Optimus: A Framework of Vulnerabilities, Attacks, Defenses and SLA Ontologies

CHEN-YU LEE, PATRICK KAMONGI, KRISHNA M. KAVI
University of North Texas
MAHADEVAN GOMATHISANKARAN
Microsoft

Maintaining security and privacy in the Cloud is a complex task. The task is made even more challenging as the number of vulnerabilities associated with the cloud infrastructure and applications are increasing very rapidly. Understanding the security service level agreements (SSLAs) and privacy policies offered by the service and infrastructure providers is critical for consumers to assess the risks of the Cloud before they consider migrating their IT operations to the Cloud. To address these concerns related to the assessment of security and privacy risks in the Cloud, we have developed a framework that relies on ontologies that obtain different objects, policies and vulnerabilities. Our framework is called Optimus and uses three related ontologies: the vulnerability knowledge base (OKB) and ontologies for representing security SLAs (SSLA). Our framework can be used to assess the risks associated with cloud services and system configurations using our vulnerability ontologies. The risk assessment may be useful to both the provider and consumer of the cloud services. Our ontologies for SSLAs can be used to understand the security agreements of a provider, to negotiate desired security levels, and to audit the compliance of a provider with respect to federal regulations (such as HIPAA). In this paper, we describe our Optimus framework and provide some examples of its application.

Keywords: Ontology, Vulnerability, Service level agreement, SLA, SSLA

1. INTRODUCTION

The use of Cloud computing services is becoming the preferred choice for many businesses because of their economic advantages over in-house IT operations. Cloud computing makes it easier to streamline IT processes and reduce expenditures on technology infrastructure. In addition, it provides economies of scale and an effective way to monitor project budgets since the business only pays for the amount of computing and services used. The main concern preventing some companies from adopting Cloud computing is the risk of privacy and security. The levels of security provided by a Cloud provider through its service level agreement (SSLA) are also very difficult to understand. Quantitatively comparing the SSLAs of different providers is even more challenging. Finally, as the number of "detected" vulnerabilities associated with Cloud services, infrastructure, and social engineering attacks is growing at an exponential rate [Zhou and Pei 2008], the ability to assess the security and privacy risk of a Cloud service and infrastructure can aid in determining the appropriate SSLAs required.

To address these concerns, we have developed a framework that relies on ontologies that relate different objects, policies, and vulnerabilities. *Ontology* is a formal framework for representing knowledge. This framework names and defines the types, properties, and interrelationships of the entities in a domain of discourse. Similar to our previous work [Lee et al. 2014], we use ontologies to conceptualize our security information for the following reasons:

—The defined ontologies are machine readable vocabularies that are specified with enough precision to allow differing terms to be precisely related.

—The security ontologies could be used by analysts/developers, databases, and applications that need to share domain information.

This research is supported in part by the NSF Netcentric and Cloud Software and Systems Industry/University Cooperative Research Center and NSF award 1128344. Author's addresses: C.Y. Lee, P. Kamongi and K.M. Kavi, Computer Science and Engineering, University of North Texas; M. Gomathisankaran, Microsoft.

—Since machines can read and interpret our ontologies, we can instantiate them automatically enabling us to seamlessly and effortlessly generate rich and powerful security knowledge bases/representations.

—We could conduct automatic reasoning on our generated knowledge bases.

We use an ontology approach instead of taxonomies for modeling security information since an ontology provides the potential for formal logic inference based on well-defined data and knowledge bases [Wang and Guo 2009]. In general terms, both an ontology and taxonomy can represent the same knowledge domain. However, ontologies are considered to be broader and can be thought of as a number of taxonomies assembled together with more expressive and interconnective relationships added (with each taxonomy organizing a subject in a particular way). This led to our choice of using ontologies in our modeling.

In this paper, we propose a novel framework called Optimus and provide examples of its application. Our Optimus framework uses two related ontologies: the vulnerability knowledge base (OKB) [Kamongi et al. 2013] and ontologies for representing security SLAs (SSLA). Our framework is used to assess the risks associated with chosen services and system configurations. The risk assessment is useful to both the provider and the consumer of the Cloud services. Our OKB relates vulnerabilities with known attacks and defenses so that the Cloud user can evaluate alternate services and configurations or apply appropriate defenses to mitigate their risks. Our ontologies for SSLAs are used to understand the security agreements of a provider, to negotiate desired security levels, and to audit the compliance of a provider with respect to federal regulations, such as HIPAA.

The rest of this paper is organized as follows. Section 2 discusses research that is closely related to ours. The Optimus ontology framework is introduced in Section 3. The two component ontologies of Optimus are presented in Section 4 and Section 5, respectively. We discuss the application of Optimus in Section 6.

2. RELATED WORKS

2.1 Security Ontology Knowledge Bases

A Knowledge Base (KB) is a special kind of database for knowledge management. In our study, we are interested in a KB class known as Semantic Web Knowledge Base (or Ontology Knowledge Base), which is a semantic web repository of data that becomes knowledge. This Ontology Knowledge Base (OKB) is machine-readable and when visualized is very amenable to human understanding. Its architecture provides the ability to represent knowledge and facilitate its retrieval and sharing among other applications. There are a number of projects that focus on providing OKB and the open source tools to manage and generate the OKB. RDFKB is a Semantic Web Knowledge Base [McGlothlin et al. 2011]. DBpedia[1] is a crowd-sourced community effort to extract structured information from Wikipedia and make it available on the Web. Protege[2] is a free open source ontology editor and knowledge base framework.

The work of Wang and Guo [Wang and Guo 2009] on Ontology for Vulnerability Management (OVM) shows their contribution to delivering both a security ontology knowledge base and tools such as the OVM Software Assessment Tool (OSAT) [Wang et al. 2009]. The proposed OVM is an ontological approach to capturing and utilizing the fundamental concepts of information security and their relationship, retrieving vulnerability data, and reasoning about the cause and impact of vulnerabilities. Other initiatives have proposed some ontological security solutions such as Fenz's work on Ontology-based Generation focusing on IT-security metrics addressing a much needed methodology for automatically generating ISO 27001-based IT-security metrics [Fenz 2010].

Depending on the type of security problem to be addressed, it has been shown that a knowledge base that uses the ontological approach enables the security practitioner to not only retrieve data

[1] DBpedia: http://wiki.dbpedia.org/

[2] Protege Editor: http://protege.stanford.edu/

from their KBs, but also infer new knowledge. Bill and Gritzalis presented a security management framework for an arbitrary information system which builds upon knowledge-based resources, such as a security ontology providing reusable security knowledge interoperability, aggregation, and reasoning exploiting security knowledge from diverse sources [Tsoumas and Gritzalis 2006].

Today, typical IT infrastructures are run in a cloud environment. This cloud paradigm introduces new economical IT solutions, but is also vulnerable to security weaknesses that could be further exploited. A security knowledge base tailored to address cloud security is missing. To address this gap, we previously proposed a Vulnerability Assessment Framework for Cloud Computing (VULCAN) [Kamongi et al. 2013]. This framework uses a security KB tailored specifically to address the cloud vulnerability assessment challenge, but could also be extended to additional sources that cover a broad range of the security assessment spectrum.

2.2 Security Service-oriented Agreement

2.2.1 *Service Level Agreements.* Service-orientation has become a basic principle of commercial IT infrastructures including the Internet of services, cloud computing, and so on. However, to guarantee the quality of the services, it is necessary to enter the exact usage conditions into contracts that can be specified in a Service Level Agreement (SLA). An SLA is described in the Information Technology Infrastructure Library ver. 3: ”A service level management negotiates, agrees and documents appropriate IT service targets with representatives of the business, and then monitors and produces reports on the Service Providers ability to deliver the agreed level of service” [of Government Commerce 2010].

Most commercial providers, such as Amazon[3], Google[4],or Microsoft Azure[5], regulate the service scopes and the service availability as listed below.

—99.999% email processing availability

—100% antivirus filtering

—99.9% monthly uptime

—SSL/TLS and Service Side Encryption (SSE) support

—99.999999999% durability of objects over a year

—Versioning support

For Web services, SLA monitoring and enforcement become increasingly important, especially when enterprise applications and services subscribe to cloud resources on-demand. However, SLA templates written in natural language lack flexibility in different domains, different organizations, and different definitions for IT parameters. The WSLA framework [Keller and Ludwig 2003] and Web Services Agreement Specification (WS-Agreement) [Andrieux et al. 2011] are XML-based frameworks to formalize the terminology, concepts, and agreement structure used in automatic negotiation, deployment, monitoring and enforcement of SLAs.

To capture and present requirements for both provider and consumer, Modica et al. proposed a SLA ontology to present the definition of a semantic domain of knowledge for the cloud business according to the Cloud Standards Consumer Council [CSCC 2012] shown in Figure 1 [Modica et al. 2012]. Based on the knowledge base, providers would be able to customize their offers according to their business strategy, and consumers can claim the resource requests consistent with their real needs.

2.2.2 *SSLA.* Henning first proposed the concept of a Security Service Level Agreement (SSLA) to specify the requirements of security of services for an enterprise [Henning 1999]. In 2013,

[3] Amazon EC2 Service Level Agreement: http://aws.amazon. com/cn/ec2/sla/

[4] Google Cloud Storage, Google Prediction API, and Google BigQuery SLA, https://developers.google.com/storage/sla

[5] Windows Azure Storage Service Level Agreement, http:// www.microsoft.com/en-us/download/details.aspx?id=6656

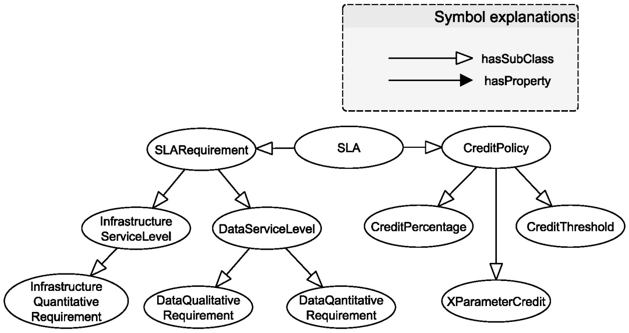


Figure. 1: Ontology for SLA

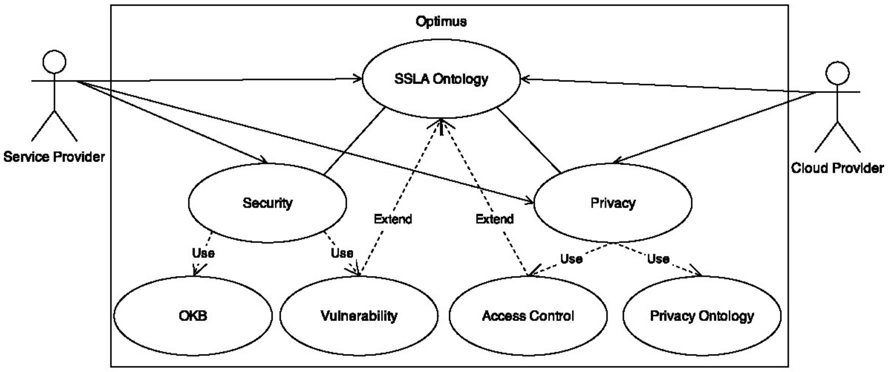


Figure. 2: The relations of Optimus framework

the terms "SSLA" and "security service-oriented agreement" were proposed by Takahashi et al. [Takahashi et al. 2013]. They proposed a non-repudiatable security service-oriented agreement mechanism that describes security requirements for users and capabilities of service providers. Rong et al. mentioned some cloud security challenges including resource location, the multi-tenancy issues, authentication and trust of acquired information, system monitoring, and cloud standards [Rong et al. 2013]. Hale et al. built an XML-based compliance vocabulary compatible with the WSLA schema [Hale and Gamble 2013].

3. OPTIMUS FRAMEWORK

Our Optimus ontology framework contains two main parts: SSLA ontology and security assessment. The SSLA ontology is modeled using the generalized SSLA covering the control domains in cloud computing illustrated in Section 5. The security assessment includes the vulnerability class extended from the SSLA ontology, and the ontology knowledge bases (OKBs) of vulnerabilities, attacks, and defenses. Each of the OKB is generated from our defined ontologies plus their instances introduced in Section 4.

4. VULNERABILITIES, ATTACKS, AND DEFENSES ONTOLOGY KNOWLEDGE BASES (OKBS) DESIGN

4.1 Design

Our proposed Ontology Knowledge Bases (OKBs) [Kamongi et al. 2013] as illustrated in Figure 3 are comprised of the main Ontology Knowledge Bases (OKBs) of Vulnerabilities, Attacks, and Defenses. Each OKB is generated from our defined ontologies plus their instances.

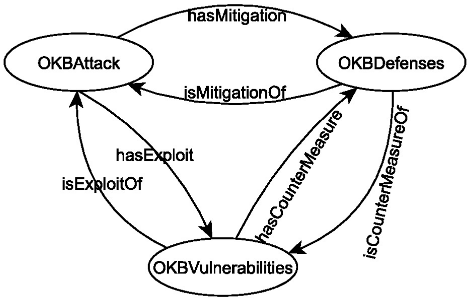


Figure. 3: The Vulnerabilities, Attacks and Defenses OKBs-High Level View

For example, Ontology Knowledge Base of Vulnerabilities (OKB-Vulnerabilities) is generated from Vulnerability, IT Products and Common Vulnerability Scoring System (CVSS) Metrics[6] ontologies. Our ontologies model the National Vulnerability Database (NVD)[7], detailed information about various reported vulnerabilities discovered for IT Products (Software or Hardware), and some indicators (CVSS metrics) of the found vulnerabilities.

As of this writing, the Ontology Knowledge Base of Attacks is modeled with the Attacks ontology. We used this model to represent and create knowledge details reported in the Exploit Database by Offensive Security[8]. We semantically built this model focusing on the open source security payload attack's code (written in languages such as C, C++, Java, Python, Ruby and so on) as well as other information about which mechanisms are used for deployment to exploit discovered vulnerabilities for relevant IT Products.

The Ontology Knowledge Base of Defenses conceptualizes the information about released patches for various NVD-reported vulnerabilities. As described above, we are modeling our security knowledge bases semantically. Semantic annotations of security information allow us to add our defined data property on how we want to represent each model, and how models interact with each other. As shown in Figure 3, we have defined object properties to express how each ontology connects with another, for example Vulnerabilities concept instances allow us to infer which potential attacks could exploit them (if found), or which countermeasures could be applied to the affected IT Products to fix the found vulnerabilities.

4.2 Implementation

The design and implementation of our OKBs follow a developed model that represents and reports all of the essential details needed to assess vulnerability information for any given IT Product or System, along with supporting insights on their exploitability and how they can be countered. Once we have our model developed and have provisioned all necessary details, we use the Protege Web Ontology Language (OWL) Application Programming Interface (API) to implement our ontologies and instantiate them to generate OKBs. Protege is an ontology editor and knowledge base creator tool.

Within the Protege API, we make frequent use of four Java main methods. The first one is used for creating our ontology's classes `OWLNamedClass`. The second and third ones are both used to create direct class data properties `OWLDatatypeProperty` and relationships between class instances `OWLObjectProperty`. The fourth one `OWLIndividual` is used to instantiate our defined classes that in turn will generate our Ontology Knowledge Bases. Other methods are used to add some constraints over our domain of knowledge. With these basic methods from OWL, we can implement our ontologies and with a little workaround we automatically extract our instances

[6] Common Vulnerability Scoring System: http://www.first.org/cvss
[7] National Vulnerability Database: http://nvd.nist.gov/
[8] The Exploit Database: http://www.exploit-db.com/

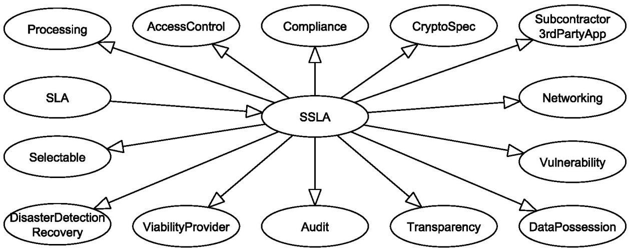


Figure. 4: All classes in SSLA

data from our repository sources (i.e., NVD, Exploit DB and so on), and generate our OKBs.

Using the OWL API allows us to create custom scripts to implement the developed models and instantiate them automatically. Note that in this work thus far, we can instantiate thousands and thousands of instances during the generation process of our OKBs. The more instances we add to our ever growing OKBs the better we can assess various classes of vulnerabilities and attack information, and determine which defense mechanisms are available to mitigate them. In addition to having these knowledge bases, we go beyond the basic search and retrieval tasks to a large-scale customizable set of queries via crafted algorithms looking to infer new knowledge from our rich and machine readable OKBs. Without this capability, discovery of new insights about a particular set of vulnerabilities, exploits or defenses from the already known ones would be very difficult if not impossible.

5. ONTOLOGY FOR SSLA

As an alternative to the traditional SLA written in natural languages, the XML-based SLA is useful for automated processing. The second part of our Optimus framework is an ontology for Security of Service Level Agreements (or SSLAs). Our SSLA ontologies extend Hales work, which is built as an XML-based compliance vocabulary [Hale and Gamble 2013]. To increase the coverage of our SSLA ontology, we take into account the challenges in covering the entire control domains specified by the Cloud Security Alliance (CSA) Cloud Control Matrix (CCM) v3 [CSA 2013].

The proposed ontology for SSLAs facilitates understanding of the security concerns in service level agreements and matches the security requirements of a consumer with the SSLAs offered by different providers. The SSLA ontology offers the following additional benefits:

—Easier understanding of the security aspects of the SLA.

—During negotiations, a consumer can compare the SLAs offered by many providers and choose the best one.

—It is easier to monitor the security requirements enforced by hosting providers, which is especially necessary for satisfying some industry compliance requirements.

5.1 Model Design

Without losing the generality of SSLAs, here we model thirteen classes including `Networking`, `Vulnerability`, `Transparency`, `DisasterDetectionRecovery`, `DataPossession`, `ViabilityOfProvider`, `CryptoSpec`, `AccessControl`, `Processing`, `Compliance`, `Audit`, `Selectable`, and `Subcontractor3rdPartyApp` as shown in Figure 4. Each class is described here.

—Networking: Figure 5 shows the networking class. This class organizes the agreements about the networking environment such as traffic isolation `TrafficIsolation` subclass; individual bandwidth for `IndividualBandwidth` subclass, which defines the guaranteed bandwidth offered to the consumer; and IP address quantity for `IPAddressQuantity` subclass, which defines the max number of IP addresses issued to the consumer.

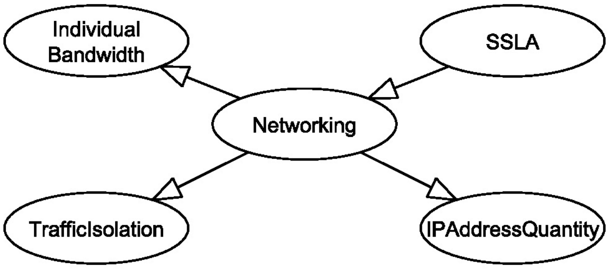


Figure. 5: Networking class in SSLA

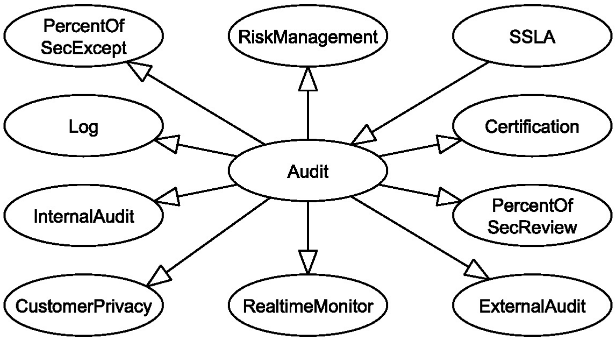


Figure. 6: Audit class in SSLA

—Vulnerability: This class defines assurance in terms of detecting and patching known vulnerabilities, including the use of malware scanners and coverage of services against attacks in `PatchPolicyComplianceRate`, `ScanFrequency`, and `ManagementCoverageRate` subclasses.

—Transparency: This class regulates the transparency of the information related to the security management processes used by the provider. The SSLA should record the responsible office that will provide the information when requested.

—Disaster detection and recovery: This involves the contingency plan and the security incident procedure that describes the regular routines of disaster detection and the recovery steps when the events occur. It also defines the data backup functions because the data is usually the most valuable asset for consumers.

—Data possession: This class rules the data storage procedures in Cloud storage, and the data verification method and frequency to ensure data usability.

—Audit: As shown in Figure 6, this class requires the architecture, management, and service of providers to be audited by internal auditors, external auditors, and issued certificates (listed in `Certification`) to build consumer trust in the providers. `InternalAudit` and `ExternalAudit` subclasses also define their audit plans and change controls. Log is the most important evidence of behaviors of attackers, consumers, and providers. To protect the security of the log, the `Log` subclass regulates the secure storing procedures and the retention time of the log. The `RiskManagement` subclass describes the risk management and data risk assessment programs. In addition, the class outlines the real time monitoring mechanisms, the acceptable percent and types of security exceptions, security review, and the protection of consumer privacy in `RealtimeMonitor`, `PercentOfSecExcept`, `PercentOfSecReview` and `ConsumerPrivacy` subclasses.

—Subcontractor and third party application: Clarifies the rights and duties with respect to

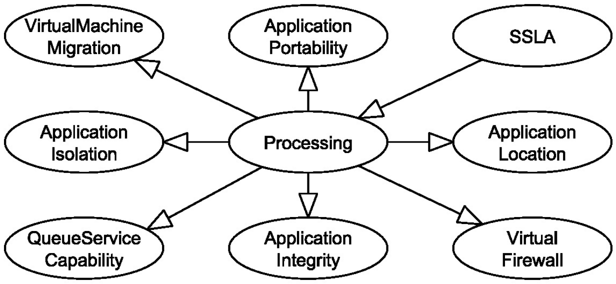


Figure. 7: Processing class in SSLA

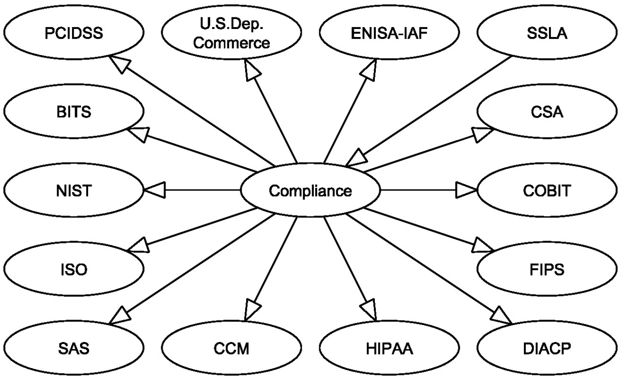


Figure. 8: Compliance class in SSLA

security of the subcontractor and the third party application providers.

—Viability of cloud provider: The system administrators of the providers' systems have the highest level of privilege. They can perform any action on any object. Thus, there is a privacy issue in defining what level of consumer data security is appropriate for a specific person and under what conditions.

—Cryptography specification: Some providers offer cryptography components optimized for their platforms. It is useful to optimize consumer data encryption while also reducing the associated computational complexity.

—Access control: Access control of the instance control panel directly impacts the security of the instance. Therefore, this class defines the access authentication, authorization, accounting schemes and rules of mobile access.

—Processing: This class covers the security demands for building a secure runtime environment in a virtual machine migration, queue service capability, virtual firewall, and the isolation, portability, location, and integrity of applications shown in Figure 7.

—Compliance: Some specific services must be certified as compliant with security and privacy standards and practices as required by law. For example, user services that involve warehousing or mining of electronic Protected Health Information (ePHI), electronic Personally Identifiable Information (ePII), or Health Insurance Portability and Accountability Act (HIPAA) data must comply with any associated federal and local standards [HIPAA 2013]. There are many subclasses defined in `Compliance` as shown in Figure 8.

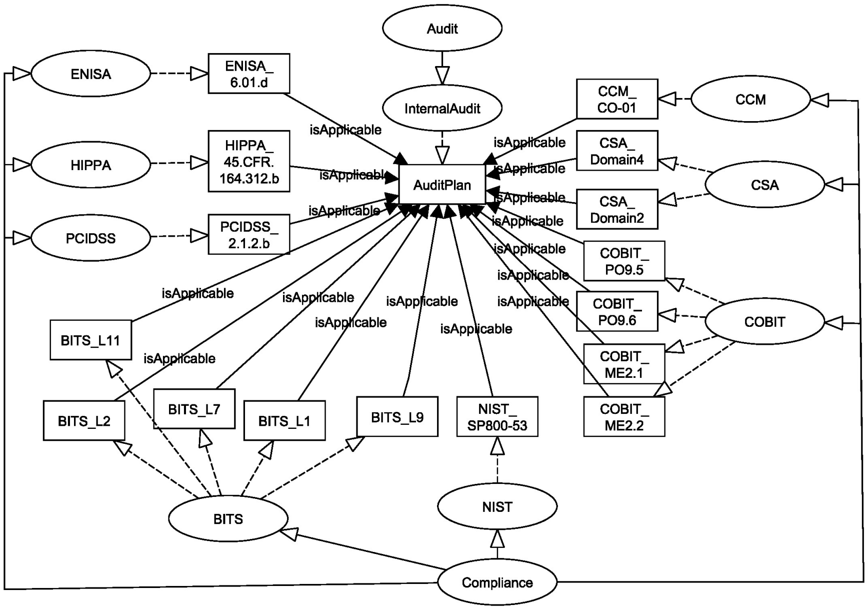


Figure. 9: Audit plan to be in compliance with the standards in CCMv3

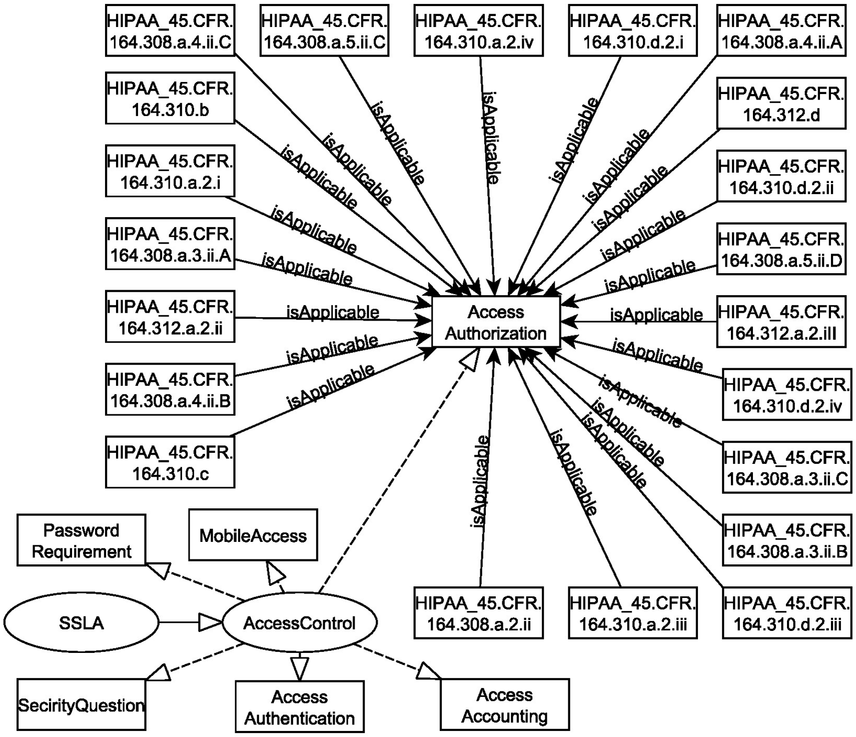

HIPAA_45.CFR. 164.308.a.4.ii.C
HIPAA_45.CFR. 164.308.a.5.ii.C
HIPAA_45.CFR. 164.310.a.2.iv
HIPAA_45.CFR. 164.310.d.2.i
HIPAA_45.CFR. 164.308.a.4.ii.A
HIPAA_45.CFR. 164.310.b
HIPAA_45.CFR. 164.312.d
HIPAA_45.CFR. 164.310.a.2.i
HIPAA_45.CFR. 164.310.d.2.ii
HIPAA_45.CFR. 164.308.a.3.ii.A
HIPAA_45.CFR. 164.308.a.5.ii.D
HIPAA_45.CFR. 164.312.a.2.ii
HIPAA_45.CFR. 164.312.a.2.iiI
HIPAA_45.CFR. 164.308.a.4.ii.B
HIPAA_45.CFR. 164.310.d.2.iv
HIPAA_45.CFR. 164.310.c
HIPAA_45.CFR. 164.308.a.3.ii.C
HIPAA_45.CFR. 164.308.a.3.ii.B
Access Authorization
isApplicable
Password Requirement
MobileAccess
SSLA
AccessControl
HIPAA_45.CFR. 164.308.a.2.ii
HIPAA_45.CFR. 164.310.a.2.iii
HIPAA_45.CFR. 164.310.d.2.iii
SecirityQuestion
Access Authentication
Access Accounting

Figure. 10: AccessControl class with HIPAA in SSLA

5.2 Implementation

Our framework can provide a method for determining whether the SSLA satisfies the specific regulations for any given compliance.

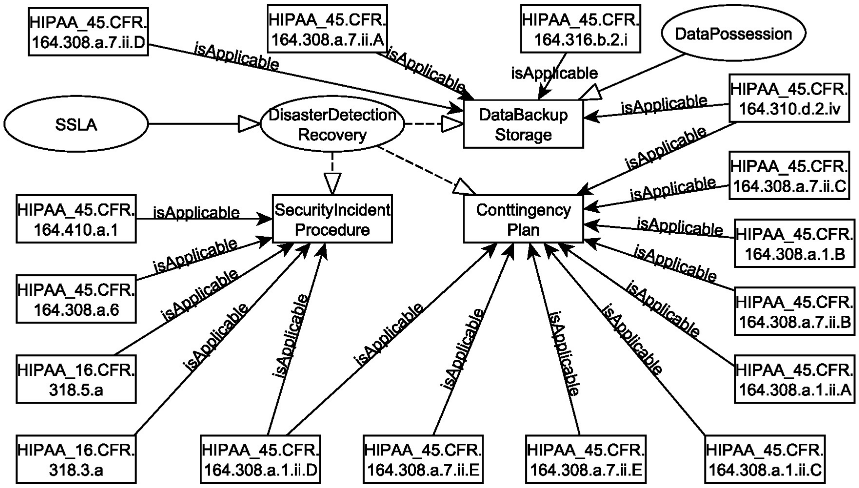


Figure. 11: Disaster Detection & Recovery class with HIPAA in SSLA

The following three examples illustrate the `AuditPlan` defined in cloud control matrix v3 [CSA 2013], the `AccessControl` regulations in the case of HIPAA compliance, and the privacy rules guaranteed by Facebook in our SSLA framework.

5.2.1 *The design of* ***`AuditPlan`****.* `AuditPlan` is a member of the `InternalAudit` class and describes the objectives, methods, and schedules of the provider's internal audit. Figure 9 shows that such an audit plan is applicable to a number of types of regulatory requirements. For HIPAA, the audit plan should regulate the requirements in HIPAA_45.CFR.164.312.b. Similarly, if the service provider is providing credit card payment systems, the audit plan has to obey PCIDSS_2.1.2.b of Payment Card Industry Data Security Standards (PCI DSS) [PCIDSS 2013].

5.2.2 *The case of HIPAA compliance.* HIPAA compliance regulates the privacy and security of the processing and storing of electronic Protected Health Information (ePHI) and electronic Personally Identifiable Information (ePII). We built the knowledge base for all HIPAA rules. We show two examples here. Figure 10 shows the 21 HIPAA rules that are applicable to `AccessAuthorization` in the `AccessControl` class. Figure 11 illustrates the rules applying to secure incident procedures, contingency plans, and data backup storage defined in the `DisasterDetectionRecovery` class.

5.2.3 *The case of Facebook privacy guarantee.* Facebook is a famous social networking service provider that provides some security and privacy guarantees for users listed in its policy page and summarized as follows:

—Facebook complies with the U.S.-EU and U.S.-Swiss Safe Harbor frameworks as set forth by the Department of Commerce regarding the collection, use, and retention of data from the European Union[9].

—Password requires at least six digits.

—Secure browsing (HTTPS) is enforced in all the connections.

—A security question helps a user to get back into his/her account if he/she can't log into the Facebook account.

—Facebook doesn't give the advertiser access to any information that identifies any user.

[9] https://www.facebook.com/full_data_use_policy#otherthings

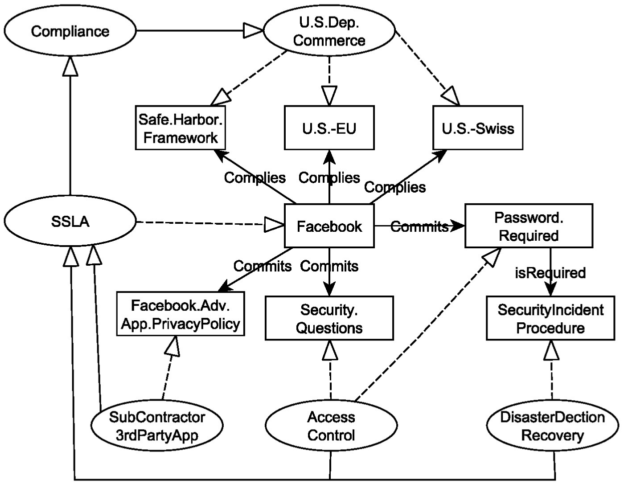


Figure. 12: Facebook privacy guarantee

The above five rules can be presented using our SSLA ontology as shown both in Figure 12 and in WS-agreement codes shown in Algorithm 1 that can be used in automatic requirement matching or negotiations.

6. DISCUSSION

6.1 Benefits for parties

This section discusses the benefits of the Optimus framework for Cloud infrastructure and service providers using HIPAA compliance as an example. It should be noted that both the service provider and the Cloud infrastructure provider are responsible for different aspects of HIPAA compliance.

—For Cloud infrastructure providers: Since an ontology is a good means for describing the knowledge, a Cloud provider could employ our SSLA ontology to present the security levels guaranteed. Additionally, the SSLA ontology also provides for negotiated agreements. With respect to HIPAA, the Cloud infrastructure provider must make sure that the Cloud environment is secure enough, at least for known vulnerabilities, and can resist known attacks. Moreover, the provider can use our OKB framework to evaluate the security risks of its resources to define the most appropriate security guarantees or price different levels of negotiated security agreements.

—For service providers: When service providers employ a Cloud environment, they can utilize our SSLA ontology framework to negotiate better levels of security guarantees from the infrastructure provider. Additionally, the service provider can use our framework to understand the compliance issues pertaining to the services they provide. With this information regarding their compliance responsibilities, the service provider can utilize our OKB (linked by the `HasExploit` property of the SSLA ontology to the OKB ontology) to identify the vulnerabilities that may be encountered by their services, including which configurations have the vulnerabilities, if a better configuration mitigates the risks. This information would allow the service provider to request a specific infrastructure configuration and negotiate issues related to monitoring for vulnerabilities, attacks and patches from the infrastructure provider.

With respect to HIPAA, maintaining the privacy of certain information is critical to any health care service providers, including email providers, insurance providers, and associated infrastructure providers like network and database providers. As mHealth applications are becoming available for use with smart phones, the mHealth applications must understand the privacy

Algorithm 1 WS-agreement codes for Facebook privacy guarantee

```
<?xml version=”1.0” encoding=”UTF-8”?>
<wsag:AgreementOffer
...
<wsag:Name>Offer1</wsag:Name>
<wsag:Context/>
<wsag:Terms>
  <wsag:All>
    <wsag:ServiceDescriptionTerm
      wsag:Name=”CompliantStandardUS-DoC01”
      wsag:ServiceName=”ComputeJob1”>
      <job:Compliance>US-EU</job:Compliance>
    </wsag:ServiceDescriptionTerm>
...
    <wsag:ServiceDescriptionTerm
      wsag:Name= ”numberofPasswordDigitalRequired”
      wsag:ServiceName=”ComputeJob1”>
      <numberOfMinDigit>6 </numberOfMinDigit>
    </wsag:ServiceDescriptionTerm>
...
    <wsag:GuaranteeTerm wsag:Name=”ConfigurationPreference” wsag:Obligated=ServiceProvider>
      <wsag:ServiceScope>
        <wsag:ServiceName>ComputeJob1< /wsag:ServiceName>
      </wsag:ServiceScope>
      <wsag:ServiceLevelObjective xsi:type=”sdtc:OpType”>
        <SDT>CompliantStandardUS-DoC01</SDT>
...
        <SDT>numberofPasswordDigitalRequired </SDT>
...
      </wsag:ServiceLevelObjective>
    </wsag:GuaranteeTerm>
  </wsag:All>
</wsag:Terms>
</wsag:AgreementOffer>
```

guarantees required by HIPAA.

6.2 Data breach

Data breaches are the most frequently occurring security incidents and can lead to lawsuits in some particular application areas such as those covered by the HITECH Act [HITECH 2013] and ENISA [ENISA 2012].

When a data breach occurs, the security group discovers the attack path from the logs in the cloud instances and the logs from the hosting providers according to the SSLA framework. Based on clues that may be found in the logs, the group queries the attacks from OKB to determine the possible vulnerabilities and the corresponding defense acts. Following the suggestions provided by the OKB, the system performs the necessary patches and upgrades any necessary components to eliminate the vulnerabilities and disrupt latent attack paths.

Table I: Transformer's Configuration

#	IT Product	Vulnerabilities	Exploits	Defenses
1	Ubuntu 12.04	97	1	0
2	Apache 2.2.16	21	11	1
3	MySql 5.0.51a	25	3	4
4	PHP 5.4.0	31	2	2
5	Elgg 1.7.10	4	1	0

6.3 Social Network Application built on Elgg

To validate our proposed Optimus Framework, we illustrate one of the use cases for a sample social network service called Transformer. Our service is built on Elgg[10], which is a powerful open source social networking engine that is offering the core components that any developer needs to build out socially aware services. Transformer is deployed in a public cloud. To use the Elgg engine, we need to ensure that we meet its basic web server configurations such as Apache web server, MySQL database system, and the PHP interpreted scripting language. We realize this by renting an Ubuntu server cloud instance with enough resources to support an XAMPP[11] server configuration required for the Elgg engine and our service customization.

Once our application is ready to go live, we want to assess its security status and ensure that all necessary privacy features are in place to satisfy user requirements. Using the Optimus framework, we can use a threat centered security approach to assess our application by evaluating its configurations. Then for each IT Product name and version, using Optimus's Vulnerabilities, Attacks, and Defenses OKBs, we iteratively search for all known vulnerabilities and if found, classify them into relevant threat types. This in turn will allow us to estimate an aggregated risk for our application and get recommended mitigation plans. As the Transformer's configuration shown in Table I, our framework shows that Ubuntu 12.04 has known 97 known vulnerabilities and one exploit in public, but there is no defense released. For MySql 5.0.51a, 25 vulnerabilities and three exploits are found, and there exist four defenses to mitigate their risks. Using these security evaluation details from the OKBs, we check with our Optimus SLA ontology for additional security information that may help us choose the best hosting provider, i.e., the one that offers the best guaranteed service agreement parameters for security of the Elgg service we want to build.

7. CONCLUSION

In this paper, we describe our Optimus framework that contains two main parts: the vulnerability knowledge base of attack and defense and the SSLA ontologies for cloud computing. Our framework can be used to evaluate risks of the known vulnerabilities and attacks of the services and the system configurations, and further to apply appropriate and available defenses to them. Our SSLA ontologies can be used to understand the security agreements of a provider, to negotiate desired security levels, and to audit the compliance of a provider with respect to federal regulations.

In the future, we envision developing some extended applications based on our framework such as a security assurance model for cloud providers. In the model, a cloud provider may offer the cloud computing services with different levels of secure configurations that guarantee security under a restricted environment operating over a limited period. The model is needed for most cloud consumers lacking security knowledge to defend against attacks from any vulnerability that needs to be patched and managed.

[10] Elgg Foundation project: http://elgg.org/

[11] XAMPP: https://www.apachefriends.org/index.html

REFERENCES

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., and Xu, M. 2011. Web services agreement specification (ws-agreement). Tech. rep., Open Grid Forum (OGF). Nov.

CSA. 2013. Cloud controls matrix version 3.0. Tech. rep., Cloud Security Alliance.

CSCC. 2012. Practical guide to cloud service level agreements. Tech. rep., Cloud Standards Customer Council.

ENISA 2012. Procure secure: A guide to monitoring of security service levels in cloud contracts. Tech. rep., European Union Agency for Network and Information Security.

Fenz, S. 2010. Ontology-based generation of it-security metrics. In *Proceedings of the 2010 ACM Symposium on Applied Computing*. 1833–1839.

Hale, M. and Gamble, R. 2013. Building a compliance vocabulary to embed security controls in cloud slas. In *Proceedings of IEEE 9th World Congress on Services*. 118–125.

Henning, R. R. 1999. Security service level agreements: quantifiable security for the enterprise? In *Proceedings of the 1999 Workshop on New Security Paradigms*. Ontario, Canada, 54–60.

HIPAA. 2013. Hipaa administrative simplification. Tech. rep., U.S. Department of Health and Human Services Office for Civil Rights. Mar.

HITECH. 2013. Health information technology for economic and clinical health (hitech) act. Tech. rep., U.S. Department of Health and Human Services Office for Civil Rights. Oct.

Kamongi, P., Kotikela, S., Kavi, K., Gomathisankaran, M., and Singhal, A. 2013. Vulcan: Vulnerability assessment framework for cloud computing. In *Proceedings of IEEE 7th International Conference on Software Security and Reliability*. 218–226.

Keller, A. and Ludwig, H. 2003. The wsla framework: Specifying and monitoring service level agreements forweb services. *Journal of Network and Systems Management 11,* 1 (Mar.), 57–81.

Lee, C.-Y., Kavi, K. M., and Gomathisankaran, M. 2014. Ontology-based privacy setting transfer scheme on social networking systems. In *Proceedings of the 2014 International Conference on Security and Management*. 506–515.

McGlothlin, J. P., Khan, L., and Thuraisingham, B. 2011. Rdfkb: a semantic web knowledge base. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. Vol. 22. 2830–2831.

Modica, G. D., Petralia, G., and Tomarchio, O. 2012. A business ontology to enable semantic matchmaking in open cloud markets. In *Proceedings of the 8th International Conference on Semantics, Knowledge and Grids*. Beijing, China, 96–103.

of Government Commerce, O. 2010. *Introduction to the ITIL Service Lifecycle*, 2nd ed. The Stationery Office, Norwich.

PCIDSS. 2013. Payment card industry data security standard: Requirements and security assessment procedures. Tech. rep., PCI Security Standards Council. Nov.

Rong, C., Nguyen, S. T., and Jaatun, M. G. 2013. Beyond lightning: A survey on security challenges in cloud computing. *Computers and Electrical Engineering 39,* 1, 47–54.

Takahashi, T., Kannisto, J., Harju, J., Heikkinen, S., Silverajan, B., Helenius, M., and Matsuo, S. 2013. Tailored security: Building nonrepudiable security service-level agreements. *IEEE Vehicular Technology Magazine 8,* 3 (Sept.), 54–62.

Tsoumas, B. and Gritzalis, D. 2006. Towards an ontology-based security management. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*. Vol. 1. 985–992.

Wang, A. J. A., Xia, M., Guo, M., Wang, H., and Zhou, L. 2009. Osat tutorial.

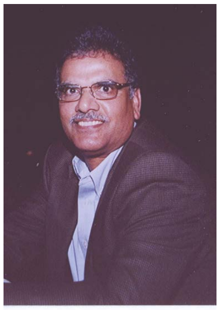
Wang, J. and Guo, M. 2009. Ovm: an ontology for vulnerability management. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*. 34.

Zhou, B. and Pei, J. 2008. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of IEEE 24th International Conference on Data Engineering*. 506–515.

Dr. Chen-Yu Lee received his Ph.D. degrees in Computer Science from the National Chiao Tung University in Taiwan in 2013. He is currently a researcher of Computer Science and Engineering at the University of North Texas. He was an engineer at Industrial Technology Research Institute during 2011-2013. He received the excellent paper awards in ILTC'11 and ECA'11 in 2011. He served on program committees for ICIT2014. His research interests include cryptography, secure protocols, data hiding, privacy, medical information system, and cloud security.

Patrick Kamongi is a PhD candidate in Computer Science and Engineering at the University of North Texas (UNT). He is a member of Computer Systems Research and Trusted Secure Systems Labs at UNT. His main research focus is on vulnerability assessment for cloud computing systems and security ontology engineering. And his other areas of interest are: Cyber Security, Cryptography, Trusted Secure Computing, Information Assurance, and Big Data.

Dr. Krishna Kavi is currently a Professor of Computer Science and Engineering and the Director of the NSF Industry/University Cooperative Research Center for Net-Centric Software and Systems at the University of North Texas. During 2001-2009, he served as the Chair of the department. He also held an Endowed Chair Professorship in Computer Engineering at the University of Alabama in Huntsville, and served on the faculty of the University Texas at Arlington. He was a Scientific Program Manager at US National Science Foundation during 1993-1995. He served on several editorial boards and program committees. His research is primarily on Computer Systems Architecture including multi-threaded and multi-core processors, cache memories and hardware assisted memory managers. He also conducted research in the area of formal methods, parallel processing, and real-time systems. He published more than 150 technical papers in these areas. He received more than US $5 M in research grants. He graduated 14 PhDs and more than 35 MS students. He received his PhD from Southern Methodist University in Dallas Texas and a BS in EE from the Indian Institute of Science in Bangalore, India.



Dr. Mahadevan Gomathisankaran was an Assistant Professor in Computer Science and Engineering at the University of North Texas. He received his Ph.D. degree in Computer Engineering from Iowa State University. He is the recipient of IBM Ph.D. Fellowship award for the academic years 2004 and 2005. Mahadevan is interested in building secure computing systems. Towards that goal he has designed various cryptographic functions that achieve the required security with minimal circuit complexity, proposed new secure processor architecture that root the security in the hardware, and designed a testing framework that can test the security of the systems. He has published more than 20 articles in leading journals and conferences. He is an Associate Editor for The Information Security Journal: A Global Perspective Security. He has served in technical program committees of several international conferences.