Towards end-to-end Continuous Monitoring of Compliance Status Across Multiple Requirements

Danny C. Cheng[1], Jod B. Villamarin[2], Gregory Cu[3], Nathalie Rose Lim-Cheng[4]
College of Computer Studies, De La Salle University
Manila, Philippines

***Abstract*—Monitoring compliance status by an organization has been historically difficult due to the growing number of compliance requirements being imposed by various standards, frameworks, and regulatory requirements. Existing practices by organizations even with the assistance of security tools and appliances is mostly manual in nature as there is still a need for a human expert to interpret and map the reports generated by various solutions to actual requirements as stated in various compliance documents. As the number of requirements increases, this process is becoming either too costly or impractical to manage by the organization. Aside from the numerous requirements, multiple of these documents actually overlap in terms of domains and actual requirements. However, since current tools do not directly map and highlight overlaps as well as generate detailed gap reports, an organization would perform compliance activities redundantly across multiple requirements thereby increasing cost as well. In this paper, we present an approach that attempts to provide an end-to-end solution from compliance document requirements to actual verification and validation of implementation for audit purposes with the intention of automating compliance status monitoring as well as providing the ability to have continuous compliance monitoring as well as reducing the redundant efforts that an organization embarks on for multiple compliance requirements. This research thru enhancing existing security ontologies to model compliance documents and applying information extraction practices would allow for overlapping requirements to be identified and gaps to be clearly explained to the organization. Thru the use of secure systems development lifecycle, and heuristics the research also provide a mechanism to automate the technical validation of compliance statuses thereby allowing for continuous monitoring as well as mapping to the enhanced ontology to allow reusability via conceptual mapping of multiple standards and requirements. Practices such as unit testing and continuous integration from secure systems development life cycle are incorporated to allow for flexibility of the automation process while at the same time using it to support the mapping between compliance requirements.**

Keywords*—Compliance management, continuous compliance monitoring; ontology mapping; natural language processing; secure systems development lifecycle

I. Introduction

The need to conduct compliance activities within an organization has never been more apparent that it is in recent times. Regulations such as the General Data Protection Regulation or GDPR which aims to protect the privacy rights of individuals make it a requirement for an organization to look into and implement compliance efforts. However, even by just considering the compliance requirements for data privacy alone, an organization would have to consider all the various versions in different countries where the law has a counterpart and the organization is dealing with data subjects from those countries. The number of regulations, standards, frameworks, architectures, and practices that an organization is required to or would benefit from by complying is overwhelming and is continuously increasing in number and complexity as technology and the environment changes over time.

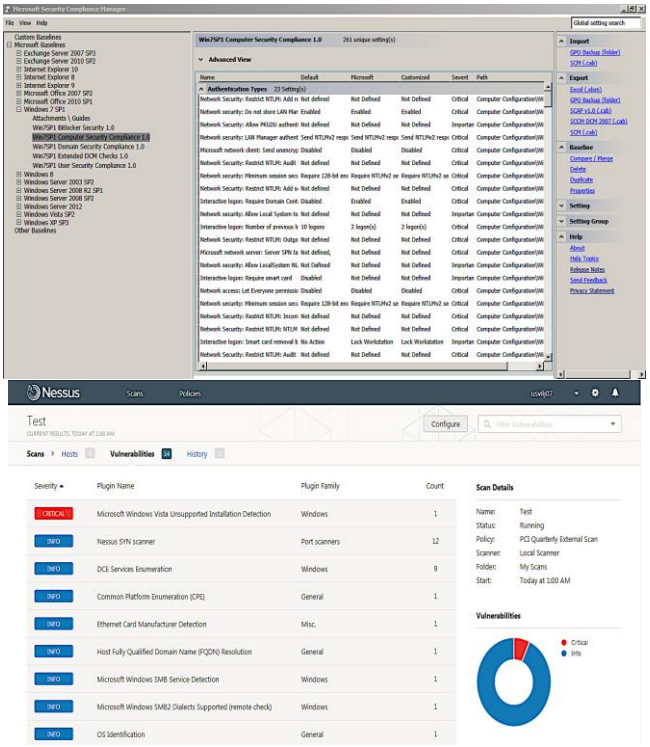
Although they are increasing, it can also be noticed that multiple requirements can also have a large number of overlaps or commonalities that are shared among various regulations, standards, and frameworks. Due to the increase, organizations are attempting to improve on their compliance practices by minimizing redundant new organizational units within the organization [1]. Governance Risk and Compliance (GRC) systems such as those from IBM, SAP, Oracle, and even open source versions such as Eramba have been developed to manage compliance monitoring for enterprises. Systems that generate compliance reports based on predefined templates are also being deployed in an effort to manage and improve on compliance activities. However, compliance monitoring or management systems commonly still rely heavily on human or expert intervention in order to generate reports that answer simple management questions like "If we are already compliant with Standard A what else are we missing to comply with Standard B?". Mapping across requirements and determining overlaps are not commonly found in such systems and thus the process of determining gaps or level of compliance across multiple compliance requirements are repeated for each new regulation as well as for every organizational unit that is to be affected or is required to comply.

Several ontology models for information security, compliance, as well as policy concepts already exist. However, these models have been developed by different researchers whereby information security and compliance are viewed as two separate activities. Although these models are conceptually correct and accurate, the concepts in these models do not capture the concepts that can be seen in the actual compliance requirements statements written in the documents in order to identify, determine, and explain the status of compliance efforts and providing understandable responses to inquiries on level of compliance and areas of non-compliance based on compliance requirements statements to senior management. [2][3][4]

This paper presents a framework for continuous compliance monitoring for multiple requirements documents by enhancing and harmonizing existing ontology definitions for security and compliance focusing on the concepts present in compliance documents. Test cases and unit testing frameworks as practiced in secure software development lifecycle are incorporated in order to audit and validate controls implementation in a flexible and customizable manner. A prototype for mapping test cases and scripts to exact words and phrases in compliance documents which serves as "use cases" allow for the ability to generate reports that show specific deficiencies in compliance based on the document requirements. The mapping can also be used as an input for a lexicon specific to the domain needed in the domain ontology defining concepts such as controls and assets as well as be used to improve the ontology mapping and alignment between different standards and compliance requirements. The research takes a different path by using testing frameworks and scripting based on software quality assurance and secure software development lifecycle methodologies and practices as opposed to the use of existing notations such as Business Process Management Notation (BPMN) as a means to model processes and map to compliance documents [5] as there still exists a great majority of organizations that are not using BPMN within the organization.

II. Ease of Use

Software tools or appliances currently available and deployed to do compliance monitoring, management, and reporting can be grouped together into categories like compliance managers, vulnerability scanners, penetration testers, security events managers, and even governance risk and compliance tools as shown in Fig. 1.



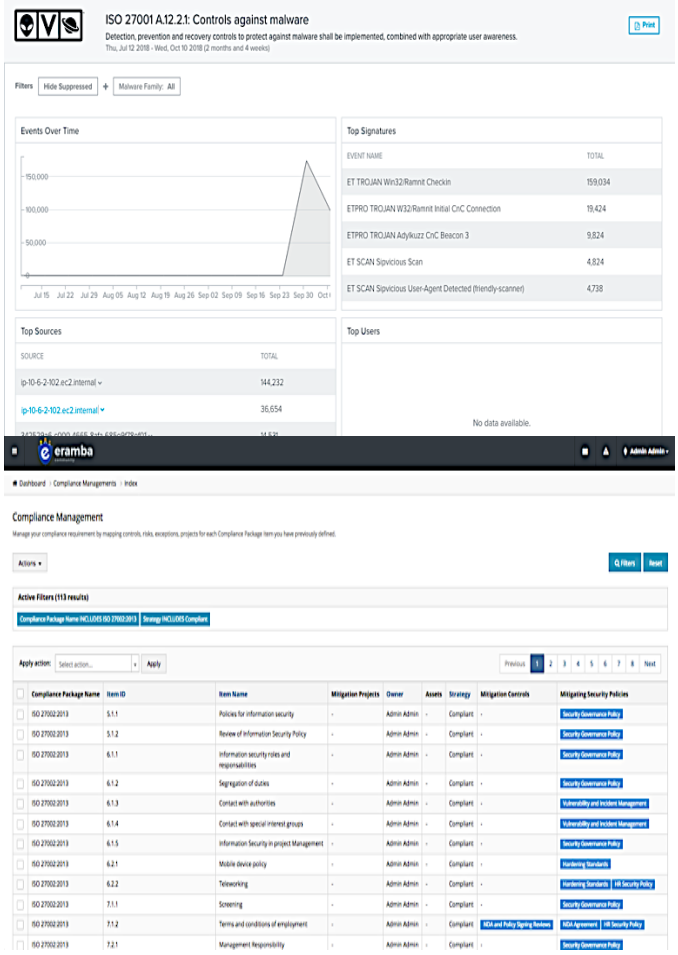


Fig. 1. Microsoft Security Compliance Manager [6], Nessus Vulnerability Assessment Tool [7], Alien Vault Unified Security Management[8], Eramba Opensource GRC [9].

Thru a survey of some common and popular tools, this research observed that most of these tools focus heavily on the technical compliance requirements and thus generate compliance reports based on templates that still require human interpretation and translation to actual compliance requirements in order to provide an actual compliance status report. As an example, one tool can generate a report that contains the top sources of malware infections, top signatures detected, and malware events over time as a report for to comply with the statement from ISO27001 stating "Detection, prevention and recovery controls to protect against malware shall be implemented, combined with appropriate user awareness." However, details like prevention and recovery are not clearly seen in the report and user awareness is not considered as well [8]. That being said, assuming that the organization is already compliant with the statement of ISO 27001, when ask the question, is the organization also compliant with Payment Card Industry Data Security Standard (PCIDSS) where the statement is "Ensure that anti-virus programs are capable of detecting, removing, and protecting against all known types of malicious software.", it is also not readily determined by the system if the two requirements are equivalent, or just overlap, or which part of the requirement is still non-compliant.

III. Our Approach

Existing security and compliance ontologies [2][3][4] model domain concepts such as controls in its ideal end state and concepts usually refer to tangible objects that should already exist. However, in considering the compliance documents statements, it can be seen that these documents do not only state the end state for specific requirements, they also state concepts like actions (e.g. "Ensure", "should be implemented") that needs to be performed and conditions (e.g. "actively running") that have to be met or maintained by the specific requirements. As such, enhancements to the existing ontologies were introduced [10] in order to model and capture the statements from the perspective of a "requirement" rather than just a "control". This perspective was taken in order to also facilitate mapping and translation between compliance documents.

TABLE I. Compliance Document Structure

Document	Structure
CoBIT 5	Stakeholder Drivers -> Stakeholder Needs -> Enterprise Goals -> IT Related Goals -> Enabler Goals -> Process Goal -> Process Practices -> Process Activities
ISO 27002	Security Control Clause/ Domains -> Security Categories -> Controls -> Objective, Implementation Guidance
PCIDSS	Requirements -> Testing Procedure -> Guidance

A. Compliance Documents Conceptual Overlaps

One of the goals of this research is to allow for a more granular understanding and assessment of the compliance level of an organization. In considering the documents to be used in the research, Table I illustrates the inherent document structure of each of the documents that are to be used in this research. To improve on the granularity aspect, the research focuses on the leaf nodes of the structure of the document as the actual requirements are stipulated in this section of the document.

High level alignments and mappings are already available both for older versions of the documents as well as the current versions. However, having the mapping stop at a higher section level of the document loses the needed details to actually determine how similar or different the requirements are to each other as well as the potential gaps among these requirements. Table IV shows the level with which current alignments and mappings are being developed and published. Although such guides provide a good starting point for mapping and determining related requirements, it lacks the level of detail to clearly describe the differences and redundancies. One advantage in looking at standards documents is that each requirement is stated in an enumerated structure such that it is possible to perform comparison without having to locate relevant sections as well as remove unneeded and unwanted text. The same cannot be said on other forms of documents such as the traditional book or news articles as well as corporate governance documents that are more descriptive based rather than being itemized [11].

To model compliance as per the compliance documents, this research took the perspective of a compliance auditor that is checking based on the statements or requirements of the compliance documents. Table II shows the definitions of the concepts from the actual document while Table III shows sample excerpts from these documents. It can be seen from both tables that there are conceptual overlaps both in definition and in the actual requirements as stated in the documents. Given the overlaps, several attempts and efforts have been performed to map and align these documents in order to enforce compliance. However, current efforts are done manually and independent of any systems that can monitor compliance. Compliance and audit practices are also currently performed at an individual standard or requirement basis as there is no definitive granular mapping that can illustrate exact overlaps and gaps among different compliance requirements.

TABLE II. Definition of the basic Concepts used as basis for Extraction

Document and Concept	Definition
CoBIT 5 - Activity	The main action taken to operate a process. Describe a set of necessary and sufficient action-oriented implementation steps to achieve a Governance ***Practice*** or Management Practice (ISACA 2012)
ISO 27002 - Control	The means of managing risk, including policies, procedures, guidelines, ***practices*** or organizational structures, which can be administrative, technical, management or legal nature. (ISACA 2012)
PCIDSS - Requirement	Compliance validation basis, considered in-place if ***controls*** are implemented or scheduled to be implemented. (Payment Card Industry 2016)

TABLE III. Excerpt on the Standards Documents Referring to Malware Protection

Document	Excerpt
CoBIT 5 - Activity	Implement and maintain preventive, detective and corrective measures in place (especially up-to-date security patches and virus control) across the enterprise to protect information systems and technology from malware (e.g., viruses, worms, spyware, spam).
ISO 27002 - Control	Detection, prevention and recovery controls to protect against malware should be implemented, combined with appropriate user awareness.
PCIDSS - Requirement	Deploy anti-virus software on all systems commonly affected by malicious software (particularly personal computers and servers).

Existing works such as [11] map compliance document concepts at a section level (see Table IV) which leads to loss of detail as well as the inability to develop a system that can automate compliance status reporting with respect to actual compliance requirement statements. Our research looks at the lower level (see Table V) in modeling the concepts for the compliance ontology in an attempt to be implementable by a system.

TABLE IV. RECENTLY PUBLISHED MAPPING DOCUMENTS THAT SHOW LEVELS OF MAPPING

Document	Structure
CoBIT 5	Stakeholder Drivers -> Stakeholder Needs -> Enterprise Goals -> IT Related Goals -> Enabler Goals -> Process Goal -> ***Process Practices*** -> Process Activities
ISO 27002	Security Control Clause/ Domains -> Security Categories -> ***Controls*** -> Objective, Implementation Guidance
PCIDSS	***Requirements*** -> Testing Procedure -> Guidance

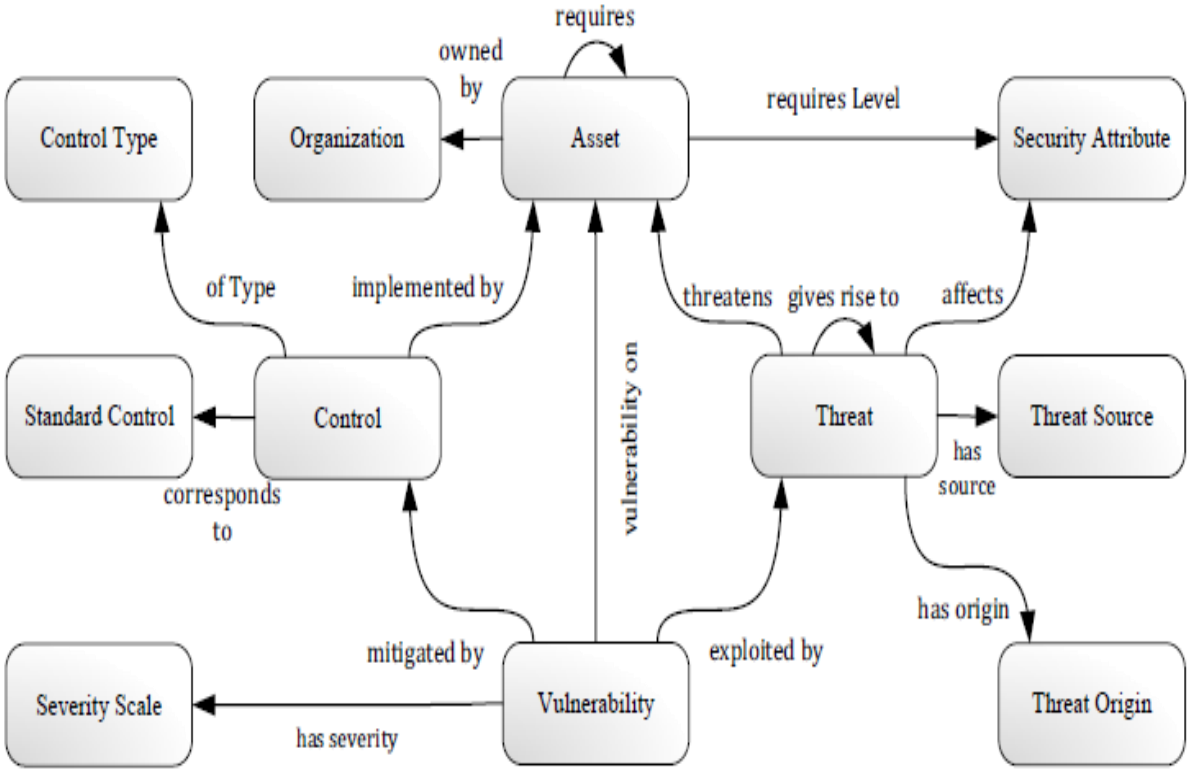
TABLE V. COMPLIANCE DOCUMENT LEVEL USED FOR MODELING

Document	Structure
CoBIT 5	Stakeholder Drivers -> Stakeholder Needs -> Enterprise Goals -> IT Related Goals -> Enabler Goals -> Process Goal -> Process Practices -> ***Process Activities***
ISO 27002	Security Control Clause/ Domains -> Security Categories -> Controls -> ***Objective, Implementation Guidance***
PCIDSS	***Requirements*** -> Testing Procedure -> Guidance

B. Compliance Ontology Enhancement

Multiple efforts in defining ontologies that can be used in security and compliance activities have been conducted and defined. As can be seen in Fig. 2, high-level conceptual modelling of information security and compliance have been developed and defined [12] [13] that shows the corresponding relationships of the different concepts involved. However, these models are not linked to the compliance documents and model mostly based on technical aspects or general concepts of information security or compliance. Several concepts defined in these researches are not readily visible or available within the statements of the compliance documents. In order to link such models to the actual statements in the documents, this research focused on the concept of Assets and Controls and put aside concepts such as Threats and Vulnerabilities as although these are valid information security concepts, such concepts would normally not be found in the statements of the compliance documents. Fig. 3 shows the enhancements performed on the information security ontology.

Upon evaluating the structure of the requirements statements in the compliance documents, a Control does not stand alone in the document as the statement included Actions to be applied to Controls. Also, although a Control is implemented by an Asset, the same Control can also be applied to other objects which are also part of the Assets of the organization (e.g. Install and regularly update anti-virus software on machines containing sensitive information). Conditions (or qualifiers) are also introduced into the ontology for the concepts of Controls and Assets as the documents contain phrases such as "regularly update", "periodic evaluation", "commonly affected", and "strong encryption". A State is also added to an Asset as the documents require checking for phrases like "actively running assets". The Action can be composed of multiple actions as some documents would provide detailed guidelines on things to do, while others would be more general on their statements. The scope of an action would cover statements that contain specific ranges or domains of applicability such as perimeter or date ranges.



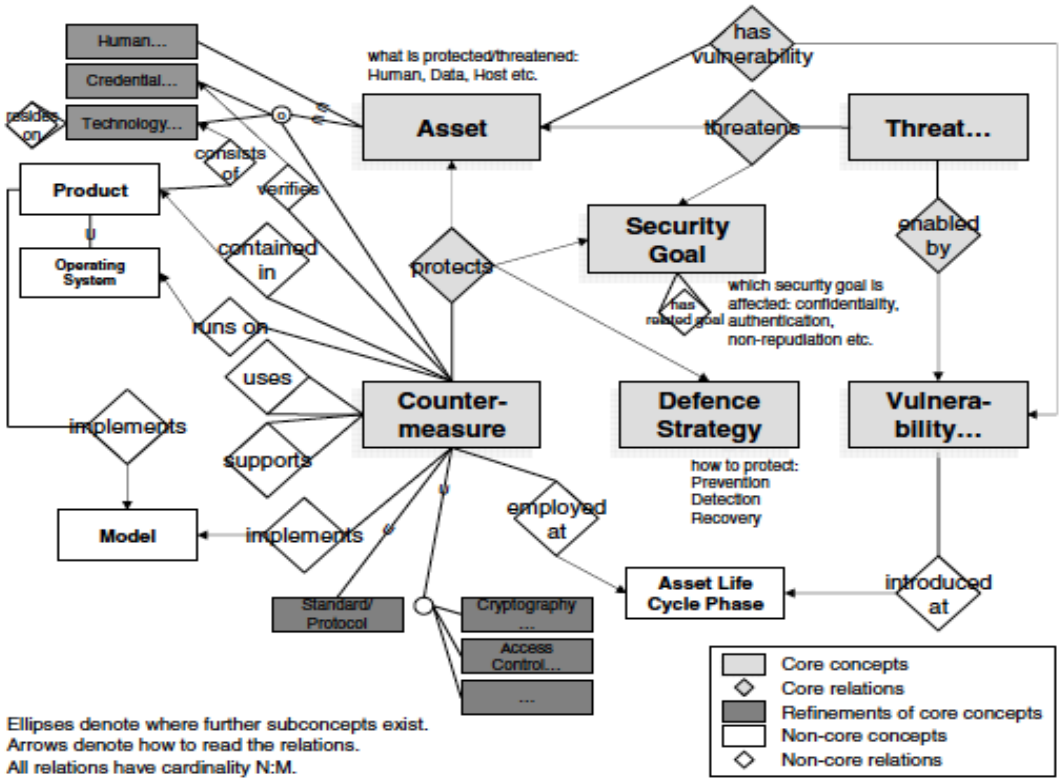


Fig. 2. Existing Informaiton Security Ontology [12][13].

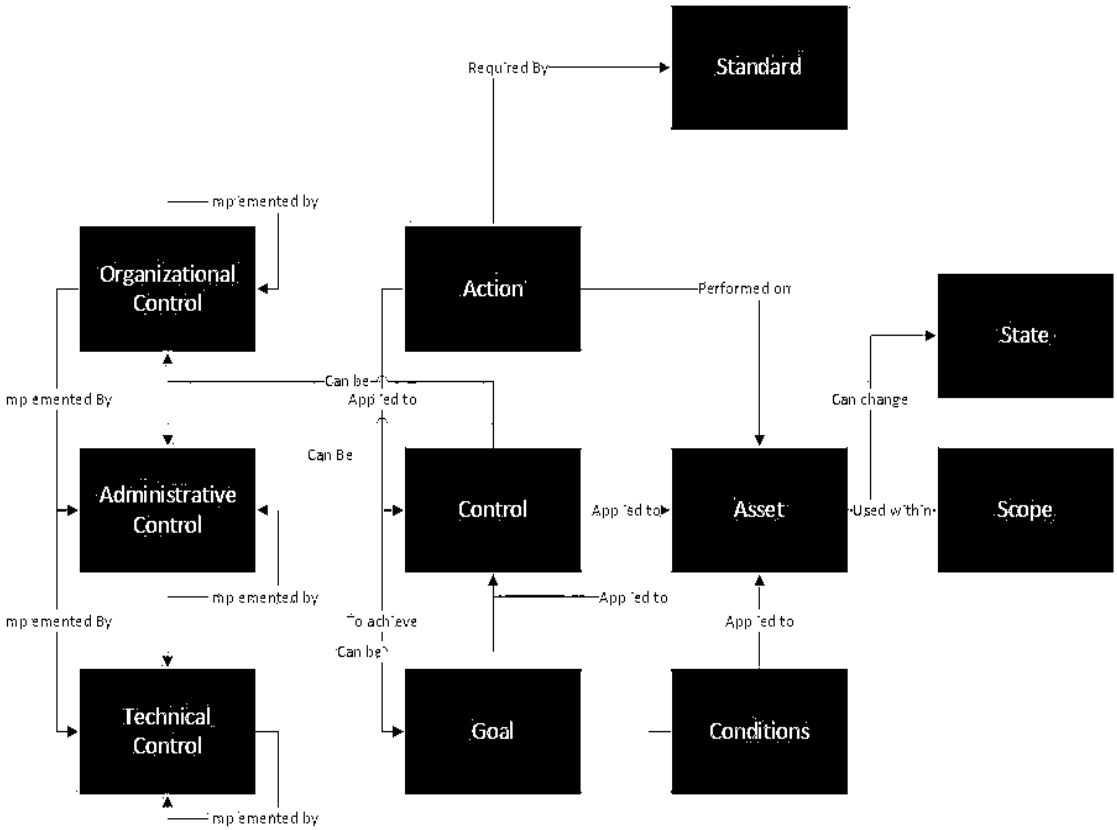


Fig. 3. Enhancements to Compliance Ontology.

In Fig. 2, the concept of the hierarchy of controls in introduced to be part of the enhanced ontology in order to tackle the complete compliance requirements rather than focusing solely on the technical aspects of controls and compliance. An example non-technical requirement would be the development of an Information Security Management Program as required by ISO 27001:2013. This requirement cannot be modelled or represented as a technical control but is an organizational requirement that is supported or implemented by administrative controls such as review process and procedures which in turn is implemented through the various technical controls deployed within an organization. The motivation behind modeling organizational and administrative controls is that although they are non-technical or manual in nature, scripts can be developed through the use of rules to automatically verify or validate if the controls are being implemented and enforced. An example case would be the administrative control requirement of an annual review process for the documents. A script can be developed with a rule to check the last date of review or revision to the document that is stored in a repository which can be used to automatically verify and validate if the requirement for annual review is being complied with or not.

C. Information Extraction in Populating the Ontology

As overlaps clearly exist among and between compliance documents, the need to reduce redundant or repetitive efforts in compliance enforcement and audit becomes paramount as the number of compliance requirements increase at an alarming rate over time [14]. In order to achieve this, automation of mapping of the compliance document to the enhanced compliance ontology is needed. To populate the enhanced ontology, Natural Language Processing tools and techniques such as information extraction approaches have been applied to construct structured information from unstructured data sources such as the compliance documents. Standards or Compliance documents state requirements as a series of imperative statements. A semantic relationship between two concepts is expressed by a verb in natural language texts [15], hence it is first necessary to identify where the verb is in the requirement statement. From there, the noun phrases and other clauses (e.g., prepositional phrases) can be processed to determine the relationships of the verb with the other entities in the sentence. For this process, Stanford CoreNLP toolkit [16] was employed. Each statement in the compliance document were pre-processed to form complete sentences and the output was given to the CoreNLP toolkit to parse and extract based on Parts of Speech (POS) generating a dependency tree [10]. The dependency tree serves as the basis for identifying what items will be used to populate the ontology (e.g. actions are populated based on verbs identified).

Upon studying the resulting annotations from Stanford CoreNLP, it was apparent that there are patterns to the type of concept to be extracted in relation to its part of speech and/or with its semantic dependency. The following lists the general patterns that can be applied to extract data and populate the ontology.

1) The ROOT is extracted as ACTION. Depending on the POS of the ROOT, the extracted data may need to be lemmatized.

a) If there is an immediate child node of this ROOT that has POS of CC and SD of CC, we then look for child nodes in the same level that has SD of CONJ to also serve as ACTION. Each ACTION extracted is stored as a separate entry in the populated ontology. Later, once the asset is determined (as stated in the following patterns), the associated ASSET and its constituents (e.g., QUANTITY and ASSET_TYPE, if applicable) are copied.

2) The first child node, where the POS is NNS and the SD is DOBJ or NSUBJPASS, is extracted as ASSET. This child node usually appears as the immediate child node or as a second-degree child node.

a) If there is an immediate child node of this ASSET that has POS of CC and SD of CC, we then identify other child nodes in the same level, with POS NNS and SD as CONJ, also as ASSET.

3) The immediate child node of the ASSET, where the POS is JJ and SD is AMOD, is the ASSET_TYPE. Similarly, the immediate child node of the ASSET, where the POS is NN and SD is COMPOUND, is extracted as ASSET_TYPE.

4) The immediate child node of the ASSET, where the POS is DT and SD is DET, is the QUANTITY.

5) The first child node of the ROOT that is a modifier of the DOBJ, e.g., NMOD that is associated with a marker like on or across, this subtree is extracted as SCOPE.

6) The first child node where the SD is ACL and has immediate child node marker to, the entire subtree is extracted as GOAL.

COBIT5	ISO:27002	PCIDSS
-> Implement/NN (root XT) -> and/CC (cc) -> maintain/VB (conj:and) -> preventive/JJ (dobj) -> ,/, (punct) -> detective/NN (conj:and) -> and/CC (cc) ->measures/NNS (conj:and) -> corrective/JJ (amod) -> detective/NN (dobj) -> measures/NNS (dobj) :::	-> implemented/ VBN (root) -> controls/NNS (nsubjpass) ->Detection/NN (compound) -> ,/, (punct) ->prevention/NN (conj:and) -> and/CC (cc) -> recovery/NN (conj:and) ->prevention/NN (compound) -> recovery/NN (compound) -> protect/VB (acl) :::	-> Deploy/VB (root) -> software/NN (dobj) -> anti-virus/JJ (amod) -> systems/NNS (nmod:on) -> on/IN (case) -> all/DT (det) -> affected/VBN (acl) -> commonly/RB (advmod) -> software/NN (nmod:by) -> by/IN (case) -> malicious/JJ (amod) -> computers/NNS (dep) :::

Fig. 4. Sample Results on using Stanford Corenlp with POS Extraction on Compliance Documents.

It should be noted that the Stanford CoreNLP tool sometimes produces erroneous tags. This phenomenon where the root identified is correct but was given a wrong POS tag appeared for quite a few samples, so far all from Control Objectives for Information and Related Technologies (COBIT)5. Similar to that in Fig. 4, the tag for "implement" was that of noun, instead of verb, even when the text is followed by a conjunction to another verb. As the assumption is that statements are written in correct English grammar, we can resolve this by comparing (and aligning) the parts of speech of both conjuncts. One possible cause for this error can be attributed to the actual structure of the document and a way to resolve this is to perform a pre-processing step to first split statements to individual goals and individual assets to, not only provide a more accurate result, but also to granularly store and perform better inferences on data later. That is, a pre-processor may first split the said COBIT5 activity into the statements in Table VI.

TABLE VI. RESTRUCTURED COBIT5 DOCUMENT FORUSE IN STANFORD CORENLP POS EXTRACTION

Implement preventive measures in place … across the enterprise to protect information systems and technology from malware.
Implement detective measures in place … across the enterprise to protect information systems and technology from malware.
Implement corrective measures in place … across the enterprise to protect information systems and technology from malware.
Maintain preventive measures in place … across the enterprise to protect information systems and technology from malware.
Maintain detective measures in place … across the enterprise to protect information systems and technology from malware.
Maintain corrective measures in place … across the enterprise to protect information systems and technology from malware.
-> data/NNS (root) -> Record/NNP (compound) -> events/NNS (nmod:on) -> on/IN (case) -> risk/NN (compound) -> caused/VBN (acl:relcl) -> that/WDT (nsubj) -> have/VBP (aux) -> or/CC (cc) -> cause/VB (conj:or) -> that/WDT (nsubj) -> may/MD (aux) -> impacts/NNS (dobj) -> IT/PRP (nmod:to) -> to/TO (case) -> enablement/NN (dobj) -> benefit/value/NN (compound)

Fig. 5. Sample Result Showing Incorrect ROOT Node.

However, incorrect tags can be more of a problem should the root identified be incorrect, as such affecting the dependency tree as well. One such example is another activity in COBIT stating: "Record data on risk events that have caused or may cause impacts to IT benefit/value enablement, IT programme and project delivery, and/or IT operations and service delivery." Fig. 5 shows an excerpt of the resulting annotation by Stanford CoreNLP. A possible resolution to this phenomenon is currently still under research.

D. Linking Compliance Documents to Verification Automation Scripts

In order to complete the link between compliance requirements and implementation verification as well as provide the ability to support continuous compliance monitoring and process audit [17], there is a need to link the populated ontology and its related compliance document to actual technical verification tools such as audit scripts in order to provide real-time compliance status feedback (see Fig. 6). There is a need to map the audit scripts to the compliance documents in order automate compliance monitoring. Existing tools that monitor compliance map to industry practices which makes them unable to directly show areas of compliance and deficiencies with respect to compliance requirements (e.g. "unsupported installation: Nessus Report" is not linked or mapped to "establishing a formal policy prohibiting the use of unauthorized software: ISO 27002:2013 section 12.2.1a" and even if its mapped, the requirement of "establishing a formal policy" cannot be seen in the initial report as the policy itself if defined is located in a different system (document management system) that is not usually part of the compliance monitoring tools.

Scripting through the use of Windows Powershell for the proof-of-concept (see Fig. 7 and Fig. 8) was implemented rather than the use of solutions such as BPMN [5] as there is a varied set of tools and controls that need to be monitored for compliance and not all tools would support BPMN. The requirement is for the script to be as atomic as possible in order for it to be reusable (e.g. 1 script for checking antivirus deployments with input parameters such as list of IPs that needs have the antivirus deployed as stated in the compliance document). In doing so, having atomic scripts and mapping it to compliance documents can also aid in the mapping and translation of requirements through the enhanced ontology by providing a common vocabulary such as common controls that was previously unavailable. Heuristics can also be incorporated to improve the mapping and translation between different compliance requirements through the enhanced ontology. Scripts mapped to similar sections or phrases within a compliance document with similar customization parameters can be used to identify overlapping requirements in multiple compliance documents as well as validation of the mapping of the ontology between documents. Relationships of controls and compliance documents such as subsumption of requirements can also be inferred by analyzing the similarity in scripts and customization parameters.

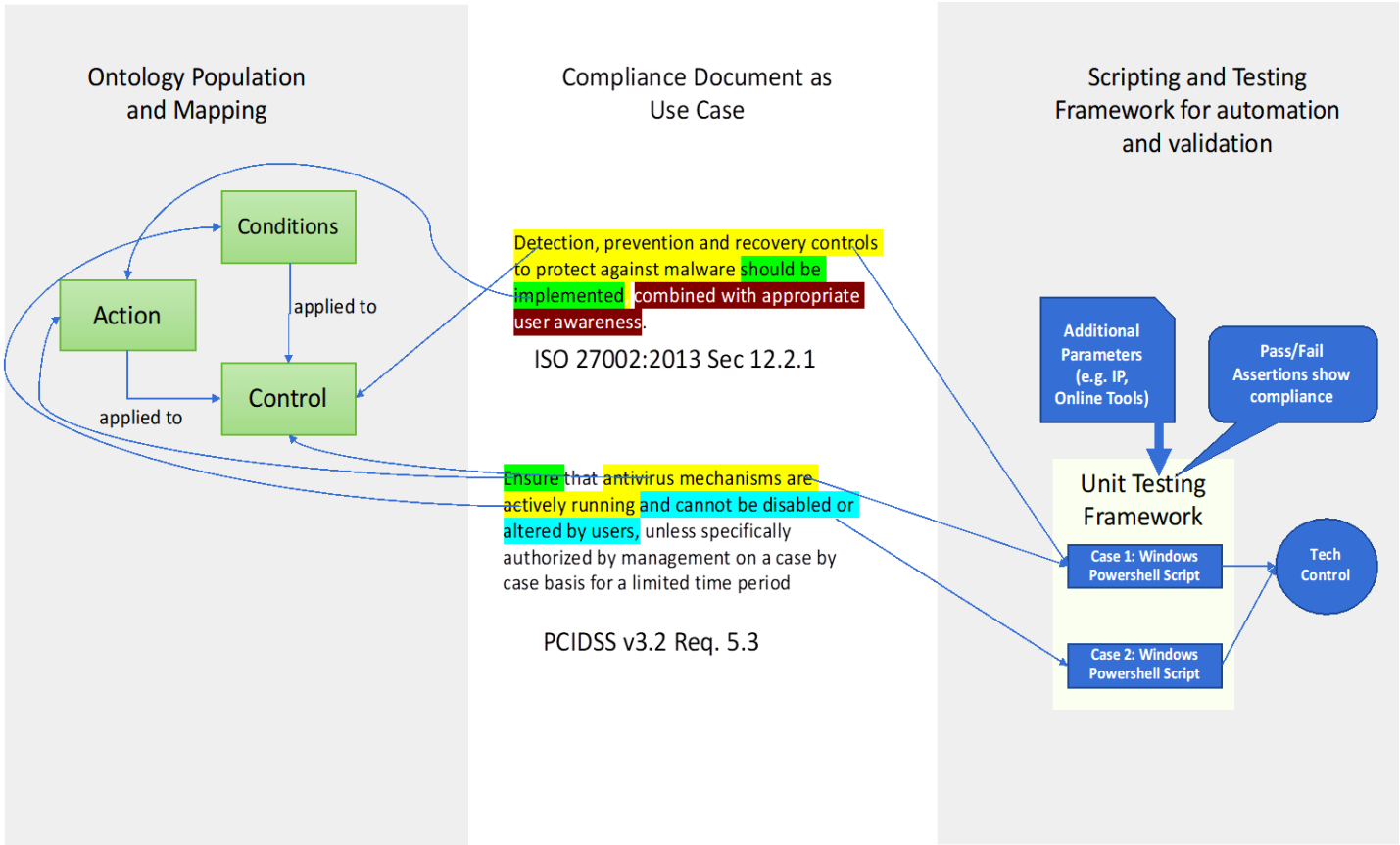


Fig. 6. Mapping and Population of Compliance Ontology based on Compliance Document and Linking it to a Testing Framework for Automation and Validation.

```
[XML]$cn = Get-Content C:\scripts\report.xml

$UserAccountArray = New-Object string[] 10
$PermissionArray = New-Object string[] 10

clear
For ($counter = 0; $counter -ne
$cn.Rsop.ComputerResults.ExtensionData[3].Extension.MsiApplication.SecurityDescriptor.Permissions.TrusteePermissions.Count; $counter++){

$UserAccountArray[$counter] = $cn.Rsop.ComputerResults.ExtensionData[3].Extension.MsiApplication.SecurityDescriptor.Permissions.TrusteePe
rmissions[$Counter].Trustee.Name.'#text'
$PermissionArray[$counter] = $cn.Rsop.ComputerResults.ExtensionData[3].Extension.MsiApplication.SecurityDescriptor.Permissions.TrusteePer
missions[$Counter].Standard.SoftwareInstallationGroupedAccessEnum
$UserAccountArray[$counter]
$PermissionArray[$counter]
```

Fig. 7. Sample Windows Powershell Script for Getting the Access Rights of the user.

```
[XML]$cn = Get-Content C:\scripts\report.xml

Function Get-PasswordSettings {

$counter = $cn.Rsop.ComputerResults.ExtensionData[7].Extension.Account.Count
$NameArray = New-Object string[] 10
$ConfiguredValueArray = New-Object string[] 10

clear

For ($x = 0; $x -ne $counter; $x++){

$NameArray[$x] = $cn.Rsop.ComputerResults.ExtensionData[7].Extension.Account[$x].Name
$ConfiguredValueArray[$x] = $cn.Rsop.ComputerResults.ExtensionData[7].Extension.Account[$x].SettingNumber
$NameArray[$x]
$ConfiguredValueArray[$x]

}

$NameArray.Count
$ConfiguredValueArray.Count


}

Get-PasswordSettings
```

Fig. 8. Sample Windows Powershell Script for Checking Password Settings.

```
public class ComplianceTest {

    public ComplianceTest() {}

    @BeforeClass
    public static void setUpClass() {}

    @AfterClass
    public static void tearDownClass() {}

    @Before
    public void setUp() {}

    @After
    public void tearDown() {
    }

    @Test
    public void testAdd() {...10 lines }

    @Test
    public void checkUserAccess(){
        try {
            Runtime runtime = Runtime.getRuntime();
            Process proc = runtime.exec("powershell C:\\GetUserAccess.ps1");
            proc.getOutputStream().close();
            InputStream is = proc.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);
            BufferedReader reader = new BufferedReader(isr);
            String line;
            while ((line = reader.readLine()) != null)
            {
                System.out.println(line);
            }
            reader.close();
            proc.getOutputStream().close();
        } catch (IOException ex) {
            Logger.getLogger(CalculatorTest.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

Fig. 9. Sample Integration of Windows Powershell Compliance Test Scripts to a unit Testing Framework to Model Compliance Requirements as Test Cases[18].

In order to fully support compliance requirements statements, the audit scripts are formalized through the use of test cases in testing frameworks used in software quality assurance to allow for customization and configuration based on audit practices (see Fig. 9) [11]. The introduction of the concept of test cases allows for a formal mapping as the compliance requirements can now be modelled also as use cases similar to what is being used in secure software development lifecycles (see Fig. 6). Complexities such as scope of control and non-technical controls can also be supported by audit scripts with the assumption that additional input may be given, or artifacts can be found in digital storage. In the case of PCIDSS v3.2 the statement "Deploy antivirus software on all systems commonly affected by malicious software (particularly personal computers and servers)" contains a scope of "all systems commonly affected". In a manual audit, the audit team would ask for network diagrams and segments to determine scope, for the audit scripts, the scope can be given as a set of IP addresses that meet the criteria. For the case of document requirements such as a formal policy, audit scripts can also be developed to check document repositories for the existence and updates to policies of the organization.

Aside from checking the existence of completed compliance implementations, the research also aims to support partial compliance and evidence mapping by linking compliance requirements to tickets within a project management tool so as to show the status of an activity or project for compliance for monitoring purposes. If the activity or project has been completed, the link can also serve as a means to derive documentary evidences for compliance report generation as needed by compliance audits.

IV. Results and Conclusions

Validation and testing for the research is currently done by parts, namely, the ontology population through information extraction, the integration of the enhanced ontology to an existing GRC system, and the validation of audit scripts with respect to compliance statements. In ontology population, the rules were manually evaluated against the resulting Parts of Speech (POS) tagging of CoreNLP and the number of terms matching the defined rules to determine the verb or action to be performed. For the integration with an existing GRC system, the results were evaluated based on the ability of the resulting extraction to conform with the process of data population or importing of data to the GRC system. Finally, in terms of the validation of audit scripts with respect to compliance statements, configurable scripts were developed that allow parameters to be given to describe the actual compliance statement requirements. These scripts and their corresponding parameters are then mapped to the parts or phrases within the compliance statement in order to determine which requirement is actually being verified by the script.

Current results of the information extraction and ontology population when tested on 356 compliance requirements statements using CoreNLP showed a 69.38% (247 statements) accuracy in action/verb identification. After additional pre-processing and testing on other similar compliance documents, the result of proper verb/action identification is currently at an average of 79% with a range of 70% to 91% as detailed in Table VII and VIII. The average result of each compliance document is used to get the average of the accuracy level across different compliance documents that were used in this research. However, the test only refers to the ability to determine the proper verb or action needed and does not yet consider the identification, extraction, and population of the other concepts as needed in the ontology. It also showed that there is potential misidentification of the action as the word can have multiple meanings [10]. Challenges currently exist in identifying the remaining concepts of the enhanced ontology from compliance documents for ontology population due to the lack of existing taxonomy or vocabularies in the domain. The ontology population is also currently mapped to an opensource GRC system (Eramba) [10] and integration to a project management system has been conceptualized through the use of hyperlinks to project tickets in order to support monitoring of partial compliance implementation.

Audit scripts have also been implemented in an atomic and customizable form in order to support the varied requirements of compliance. Scripts developed include checking antivirus deployment status, checking installed software, checking hardware inventory, and network scanning. These are initially developed as PCIDSS was the compliance document used for the basis of determining what scripts to develop. For example, the network scanning script is to be used to determine compliance to PCIDSS requirement 11.1.1 "Maintain an inventory of authorized wireless access points including a documented business justification." The same script can also be used to determine compliance for ISO 27002:2013 Section 13.1.1 Network Controls (f)(g) stating "systems on the network should be authenticated" and "systems connection to the network should be restricted" Current implementation is limited to controls that can be validated through scripting in a Windows Powershell environment.

TABLE VII. Definition of the basic Concepts used as basis for Extraction

Documents	**Requirements or Domains or Sections**	**No. of Sentences**	**No. of Sentences w/ acceptable processing by CoreNLP**	**%**
CSC CIS Requirements	20	272	202	74%
ISO 27001	14	111	87	78%
ISO 9001 2015	7	126	115	91%
PCI V3	13	297	229	77%
PCI DSS 3.2	12	357	249	70%
NIST 80053	24	553	453	82%
TOTAL	**90**	**1716**	**1335**	**79%**

TABLE VIII. DEFINITION OF THE BASIC CONCEPTS USED AS BASIS FOR EXTRACTION

Documents	Requirements or Domains or Sections	No. of Sentences	No. of Sentences w/ acceptable processing by CoreNLP	%
CSC CIS Requirements	1	10	8	80%
	2	9	9	89%
	3	19	15	84%
	4	19	14	81%
	5	13	8	77%
	6	11	9	78%
	7	17	15	80%
	8	11	7	78%
	9	8	7	79%
	10	9	7	79%
	11	12	11	80%
	12	20	14	78%
	13	16	11	78%
	14	10	7	77%
	15	16	9	76%
	16	23	16	75%
	17	10	5	74%
	18	16	13	74%
	19	10	9	75%
	20	13	8	74%
TOTAL	20	272	202	74%
ISO 27001	5	2	2	100%
	6	7	6	89%
	7	6	3	73%
	8	10	10	84%
	9	13	9	79%
	10	2	2	80%
	11	15	13	82%
	12	14	10	80%
	13	7	4	78%
	14	13	9	76%
	15	4	3	76%
	16	7	6	77%
	17	4	3	77%
	18	7	7	78%
TOTAL	14	111	87	78%
ISO 9001 2015	4	13	11	85%
	5	6	6	89%
	6	8	8	93%
	7	28	24	89%
	8	49	46	91%
	9	15	13	91%
	10	7	7	91%
TOTAL	7	126	115	91%
PCI V3	1	25	20	80%
	2	18	15	81%
	3	28	24	83%
	4	7	3	79%
	5	6	4	79%
	6	36	22	73%
	7	11	8	73%
	8	31	25	75%
	9	41	35	77%
	10	26	17	76%
	11	21	13	74%
	12	42	39	77%
	A	5	4	77%
TOTAL	13	297	229	77%
PCI DSS 3.2	1	24	15	63%
	2	19	16	72%
	3	37	28	74%
	4	6	5	74%
	5	8	5	73%
	6	37	17	66%
	7	11	6	65%
	8	37	26	66%
	9	41	24	65%
	10	49	47	70%
	11	30	21	70%
	12	58	39	70%
TOTAL	12	357	249	70%
NIST 80053	1	43	35	81%
	2	22	19	83%
	3	14	13	85%
	4	22	15	81%
	5	28	24	82%
	6	25	19	81%
	7	29	27	83%
	8	9	6	82%
	9	11	9	82%
	10	37	33	83%
	11	13	13	84%
	12	28	24	84%
	13	19	15	84%
	14	9	6	83%
	15	29	22	83%
	16	23	21	83%
	17	40	37	84%
	18	31	21	83%
	19	21	15	83%
	20	34	27	82%
	21	18	13	82%
	22	4	4	82%
	23	29	23	82%
	24	15	12	82%
TOTAL	24	553	453	82%

Although the continuous compliance monitoring framework has been identified and defined, the research still needs to validate if the use of heuristics from the perspective of audit scripts can help build the taxonomy or vocabulary needed to improve the population of the enhanced compliance ontology which in turn will improve the mapping and translation of compliance requirements with the eventual goal of reducing compliance efforts and activities in an ever growing complexity of compliance requirements.

ACKNOWLEDGMENT

This research is partly funded by the University Research Coordination Office (URCO) of De La Salle University Manila and was conducted with the assistance of research assistant Julie Ann Salido.

REFERENCES

[1] Falcione A., McKillop, J. 2016. "PwC State of Compliance Study 2016 Laying a strategic foundation for strong compliance risk management" in https://www.pwc.com/us/en/risk-assurance/state-of-compliance-study/assets/state-of-compliance-study-2016.pdf

[2] N. S. Abdullah, M. Indulska, and S. Sadiq. 2016. "Compliance management ontology --- a shared conceptualization for research and practice in compliance management." Information Systems Frontiers 18, 5 (October 2016), 995-1020. DOI: https://doi.org/10.1007/s10796-016-9631-4)

[3] Fenz, S., Ekelhart, A.: 2009. "Formalizing information security knowledge." in ASIACCS 2009: Proceedings of the 2009 ACM symposium on Information, computer and communications security. ACM, New York

[4] Schmidt, R., Bartsch, C., Oberhauser, R., 2007 "Ontology-based representation of compliance requirements for service processes" SBPM 2007 Semantic Business Process and Product Lifecycle Management. http://ceur-ws.org/Vol-251/paper4.pdf

[5] Sunkle S., Kholkar D., and Kulkarni V. 2016 "Toward Better Mapping between Regulations and Operational Details of Enterprises Using Vocabularies and Semantic Similarity" Complex Systems Informatics and Modeling Quarterly CSIMQ, Issue 5, December 2015 / January 2016, Pages 39-60

[6] Microsoft Security Compliance Management (retrieved October 2018) https://www.microsoft.com/en-us/download/details.aspx?id=53353

[7] Nessus Vulnerability Assessment Tool (retrieved October 2018) https://www.tenable.com/products/nessus/nessus-professional

[8] Alienvault Unified Security Management (retrieved October 2018) https://www.alienvault.com/products

[9] Eramba Opensource Governance Risk and Compliance System (retrieved October 2018) http://www.eramba.org/

[10] D. C. Cheng and N. R. Lim-Cheng, "An ontology based framework to support multi-standard compliance for an enterprise," 2017 International Conference on Research and Innovation in Information Systems (ICRIIS), Langkawi, 2017, pp. 1-6.

[11] D. Lacey, "A Practical Guide To The Payment Card Industry Data Security Standard (PCI DSS)" ISACA 2015 ISBN 978-1-60420-586-2

[12] F., S., Ekelhart, A.: 2009. "Formalizing information security knowledge." in ASIACCS 2009: Proceedings of the 2009 ACM symposium on Information, computer and communications security. ACM, New York

[13] H., Almut & S., Nahid & D., Claudiu. (2007). An Ontology of Information Security. IJISP. 1. 1-23. 10.4018/jisp.2007100101.

[14] J. Verver (2017). Top 8 Better Practices In Compliance Management. Retrieved from http://www.acl.com/pdfs/ebook-top-8-better-practices-in-compliance-management.pdf

[15] Mercier-Laurent, E. and D. Leake, D. 2008. "Intelligent Information Processing IV" In Proceedings of the 5th IFIP International Conference on Intelligent Information Processing. Volume 288. Springer Science & Business Media.

[16] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. 2014. "The Stanford CoreNLP Natural Language Processing Toolkit." Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.

[17] N. Subhani and R. D. Kent, "Continuous process auditing (CPA): An audit rule ontology based approach to audit-as-a-service," 2015 Annual IEEE Systems Conference (SysCon) Proceedings, Vancouver, BC, 2015, pp. 832-838.

[18] Y. H. Tung, S. C. Lo, J. F. Shih and H. F. Lin, "An integrated security testing framework for Secure Software Development Life Cycle," 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Kanazawa, 2016, p.