule Model to trace regulations to controls. Finally, for verification we propose a method that enables organizations to identify potential threats due to non-compliance by generating a system wise compliance report. We recognize the validity to threat associated with the focus on PCI-DSS regulations and CIS security controls for testing the approach and posit that this is but a first step towards addressing security compliance in an integrated and automated manner.

5. REFERENCES

[1] Abdullah N. S., Indulska M., Sadiq S., A Study of Compliance Management in Information Systems Research In: The 17th European Conference on Information Systems, Verona, Italy (2009) Verona, Italy, pp. 1–10.

[2] Rees J., The challenges of PCI-DSS compliance, Computer Fraud & Security Volume 2010, Issue 12, pp. 14–16.

[3] Giblin C., Muller S., Pfitzmann B., From regulatory policies to event monitoring rules: towards model driven compliance automation, IBM Research Report. Zurich Research Laboratory Oct' 2006.

[4] Kharbili M., Business process regulatory compliance management solution frameworks: a comparative evaluation, in: APCCM 2012. CRPIT, vol. 130, pp. 23–32.

[5] Maxwell J., Reasoning about legal text evolution for regulatory compliance in software systems, Ph.D. dissertation, North Carolina State University, 2013.

[6] Lovrić Z., Model of simplified implementation of PCI DSS by using ISO 27001 standard, In proceedings of Central European Conference on Information and Intelligent System, Sept 19-21' 2012, pp. 347-351 Varaždin, Croatia.

[7] Srivastav A., Ali I., Kumar N., Shanker R., A simple prototype for implementing PCI DSS by using ISO 27001 frameworks, In Intl. Journal of Advanced Research in Computer Science and Software Engg., Vol 4, Issue 1, Jan 2014, pp 886-889.

[8] Gordon G. D., Without borders: addressing legal requirements in multi-jurisdictional IT environments, PhD dissertation, Carnegie Mellon University Pittsburgh, 2014.

[9] Sapkota K., Aldea A., Younas M., Duce D. A., and Banares-Alcantara R., Extracting meaningful entities from regulatory text, in Proceedings of the 5th Intl. Workshop on Requirements Engineering and Law, 2012, pp. 29-32.

[10] Rempel P., Mäder P., Kuschke T., Huang J. C., Mind the gap: assessing the conformance of software traceability to relevant guidelines, In Proceedings of the 36th ICSE, May 31-June 07, 2014, Hyd., India, pp. 943-954.

[11] Natarajan K., and Ramyachitra D., Engineering Perspective on Enterprise Software Quality towards Compliance Management for better GRC., Intl. Journal of Engineering Sciences Research, vol. 5, article 08375; Oct' 2014.

[12] CIS Tools http://benchmarks.cisecurity.org/downloads/audit-tools/ last accessed on March 6, 2015

[13] Huang J. C., Czauderna A., Gibiec M., and Emenecker J., "A machine learning approach for tracing regulatory codes to product specific requirements," In proceedings of the 32nd ACM/AEEE ICSE, pp.155-164

[14] Payment Card Industry Data Security Standard (PCI-DSS): https://www.pcisecuritystandards.org/index.php last accessed on March 6, 2015.

[15] Center for Information Security (CIS): http://www.cisecurity.org/ last accessed on March 6, 2015.

[16] Ghaisas S., Motwani M., Anish P. R., Detecting system use cases and validations from documents, In proceedings of the 24th IEEE/ACM Intl. Conference on Automated Software Engineering (2013), pp.568-573.

[17] OpenNLP Part-of-Speech (POS) Tags: Penn English Treebank, http://blog.dpdearing.com/2011/12/opennlp-part-of-speech-pos-tags-penn-english-treebank/ last accessed on March 6, 2015.

[18] Rus V., Niraula N. and Banjade R., Similarity measures based on Latent Dirichlet Allocation, Computational Linguistics and Intelligent Text Processing. Springer Berlin Heidelberg. pp. 459-470, 2013.

[19] Breaux T. D. and Gordon D. G., Regulatory requirements traceability and analysis using semi-formal specifications, Requirements Engineering: Foundation for Software Quality (2013), pp.141-157, Lecture Notes in Computer Science Volume 7830.

[20] Regulatory Compliance Demystified: An Introduction to Compliance for Developers. http://msdn.microsoft.com/en-us/library/aa480484.aspx. last accessed on March 6, 2015

[21] Searle, I. R, Speech Acts: an essay in the philosophy of language, Cambridge : Cambridge University Press, 1969.

[22] Boella G., Janssen M., Hulstijn J., Humphreys L., & Torre van der L., Managing legal interpretation in regulatory compliance. In proceedings of the 14th Intl. Conference on Artificial Intelligence and Law - 2013 (pp. 23–32). New York: ACM Press.

[23] The Sarbanes-Oxley Act (SoX): http://www.soxlaw.com/ last accessed on March 6, 2015.

[24] Massey A. K., Eisenstein J., Anton A. I. and Swire P. P., Automated text mining for requirements analysis of policy documents, In proceedings of 21st IEEE Intl. Requirements Engineering Conference, 2013, pp. 4 – 13.