How Good is a Security Policy against Real Breaches? A HIPAA Case Study

Özgür Kafalı*, Jasmine Jones†, Megan Petruso‡, Laurie Williams*, and Munindar P. Singh*
*Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206, USA
{rkafali,laurie_williams,singh}@ncsu.edu
†College of Arts and Sciences, Elon University, Elon, NC 27244, USA
jjones92@elon.edu
‡Department of Computer Science, Appalachian State University, Boone, NC 28608, USA
petrusomc@appstate.edu

***Abstract*—Policy design is an important part of software development. As security breaches increase in variety, designing a security policy that addresses all potential breaches becomes a nontrivial task. A complete security policy would specify rules to prevent breaches. Systematically determining which, if any, policy clause has been violated by a reported breach is a means for identifying gaps in a policy. *Our research goal is to help analysts measure the gaps between security policies and reported breaches by developing a systematic process based on semantic reasoning.* We propose SEMAVER, a framework for determining coverage of breaches by policies via comparison of individual policy clauses and breach descriptions. We represent a security policy as a set of norms. Norms (commitments, authorizations, and prohibitions) describe expected behaviors of users, and formalize who is accountable to whom and for what. A breach corresponds to a norm violation. We develop a semantic similarity metric for pairwise comparison between the norm that represents a policy clause and the norm that has been violated by a reported breach. We use the US Health Insurance Portability and Accountability Act (HIPAA) as a case study. Our investigation of a subset of the breaches reported by the US Department of Health and Human Services (HHS) reveals the gaps between HIPAA and reported breaches, leading to a coverage of 65%. Additionally, our classification of the 1,577 HHS breaches shows that 44% of the breaches are accidental misuses and 56% are malicious misuses. We find that HIPAA's gaps regarding accidental misuses are significantly larger than its gaps regarding malicious misuses.**

***Index Terms*—Security and privacy breaches, social norms, breach ontology, semantic similarity**

I. INTRODUCTION

A *security policy* describes the requirements, regulations, and standards that an organization should meet to protect its assets, and enables technical and social protocols to be implemented accordingly. Designing a comprehensive policy is the first step for implementing security controls, though not a trivial task, especially for modern information systems where users play an important role. As a result, policies are often stated in an ambiguous manner [16], [27], and fail to address specific breaches that happen in real life. A *security breach* may be an *accidental misuse* or a *malicious misuse*. Malicious misuses correspond to outsider attacks, whereas accidental misuses correspond to insider attacks or human errors, some of which are unavoidable given the needed functionality.

Gaps between (design time) security policies and (run time) breaches are common in healthcare [20], [25]. Consider the following breach and the corresponding US Health Insurance Portability and Accountability Act (HIPAA) [8] clause:

Example 1. In 2010, a failure to erase data contained on disposed photocopiers' hard drives led to the disclosure of patient records [9]. HIPAA clause *45 CFR 164.310–(d)(2)(i)* describes disposal of electronic records as follows: "Implement policies and procedures to address the final disposition of electronic protected health information, and/or the hardware or electronic media on which it is stored."

Identifying the commonalities and differences between policy clauses and breach descriptions is important for determining which, if any, policy clause has been violated by a reported breach and identifying the gaps in between. In Example 1, HIPAA states that *electronic media* on which patient records are stored must be properly disposed of. According to the breach, a specific incident occurred regarding *photocopiers' hard drives*. A domain ontology captures relationships between such concepts, e.g., hard drives are electronic media.

Our research goal is to help analysts measure the gaps between security policies and reported breaches by developing a systematic process based on semantic reasoning.

Accordingly, we propose SEMAVER, a semantic reasoning framework for measuring the gaps between a security policy and reported breaches. We represent a security policy as a set of norms. Norms (commitments, authorizations, and prohibitions) describe the expectations of users from each other regarding their social interactions, and formalize who is accountable to whom and for what [13], [34]. For example, healthcare employees are prohibited by their hospital from disclosing a patient's medical condition. Norms may be violated since the actions of users are unpredictable, e.g., an employee shares a patient's condition with a friend. A security breach corresponds to a norm violation [15]. Therefore, we represent a breach via a norm that has been violated in the breach.

We seek to address the following research questions:

RQ_1: How can we formalize security policies and breaches to bring out their mutual correspondence?

RQ_2: What are the commonalities and differences between

concepts in security policies and breach descriptions?

RQ$_3$: How do commonalities and differences between individual concepts correspond to gaps between security policies and breaches?

RQ$_4$: How prevalent are accidental misuses among reported breaches, and do security policies account for them?

Current efforts on identifying potential breaches and eliciting security requirements propose visual representations to assist in policy design [12], [17], [28], [36]. However, these representations are either informal or not rich enough to perform semantic reasoning. A normative representation enables us to build a correspondence between policies and breaches (RQ$_1$). We develop a healthcare breach ontology using breach descriptions reported by the US Department of Health and Human Services (HHS) [9]. We propose a semantic similarity metric to understand how various breach concepts relate to each other (RQ$_2$). We extend the similarity metric for pairwise comparison between a norm that represents a policy clause and a norm that represents a breach. We further extend norm similarity as a general metric (policy coverage) to measure the gaps between a security policy and breaches (RQ$_3$). We provide a classification of breaches to differentiate between accidental and malicious misuses (RQ$_4$).

Our contributions include (i) a formal representation of security policies and breaches via norms; (ii) a semantic similarity metric for the pairwise comparison of norms; (iii) a policy coverage metric to measure the gaps between a policy and breaches; and (iv) a classification of breach types.

The rest of the paper is structured as follows. Section II reviews the technical background for our framework. Section III describes the elements of the SEMAVER framework. Section IV presents a HIPAA case study. Section V describes the limitations of our framework. Section VI reviews the relevant literature. Section VII presents future directions.

II. TECHNICAL BACKGROUND

We now review the necessary background for SEMAVER.

A. Norms

Definition 1 describes a norm as a directed relation between two parties [34]. We consider three types of norms: commitments, authorizations, and prohibitions.

Definition 1. *A norm is a tuple* $\langle n,$ SBJ, OBJ, *ant, con*$\rangle$*, where* n*, its type, is one of* $\{c, a, p\}$*;* SBJ *is its subject;* OBJ *is its object;* ant *is its antecedent; and* con *is its consequent. Here,* SBJ *and* OBJ *are roles adopted by people or organizations;* ant *and* con *are propositional conditions. We write a norm as* $n($SBJ, OBJ, *ant, con*$)$*.*

A norm is detached when its antecedent holds, meaning that the norm is active. Violation conditions differ according to the type of norm, as we describe next.

A *commitment* means that its subject is committed to its object to bringing about the consequent if the antecedent holds. For example, healthcare workers are committed to their hospital to properly disposing of patients' electronic health records (EHRs) from any media that is obsolete. The healthcare worker is accountable for improper disposal of EHRs. If a patient's protected health information (PHI) is improperly disposed of, the commitment is violated.

An *authorization* means that its subject is authorized by its object to bring about the consequent if the antecedent holds. For example, physicians are authorized by their hospital to access all patients' EHRs when there is an emergency. The hospital is accountable for not allowing physicians to do so. If a physician tries to access a patient's EHR in an emergency, but is denied access, the authorization is violated.

A *prohibition* means that its subject is prohibited by its object from bringing about the consequent if the antecedent holds. For example, physicians are prohibited by their hospital from sharing patients' PHI with outsiders. A physician is accountable for disclosure of a patient's PHI. If a patient's PHI is disclosed to outsiders, the prohibition is violated.

B. Ontologies

An *ontology* is a conceptualization of a particular domain [6]. By having a formal ontology, one can specify domain concepts and relations among them to perform semantic reasoning. Ontology concepts are tied together via relations. Two domain-independent relations are important. The *is-a* relation denotes that a concept is a type of another concept, e.g., an *Accidental Misuse* is a *Breach*. Certain properties of a concept can be described via the *has-a* relation, e.g., an *Accidental Misuse* has an *Actor*. Here, we can specify the property *hasActor* for the concept *Accidental Misuse*. The range of the property is another concept *User*, and its arity is $1..N$ as an accidental misuse may involve one or more actors. Additional relations can be added to the ontology to make a domain representation richer. Once a domain is represented as an ontology, one can perform semantic reasoning on it using inference rules. A sample rule is that a physician needs a valid license to operate upon patients. Now, the type of license (which may also be given as a taxonomy in the ontology) may depend upon the type of operation.

From a security and privacy perspective, the nature of information content can be represented in an ontology. For example, if a hospital prohibits its employees from disclosing patients' PHI, then given an ontology for healthcare information, semantic reasoning can discover that a patient's laboratory results are part of the patient's PHI and should not be disclosed, whereas the state the patient resides in is not part of the patient's PHI and disclosing it would not violate the prohibition.

Ontologies for security and privacy are emerging. Souag et al. [36] propose a security ontology to anticipate cybersecurity attacks during early requirements elicitation stage. They gather related knowledge from security standards and analyze other (incomplete) security ontologies to develop their ontology. We share the motivation that a formal ontology would help security analysts design security policies with wider coverage of concepts. We adopt some of Souag et al.'s concepts, including organizations (sociotechnical systems with people

and software) and methods of attack. Moreover, we take a more practical approach and focus on reported security breaches. Our ontology can be used to understand the common properties of breaches as well as to compute how well they are covered by security policies.

Slavin et al. [35] propose a privacy-policy-phrase ontology to detect misalignments between privacy policies and API methods of Android applications. They aim to understand which applications that have access to sensitive information violate the stated privacy policies. Slavin et al.'s approach provides a mapping between the API methods and policy terminology, and detects whether there is no violation, a potential weak violation, or a potential strong violation. Their ontology is limited to a taxonomy. Apart from the fact that their domain and ours are different, our ontology is enriched with properties that enable us to perform semantic reasoning.

C. Semantic Similarity

A formal ontology enables semantic reasoning over concepts and relationships, especially to determine how various breach concepts relate to each other. We extend this reasoning to provide a formal comparison of norms, which is used to measure how much a security policy differs from breaches.

Resnik [29] proposes a measure to compute similarity of concepts given in a taxonomy with is-a relations. A common way to compute similarity is to calculate the distance between concepts, with the shorter path indicating a higher similarity between them. However, links in a taxonomy (is-a relations) do not always represent uniform distances. Resnik proposes an alternative way to compute similarity based on information content. Empirical evaluation shows that Resnik's measure is the closest to human judgment among other taxonomy based distance metrics such as node distance and edge counting [32]. Lin [21] builds upon Resnik's information theoretic metric to compare concepts in a taxonomy by identifying the commonalities as well as the differences between the concepts. We go beyond similarity in a taxonomy, and use properties of concepts to provide a richer similarity metric.

Rodríguez et al. [31] propose a method to compute similarity of concepts contained in different ontologies. Their method systematically detects similar concepts across ontologies via a matching process based on specifications such as synonym sets and semantic neighborhoods. We currently have a single breach ontology developed based on the knowledge gathered from HHS incidents. Finding similarity between random words such as *triangle* and *breach* [21], [29], or identifying synonyms [18], [24] are out of our scope. However, it would be interesting to extend our ontology with knowledge from other domains, and compare various breach concepts using the proposed methods.

III. SEMAVER FRAMEWORK

This section describes our development of SEMAVER: (i) a breach ontology based on reported breach incidents, (ii) a formal representation of policies and breaches via norms, (iii) semantic similarity relations for ontology concepts as well as norms, and (iv) a methodology for measuring how much a security policy differs from associated breaches (coverage). Throughout this section, we use examples from the healthcare domain to explain elements of SEMAVER.

A. Ontology Development

Figure 1 shows how a breach ontology can be constructed using both generic and domain-dependent concepts. Under the general concept *Breach*, we have its subclasses described via *is-a* relations, e.g., *Unintentional disclosure* is a *Breach*. The action *Share PHI with family*, while performed intentionally by the physician, does not involve any malicious intent of disclosing patient's data. Note that some concepts (e.g., *Accidental Misuse*) are omitted for brevity. A detailed description of our healthcare breach ontology is presented in Section IV-B (see also Figure 4). Having such a hierarchy of breach concepts in an ontology is helpful for developing semantic similarity based on the distance between concepts. However, taxonomic distance itself is not a sufficient measure to determine similarity: pairs of concepts that have the same taxonomic distance from each other may have different conceptual similarity. Therefore, we describe properties of concepts with *has-a* relations, e.g., *Malware* has actor *Adversary*. Note that an ontology concept can have multiple properties: the figure shows only one property for brevity. Moreover, properties can be related to each other via is-a relations (see Figure 2 for an ontology of healthcare users regarding the *hasActor* property).

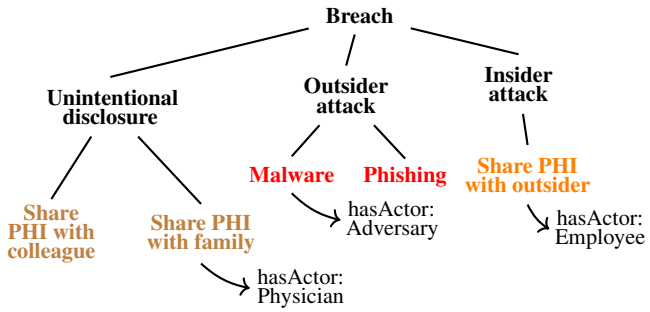

Fig. 1. Breach concepts. Lines represent subclass (is-a) relations among concepts. Arrows represent properties of concepts (has-a relations).

B. Representing Policies and Breaches

A norm supports accountability between its subject and object, thus provide a natural means of formalizing policy clauses that capture security and privacy requirements. Examples 2 and 3 describe two clauses of HIPAA together with the norms that formalize them.

Example 2. HIPAA clause *45 CFR 164.510* describes disclosure conditions of patients' PHI as follows:

> *"The covered entity may orally inform the individual of and obtain the individual's oral agreement or objection to a use or disclosure permitted by this section."*

This clause is represented with the following prohibition: p(PHYSICIAN, COVERED_ENTITY, ¬consent,

share_PHI_family). That is, a physician working in the covered entity (e.g., a healthcare provider organization) is prohibited from sharing a patient's PHI with the patient's family unless a consent is obtained from the patient. Note that the ¬ symbol denotes negation.

Moreover, subclause *45 CFR 164.510–(b) Permitted uses and disclosures* of the above clause states the following:

> *"The requirements to obtain a patient's agreement to speak with family members or friends involved in the patients care can be waived during national disasters."*

This subclause can be represented with the following authorization: a(PHYSICIAN, COVERED_ENTITY, national_disaster, share_PHI_family). That is, a physician is authorized to share a patient's PHI with the patient's family in a national disaster.

Example 3. Consider HIPAA clause *45 CFR 164.310–(d)(2)(i)* from Example 1, which we represent as the following commitment: c(HEALTHCARE_WORKER, COVERED_ENTITY, media_disposal, erase_media). That is, healthcare workers are committed to a covered entity for proper destruction of patients' EHRs that are stored on electronic media.

The above examples demonstrate how we can specify norms to formalize individual clauses of a security policy. Definition 2 describes a security policy as a set of norms, where each norm corresponds to a clause of the policy.

Definition 2. *A security policy $\mathcal{S}$ is a set of norms, $\mathcal{S} = \{n_1, \ldots, n_k\}$.*

Next, we formalize breaches. Definition 3 describes a breach as a norm violation.

Definition 3. *A breach b_i is a violation of a norm n_i, i.e., b_i = violated(n_i).*

Let us see example breaches summarized from the HHS descriptions [9], and corresponding norm violations. Example 4 revisits Example 1 regarding improper disposal of EHRs.

Example 4. Consider the breach in Example 1, where a healthcare worker did not erase the photocopiers' hard drives. That breach is a violation of the commitment c(HEALTHCARE_WORKER, COVERED_ENTITY, media_disposal, erase_media), because the commitment is detached when the photocopier (a certain type of media) is disposed of, but the subject of the commitment failed to bring about the consequent.

Example 5 describes an incident regarding the use of personal devices for work purposes.

Example 5. Consider the following breach reported by HHS regarding Iowa Department of Human Services:

> *"Employees of the covered entity used personal email accounts, personal online storage accounts and personal electronic devices for work purposes. From February 5, 2010 to January 17, 2014, the protected health information (PHI) of 2,042 individuals was transferred outside of the covered entity's secure network in this manner."*

This breach is a violation of the prohibition p(HEALTHCARE_WORKER, COVERED_ENTITY, true, use_personal_device), because healthcare workers are prohibited from using personal devices for work at all times (the prohibition is detached since the antecedent is true).

C. Similarity Metric

We develop a similarity metric to compare ontology concepts and norms. Our similarity metric adopts ideas from the literature on ontologies [29], [32], [33], [39], and extends it for pairwise comparison of norms. Moreover, our similarity metric uses the specified properties of concepts as well as the relations among properties to enable a deeper understanding of similarity.

Equation 1 describes the distance between two ontology concepts c_1 and c_2 via edge_count(c_1, c_2), which is the number of edges connecting concepts c_1 and c_2.

$$\Delta_{c_1,c_2} = \text{edge\_count}(c_1, c_2) \tag{1}$$

Assumption 1. *There are no multiple inheritance relations in the healthcare breach ontology.*

Assumption 2. *Subclass (is-a) relationships in the healthcare breach ontology are of equal importance.*

Equation 1 computes distance for tree-like taxonomies, and does not work when multiple inheritance is allowed (Assumption 1). Edges have uniform weights (Assumption 2). Therefore, we count each edge as one.

Equation 2 describes our similarity metric between two ontology concepts c_1 and c_2 (sim_{c_1,c_2}). We denote taxonomy-based distance between concepts c_1 and c_2 via Δ_{c_1,c_2}, and property similarity between concepts c_1 and c_2 via $sim^{prop}_{c_1,c_2}$.

$$sim_{c_1,c_2} = \frac{1}{1 + \Delta_{c_1,c_2}} \times sim^{prop}_{c_1,c_2} \tag{2}$$

The maximum possible similarity value, of one, is only achieved when two concepts are identical. Let us revisit Example 1. HIPAA states that electronic media on which patient records are stored must be properly disposed of. According to the breach, a specific incident occurred regarding photocopiers' hard drives. HIPAA states the parent concept (electronic media), which provides maximum coverage for photocopiers.

Equation 3 describes property similarity between concepts c_1 and c_2 based on the set of common properties (P) of c_1 and c_2, and distances between the values of those properties.

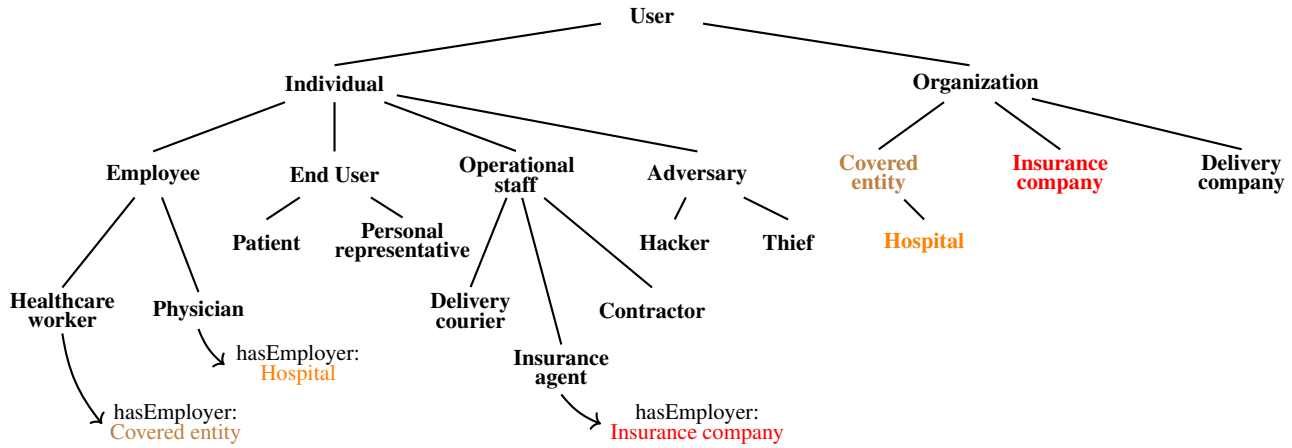

Fig. 2. Ontology of healthcare users. Lines represent subclass (is-a) relations among concepts. Arrows represent properties of concepts (has-a relations).

$$sim^{prop}_{c_1,c_2} = \begin{cases} \text{min\_sim} & \text{if P} = \emptyset \\ \prod_{p_i \in \text{P}} \frac{1}{1+\Delta_{p_i}} & \text{otherwise} \end{cases} \quad (3)$$

Equation 3 resembles Lin's [21] commonality measure in the sense that we explore the common properties of concepts to determine their similarity. Lin's commonality and differences measures rely on information content derived from instances of concepts (e.g., a probability distribution). We look at the similarity between generic concepts such as *Breach* and *Adversary*. Deriving such a probability distribution from breach descriptions is left for future work.

Assumption 3. *min_sim* = 0.001.

When two concepts have no common properties, we assign a minimum value (min_sim) to the similarity between them according to Assumption 3. Note that min_sim can be defined based on the size of the ontology. For example, for an ontology containing 100 to 1,000 concepts, min_sim = 10^{-3} = 0.001.

Let us revisit the ontology of Figure 1 and see examples of how taxonomy-based distance and property similarity are calculated.

Example 6. Consider concepts *Share PHI with family* and *Share PHI with colleague*. These concepts have the same parent. Thus, $sim_{\text{Share PHI with family,Share PHI with colleague}} = 0.33 \times sim^{prop}_{\text{Share PHI with family,Share PHI with colleague}}$.

Example 7. Consider concepts *Share PHI with family* and *Share PHI with outsider*. These concepts are not as similar as the concepts in Example 6, because there are three concepts in between the two concepts. Thus, $sim_{\text{Share PHI with family,Share PHI with outsider}} = 0.2 \times sim^{prop}_{\text{Share PHI with family,Share PHI with outsider}}$.

Taxonomy distance gives a good estimate of how similar the concepts are (Examples 6 and 7). However, taxonomy distance alone is not always adequate for determining similarity when several concepts have the same distance from each other. Example 8 calculates property similarity using the ontology of healthcare users shown in Figure 2.

Example 8. The distance between concepts *Share PHI with family* and *Share PHI with outsider* and the distance between concepts *Share PHI with family* and *Malware* are the same. However, *Share PHI with outsider* and *Malware* are not necessarily similar concepts. Therefore, we need to take into account the properties of those concepts to determine similarity. Let us assume that the only common property of the concepts *Share PHI with family*, *Share PHI with outsider*, and *Malware* is *hasActor*. According to Equation 3, similarity between the actors *Physician* (for *Share PHI with family*) and *Employee* (for *Share PHI with outsider*) is 0.5. Thus, $sim_{\text{Share PHI with family,Share PHI with outsider}} = 0.2 \times 0.5 = 0.1$. Likewise, similarity between the actors *Physician* (for *Share PHI with family*) and *Adversary* (for *Malware*) is 0.25. Thus, $sim_{\text{Share PHI with family,Malware}} = 0.1 \times 0.25 = 0.025$.

We build upon Equation 2 to compare norms based on their individual elements. Note that the subject and object of a norm correspond to concepts given in Figure 2, and the antecedent and consequent correspond to concepts given in Figure 1. Definition 4 describes norm similarity.

Definition 4. *The similarity between norms* n_1(SBJ$_1$, OBJ$_1$, ant_1, con_1) *and* n_2(SBJ$_2$, OBJ$_2$, ant_2, con_2) *is the average similarity of its elements:* $sim_{n_1,n_2} = (sim_{\text{SBJ}_1,\text{SBJ}_2} + sim_{\text{OBJ}_1,\text{OBJ}_2} + sim_{ant_1,ant_2} + sim_{con_1,con_2})$ / 4.

Assumption 4. $sim_{\phi,\text{true}}$ = *min_sim*.

Note that the antecedent of a norm can be true. According to Assumption 4, we assign the minimum similarity value to any predicate that is compared with true. Example 9 shows an example of norm similarity.

Example 9. Consider norms p_1(PHYSICIAN, HOSPITAL, ¬emergency, share_PHI_family) and p_2(HEALTHCARE_WORKER, COVERED_ENTITY, true, share_PHI_outsider). Similarity between subjects is $sim_{\text{PHYSICIAN,HEALTHCARE\_WORKER}} = 0.33 \times 0.5 = 0.17$. Similarity

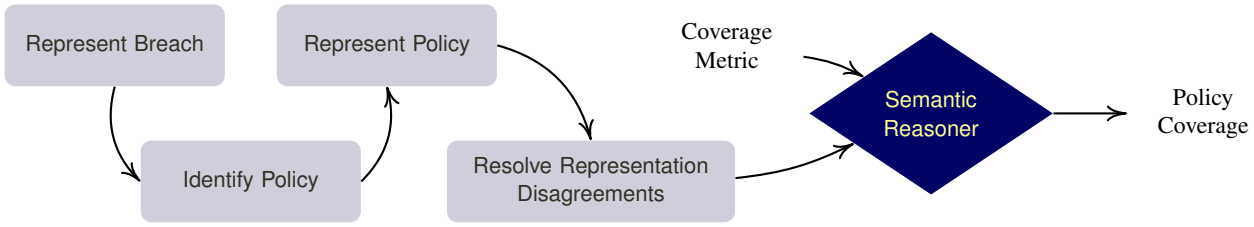

Fig. 3. SEMAVER methodology for determining policy coverage.

between objects is $sim_{\text{HOSPITAL,COVERED\_ENTITY}} = 0.5$. Similarity between antecedents is $sim_{\neg\text{emergency,true}} = 0.001$. Similarity between consequents is $sim_{\text{share\_PHI\_family,share\_PHI\_outsider}} = 0.2 \times 0.5 = 0.1$. Thus, $sim_{p_1,p_2} = 0.19$.

Norms of different types are not similar. According to Assumption 5, we assign the minimum similarity value to comparisons between different norm types.

Assumption 5. $sim_{c,a} = sim_{c,p} = sim_{a,p} = min\_sim$.

Now, we are ready to describe our policy coverage metric based on norm similarity. First, we present a list of reasoning postulates that transform norms with conjunctive and disjunctive propositions to norms with only atomic propositions.

Postulate 1. *A norm whose antecedent is a disjunction of two propositions can be represented via two norms whose antecedents are the individual propositions:* n(SBJ, OBJ, $ant_1 \vee ant_2$, con) *if and only if* n(SBJ, OBJ, ant_1, con) *and* n(SBJ, OBJ, ant_2, con).

Postulate 2. *A commitment whose consequent is a conjunction of two propositions can be represented via two commitments whose consequents are are the individual propositions:* c(SBJ, OBJ, ant, $con_1 \wedge con_2$) *if and only if* c(SBJ, OBJ, ant, con_1) *and* c(SBJ, OBJ, ant, con_2).

Postulate 3. *An authorization whose consequent is a disjunction of two propositions can be represented via two authorizations whose consequents are the individual propositions:* a(SBJ, OBJ, ant, $con_1 \vee con_2$) *if and only if* a(SBJ, OBJ, ant, con_1) *and* a(SBJ, OBJ, ant, con_2).

Postulate 4. *A prohibition whose consequent is a disjunction of two propositions can be represented via two prohibitions whose consequents are the individual propositions:* p(SBJ, OBJ, ant, $con_1 \vee con_2$) *if and only if* p(SBJ, OBJ, ant, con_1) *and* p(SBJ, OBJ, ant, con_2).

Definition 5 describes how norm n_1 *covers* norm n_2 so that n_2 can be replaced by n_1. Simply put, n_1 covers n_2 if n_1 is satisfied whenever n_2 is satisfied [13]. Note that the $\vdash$ symbol represents logical consequence.

Definition 5. *Commitment* c(SBJ, OBJ, ant_1, con_1) *covers commitment* c(SBJ, OBJ, ant_2, con_2) *if and only if* $ant_2 \vdash ant_1$ *and* $con_1 \vdash con_2$. *Authorization* a(SBJ, OBJ, ant_1, con_1) *covers authorization* a(SBJ, OBJ, ant_2, con_2) *if and only if* $ant_1 \vdash ant_2$ *and* $con_1 \vdash con_2$. *Prohibition* p(SBJ, OBJ, ant_1, con_1) *covers prohibition* p(SBJ, OBJ, ant_2, con_2) *if and only if* $ant_2 \vdash ant_1$ *and* $con_2 \vdash con_1$.

Note that the subjects and objects must be identical for a norm to cover another norm. However, we can use the covers relation in combination with our similarity metric for the subjects and objects to determine to what extent a norm covers another norm. That is, if norm n_1 covers n_2, then the similarity between their antecedents and consequents is one, which constitutes half of norm similarity according to Definition 4. The remaining half is determined by the similarity between their subjects and objects. Equation 4 describes how similarity is determined when n_1 covers n_2.

$$sim^{covers}_{n_1,n_2} = 0.5 + (sim_{\text{SBJ}_1,\text{SBJ}_2} + sim_{\text{OBJ}_1,\text{OBJ}_2})/2 \quad (4)$$

Equation 5 builds upon the above development to describe policy coverage that is an average of the similarity values between the norms representing each breach and the corresponding policy clause. B represents the set of all breaches, and |B| represents the cardinality of B.

$$coverage = \frac{\sum_{b_i \in \text{B}} \begin{cases} sim^{covers}_{n_{\text{policy}},n_{b_i}} & \text{if } n_{\text{policy}} \text{ covers } n_{b_i} \\ sim_{n_{\text{policy}},n_{b_i}} & \text{otherwise} \end{cases}}{|\text{B}|} \quad (5)$$

D. Methodology

Figure 3 summarizes our methodology for measuring how well a security policy covers reported breaches. Steps 1 and 3 presuppose familiarity with conceptual modeling based on norms. For Steps 1 to 3, we can employ multiple modelers who work independently. Step 4 is needed only when two or more modelers are employed. Let us review each step:

1) *Represent breach:* Specify the relevant norms for each breach.
2) *Identify policy clause:* For each breach in Step 1, identify a clause of the policy that is the most relevant to the scenario described in the breach. Note that there might be cases where the incident describes a general privacy breach, and it might not be clear which HIPAA policy clause is violated.
3) *Represent policy clause:* Specify the norms for the identified policy clauses in Step 2. Note that one would normally decide whether an incident is a breach by first representing the policy [15]. Here, a breach is an established fact: thus, our reasoning begins by investigating the breaches. Working backwards from breaches saves

significant effort by investigating only relevant policy clauses.

4) *Resolve disagreements:* Discuss and resolve any disagreements for Steps 1–3. For any unresolved disagreement, employ a more experienced modeler to specify the norms after reviewing the disagreement.
5) *Calculate policy coverage:* Feed the norms that represent breaches and relevant policy clauses together with the coverage metric (Equation 5) into the semantic reasoner. The output (policy coverage) shows how much the policy differs from reported breaches.

IV. Case Study: HHS Security and Privacy Breaches

We now evaluate the SEMAVER framework on the HIPAA policy using the HHS breach reports.[1] We are mainly interested in HIPAA security and privacy clauses that are relevant to the breaches reported by HHS.

A. HHS Breach Report

We investigate 1,577 security and privacy incidents reported in an HHS breach report [9]. HHS provides a classification of the breaches contained in the dataset as shown in Table I. Out of the 1,577 breaches, 219 incidents are unclassified. Of these, 40 incidents are marked as other, 5 incidents are marked as unknown, and 174 incidents are unmarked.

TABLE I
HHS BREACH CATEGORIES.

Category	Count	Description
Hacking	191	Incidents where an adversary exploits a software vulnerability to access patients' EHRs
Theft	642	Incidents where an employee (insider) accesses patients' EHRs and discloses their PHI to outsiders
Loss	129	Incidents where electronic media that contain patients' PHI are lost from the possession of an employee
Unauthorized access/disclosure	338	Incidents where a patients' EHR is disclosed due to unauthorized access
Improper disposal	58	Incidents where a healthcare worker fails to properly dispose of patients' EHRs, leaving their PHI exposed
Unclassified	219	Not classified by HHS

B. Healthcare Breach Ontology

Figure 4 shows a screenshot of our breach ontology from the Protégé ontology development tool. In addition to the concepts described in Figures 1 and 2, the complete ontology contains a total of 104 concepts as well as some instances of concepts gathered from the HHS incidents. For example, *Iowa Department of Human Services* is an instance of the ontology concept *Covered entity* (see Example 5).

[1]Case study materials can be found in https://research.csc.ncsu.edu/mas/code/security/semaver/.

The complete ontology describes additional concepts and properties such as the scale of an organization and type of attack. Correlation among these concepts would help us understand whether attackers are targeting large organizations where the potential damage (e.g., the number of affected people) is large, but the chances of a successful attack may be low. We can also calculate the frequency of such attacks. Further investigation in this area is left for future work.

Our breach ontology contains an ontology of norms, which is connected to the rest of the ontology through the elements of a norm. For example, the subject and object of a norm are users from Figure 2, and the antecedent and consequent of a norm include propositions described in Figure 1.

C. Application of SEMAVER

Now, we describe how we apply the SEMAVER methodology (Section III-D) for the HIPAA policy and associated HHS breaches. We randomly selected a subset of the breaches from the HHS dataset which represent unique cases. Note that most of the incidents describe similar breach scenarios and thus correspond to the same HIPAA clauses. Two researchers (undergraduate students) independently specified the norms. In a previous study [13], we have shown that users can specify norms for requirements with minimal training.

Examples 10 and 11 demonstrate sample breaches and the application of the SEMAVER methodology on them.

Example 10. In 2014, a contractor of a covered entity, who is also the husband of an employee of the covered entity, accessed patient records without proper authorization and disclosed their PHI. The associated HIPAA clause *45 CFR 164.308–(b)(2)* states the following regarding business associates: "A business associate may permit a business associate that is a subcontractor to create, receive, maintain, or transmit electronic protected health information on its behalf only if the business associate obtains satisfactory assurances."

The researchers individually specified the following norm for the breach in Example 10: p_1(CONTRACTOR, COVERED_ENTITY, $\neg$consent, access_EHR). The researchers agreed on the following norm regarding the HIPAA clause: p_2(CONTRACTOR, COVERED_ENTITY, $\neg$consent, access_EHR $\vee$ disclose_PHI). According to Definition 5, p_2 covers p_1. Moreover, the subjects and objects of the norms are identical. Thus, $\mathit{sim}_{p_2,p_1} = 1$.

Example 11. In 2015, an employee of a covered entity emailed a questionnaire to patients without using blind carbon copy (bcc) to hide patient names. The most relevant HIPAA clause *45 CFR 164.502–(a)(1)* states the following regarding the disclosure of PHI: "A covered entity is permitted to disclose protected health information to the individual."

The researchers individually specified the following norms for the breach in Example 11: c_1(HEALTHCARE_WORKER, HOSPITAL, email, bcc) and c_2(EMPLOYEE, COVERED_ENTITY, email_patients, bcc). After discussion, c_1 and c_2 are resolved into the

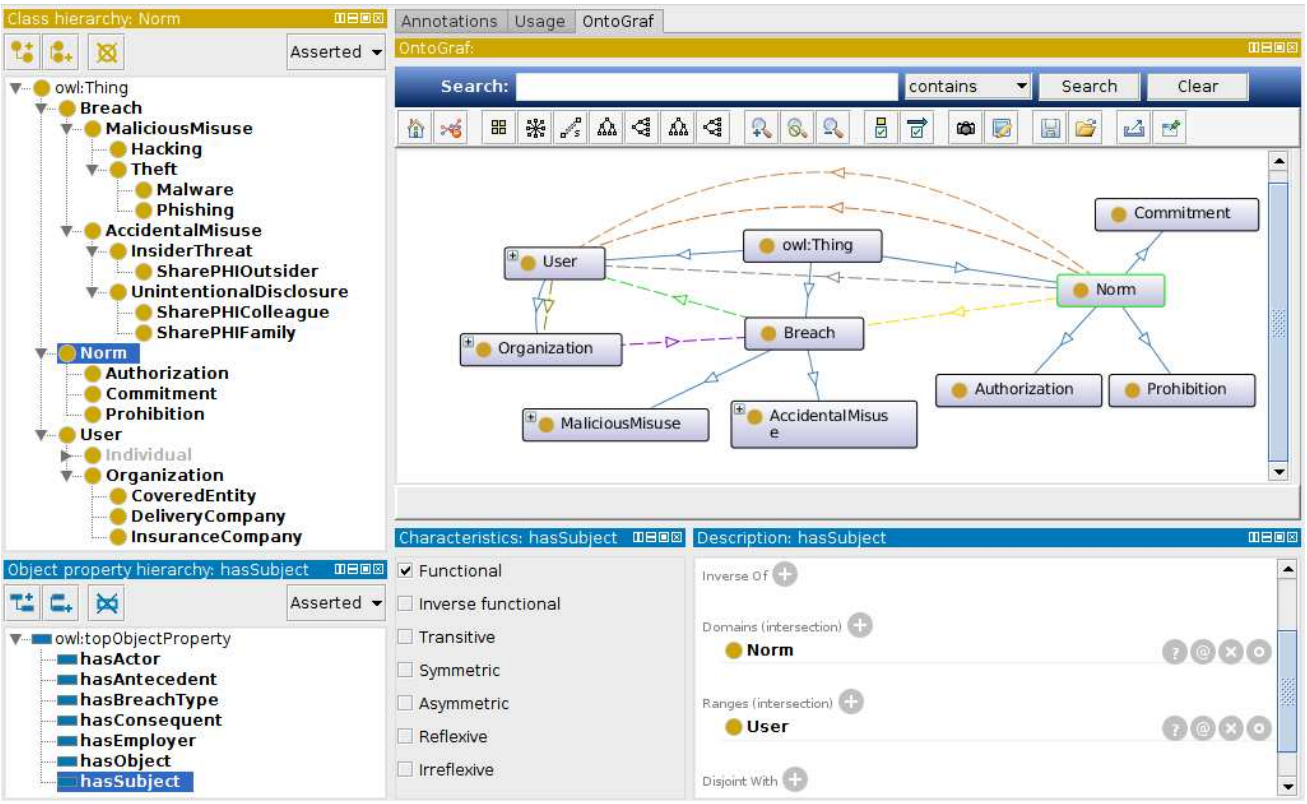

Fig. 4. Healthcare breach ontology in Protégé. The top left pane shows the class hierarchy, and the top right pane depicts the class hierarchy as a diagram. In addition to the class hierarchy, concepts can be connected to each other via properties. The bottom left pane shows object properties, and the bottom right pane shows the domain and range of an object property.

following: c_3(HEALTHCARE_WORKER, COVERED_ENTITY, email_patients, bcc_patients). The researchers agreed on the following norm regarding the HIPAA clause: a_1(HEALTHCARE_WORKER, COVERED_ENTITY, request_individual, disclose_PHI_individual). According to Assumption 5, $sim_{a_1,c_3} = 0.001$.

D. Results

We now present our findings based on the 1,577 breaches reported in the HHS dataset. We randomly selected 40 breaches representative of the number of occurrences in the HHS classification (Table I): seven hacking incidents, 15 theft incidents, five loss incidents, 10 unauthorized access/disclosure incidents, and three improper disposal incidents.

RQ$_1$ and RQ$_2$: We investigate how policy clauses differ from breaches with respect to individual elements of a norm. Figure 5 shows that the similarity between the actors (subject/object of a norm) stated in policies and breaches is higher than the assets (antecedent and consequent of a norm). The similarity of individual norm elements enables us to identify where commonalities and differences reside between policy clauses and breach descriptions. Norm similarity (Definition 4) can be refined according to these findings (e.g., consequent may be given a higher weight since it has the lowest similarity value) to provide a more realistic measure of coverage.

RQ$_3$: Using policy coverage (Equation 5), we measure the gaps between HIPAA and HHS breaches. Based on the 40 incidents, we find that HIPAA has a general coverage of 65%. Moreover, the fact that consequent similarity is the lowest (Figure 5) implies that there are gaps in HIPAA policy for stating what needs to be done or avoided to prevent breaches.

RQ$_4$: We extend the classification provided by HHS for the 1,577 breaches. We differentiate between two types of breaches: accidental misuses (due to interactions of healthcare workers) and malicious misuses (outsider attacks). We confirm via the descriptions of the breaches that the categories *hacking* and *theft* contain malicious misuse incidents, and *loss*, *unauthorized access/disclosure*, and *improper disposal* contain accidental misuse incidents. For the remaining unclassified incidents (*other*, *unknown*, and *unmarked*), we have provided a classification based on the given descriptions. Out of the 40 incidents in the *other* category, one is malicious misuse, 10 are accidental misuses, and 29 have no description. Out of the five incidents in the *unknown* category, two are accidental misuses and three have no description. Out of the 174 *unmarked* incidents, 27 are malicious misuses, 132 are accidental misuses, and 15 have no description. Excluding incidents without descriptions, we obtain a total of 1,530 classified breaches, of which 669 are accidental and 861 are malicious misuses. That is, 44% of the incidents are caused by accidental misuses, which indicates that regulating the social interactions of users is as important as developing a

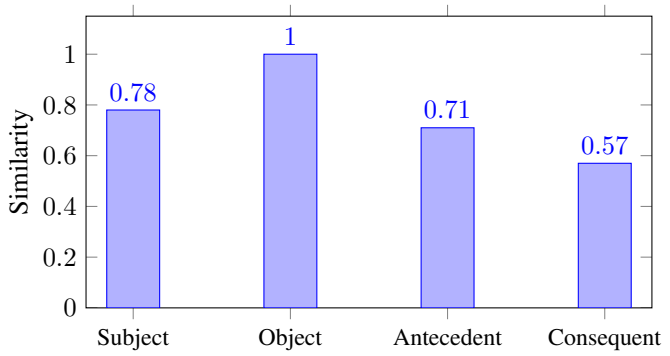

Fig. 5. Similarity for individual norm elements.

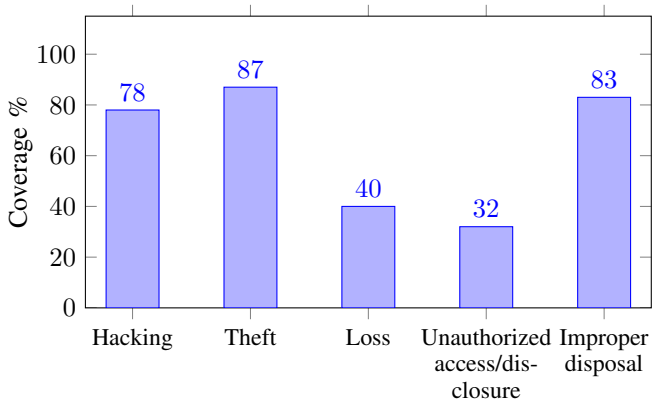

Fig. 6. Policy clause coverage by breach category.

secure software infrastructure to prevent security breaches. Moreover, we compute coverage for each breach category to understand which parts of HIPAA contain more gaps and need revision. Figure 6 shows that HIPAA provides better coverage for malicious misuses than accidental misuses (except for the improper disposal category which constitutes the smallest portion of the breaches). This result reinforces the fact that designing policies to address all potential accidental misuses is a nontrivial task as users become central to information systems. Recent cybersecurity reports [4] and surveys [10] corroborate our findings that healthcare security and privacy policies need continual revision, especially due to emerging threats regarding accidental misuses.

Our coverage metric further breaks down the 65% coverage result, and uncovers practically valuable information: which elements of a policy clause (Figure 5) and which policy categories (Figure 6) need more attention. Although increased coverage does not necessarily imply superior breach prevention policies with increased coverage of breaches is crucial for implementing improved security. Semantic reasoning can help fill the gaps between policies and breaches. For example, if a policy clause states protection of one asset, and a reported breach indicates another asset, it would be reasonable to consider all similar assets in between those two assets.

V. Limitations and Threats to Validity

We identify the following limitations and threats to validity for SEMAVER. First, modeling of security policies and breaches is subjective and inherently error prone. Although we minimize subjectivity and potential errors by independently specifying norms with multiple researchers and resolving disagreements, we cannot completely eliminate subjectivity in specifying the norms.

Second, we cannot assess the completeness of the breaches reported in the HHS dataset. Although we classified all incidents into specific breach categories and identified that some breaches are more common than others and most scenarios overlap with each other, we cannot construct a theoretical proof towards completeness that will guarantee no other breach category will emerge. However, our investigation provides a thorough study of the reported healthcare breaches and proposes a realistic measure of policy coverage.

Third, the coverage of breach concepts by our ontology is based only on the incidents from the healthcare domain. While relying upon HHS breaches is fine for measuring the coverage of the HIPAA policy, not including breaches from other domains poses a threat to the validity of our framework for computing similarity in those domains. We will investigate other domains and extend our ontology to mitigate this threat.

VI. Related Work

Brost and Hoffmann [2] discuss security and privacy related misuses in eHealth systems and their connection with the STRIDE threat model [11]. They consider misuse case diagrams [17] in identifying and mitigating threats, and experiment on a platform that brings together physicians, nurses, and patients to study cases where patient data can be disclosed from smartphones via malicious applications. Matulevičius et al. [23] investigate misuse cases for security modeling using the Information System Risk Management (ISSRM) model. Misuse case diagrams are helpful in visually representing the vulnerabilities of a software system. However, they are often given in general terms, and can cause misinterpretation. Therefore, Matulevičius et al. take into account risk-related concepts, i.e., the likelihood of a threat occurring. Kökciyan and Yolum [19] investigate and categorize privacy breaches in online social networks via a semantic understanding of the events that lead to the breaches. Such breaches mostly correspond to accidental misuses where the users' commitments are violated. We investigate cases where the interactions among healthcare users lead to misuses. In SEMAVER, we represent misuse cases (breaches) as norm violations, which enables us to formally reason about them.

Hao et al. [7] propose a method for designing minimal effective normative models to coordinate agents in an open system. It would be interesting to apply their method to security policy design, and come up with a minimal policy that fills the gaps regarding associated breaches.

Elahi et al. [5] propose an ontology for analyzing attacks and vulnerabilities regarding security requirements. They provide a catalog for vulnerabilities based on knowledge gathered from portals such as the *National Vulnerability Database*

or *Common Weakness Enumeration*. Their ontology includes formal definitions of concepts such as attack, risk, and countermeasure, and is integrated into misuse case diagrams and other conceptual modeling frameworks. In contrast, we develop a richer sociotechnical model of policies and breaches that is centered on norms. Such a formal representation can enable further kinds of reasoning. Moreover, we introduce concepts regarding accidental misuse.

Yskout et al. [41] show that although teams tasked with implementing the security requirements of a banking system prefer security patterns although they do not perform conclusively better than teams with no pattern aid. We posit that patterns can be helpful for security design, provided these patterns are expressed in high-level terms closer to stakeholder requirements than to technical specifications. Our representation centered on norms can potentially provide a basis for the appropriate patterns and tools.

We have proposed design patterns for the revision of (social) system specifications with respect to changing requirements [13]. It would be interesting to drive how the patterns apply from observed breaches placed in a breach ontology. We have also developed a way to compare sociotechnical specifications in terms of their liveness and safety [14]. It would be interesting to identify the underlying requirements that lead users to accidentally or knowingly cause security breaches, as a way of specifying policies that accommodate user needs and reduce the temptation for workarounds.

Alrajeh et al. [1] propose requirements revision in the case of risks that hinder expected behavior of software. They propose a goal-driven risk analysis method via obstacle analysis. Rashid et al. [28] perform multi incident analysis to discover *unknown known* security requirements, which represent emergent requirements that implicitly appear in security incidents (known), but are not familiar to requirements engineers (unknown). Our investigation of HHS breaches reveals important risks in the healthcare domain that need to be addressed, especially regarding interactions among healthcare employees.

Natural language requirements documents are helpful in providing a high level understanding of stakeholder requirements. However, they create ambiguity for requirements engineering [16], where a requirement is interpreted differently by individual stakeholders. Detecting and resolving such ambiguities in earlier stages of software development is crucial as ambiguities may lead to greater problems in later stages. Our work aims at identifying gaps between policies and breaches, which would enable analysts to resolve ambiguities in specific policy clauses. Popescu et al. [27] propose a semiautomated process and a tool, *Dowser*, for identifying ambiguities in natural language software requirements specifications. Dowser relies on a semantic model and a formal grammar, and aims at overcoming the difficulties humans have when identifying ambiguities. Human judgment is still needed to determine whether a produced model represents a *good* set of requirements. Yang et al. [40] propose a machine learning method for detecting whether an ambiguity is *nocuous* (potentially harmful) or not. Riaz et al. [30] propose a framework to infer *implied* security requirements from functional requirements written in natural language via security goal patterns. They evaluate the usability and efficiency of their patterns via a user study on various security scenarios. We do not classify ambiguities in policy clauses based on their harmfulness. However, our approach helps understand which policy clauses need revision with respect to the severity of the associated breaches. Our foundation in norms can help reason about security requirements from a higher-level perspective that is closer to stakeholders' understanding.

VII. Conclusions and Future Work

We proposed SEMAVER, a semantic reasoning framework for identifying gaps between security policies and breaches. We investigated breaches reported by HHS, and found that accidental misuses are almost as prevalent as malicious misuses, which indicates that human factors are as important as fixing vulnerabilities for preventing breaches. To our best knowledge, SEMAVER is the first attempt to investigate real breaches for evaluating policies. A natural next step is to perform formal revision of policies based the identified gaps.

We found that HIPAA has better coverage for malicious misuses than accidental misuses. HIPAA is a dominant healthcare standard, and our findings would illustrate similar concerns in other domains. In essence, our coverage metric measures whether the incident described by a breach description has been considered by policy designers. Validation of the coverage metric is left for future work, which would involve additional breach incidents from other domains [22]. In particular, we plan to investigate the Verizon Data Breach Investigations Reports [38], the DataLoss Database [3], and the Principedia privacy incidents database [37].

Most parts of SEMAVER can be automated. First, norm similarity can be used to identify the most relevant policy clause to a given breach, which would reduce the manual effort and prevent potential human errors. Second, natural language processing can be adopted to automatically extract norms from policies and breaches. Third, we can develop heuristic guidelines for the development of the breach ontology, where concepts and properties are automatically extracted from breach descriptions.

Threat models, such as misuse case diagrams or attack/defense (A/D) trees, describe potential ways a software system can be attacked and how those attacks can be mitigated [12], [26]. While threat models identify potential breaches, they do not directly translate to policy designs. It would be an important contribution to investigate how SEMAVER can help bridge this gap, in particular via norm-based patterns [13].

Acknowledgments

This research is supported by the US Department of Defense under the Science of Security Lablet grant. Jasmine Jones and Megan Petruso were supported by the US National Science Foundation under the Science of Software REU grant at NCSU during the summer of 2016. We thank the Realsearch group at NCSU and the anonymous referees for their helpful comments.

REFERENCES

[1] Dalal Alrajeh, Axel van Lamsweerde, Jeff Kramer, Alessandra Russo, and Sebastian Uchitel. Risk-driven revision of requirements models. In *Proceedings of the 38th International Conference on Software Engineering (ICSE)*, pages 855–865. ACM, 2016.

[2] Gerd Stefan Brost and Mario Hoffmann. Identifying security requirements and privacy concerns in digital health applications. In *Requirements Engineering for Digital Health*, pages 133–154. Springer, 2015.

[3] DataLossDB. 2015 reported data breaches surpasses all previous years, 2015. https://blog.datalossdb.org/2016/02/11/2015-reported-data-breaches-surpasses-all-previous-years/.

[4] DoD. The United States Department of Defense cybersecurity culture and compliance initiative, September 2015. http://www.defense.gov/Portals/1/Documents/pubs/OSD011517-15-RES-Final.pdf.

[5] Golnaz Elahi, Eric Yu, and Nicola Zannone. A modeling ontology for integrating vulnerabilities into security requirements conceptual foundations. In *Proceedings of the 28th International Conference on Conceptual Modeling (ER)*, pages 99–114, 2009.

[6] Nicola Guarino. *Formal Ontology in Information Systems*. IOS Press, Amsterdam, 1st edition, 1998.

[7] Jianye Hao, Ensunk Kang, Jun Sun, and Daniel Jackson. Designing minimal effective normative systems with the help of lightweight formal methods. In *Proceedings of the 24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE)*, pages 50–60, 2016.

[8] HHS. Summary of the HIPAA privacy rule, 2003. United States Department of Health and Human Services (HHS). http://www.hhs.gov/ocr/privacy/hipaa/understanding/summary/.

[9] HHS. Breach portal: Notice to the Secretary of HHS breach of unsecured protected health information affecting 500 or more individuals, 2016. United States Department of Health and Human Services (HHS). https://ocrportal.hhs.gov/ocr/breach/.

[10] HIMSS. The healthcare information and management systems society (HIMSS) cybersecurity study, 2016. http://www.himss.org/sites/himssorg/files/2016-cybersecurity-report.pdf.

[11] Michael Howard and Steve Lipner. *The Security Development Lifecycle*. Microsoft Press, Redmond, WA, 2006.

[12] Marieta Georgieva Ivanova, Christian W. Probst, René Rydhof Hansen, and Florian Kammüller. Attack tree generation by policy invalidation. In *Proceedings of the Ninth International Conference on Information Security Theory and Practice (WISTP)*, pages 249–259, 2015.

[13] Özgür Kafalı, Nirav Ajmeri, and Munindar P. Singh. Revani: Revising and verifying normative specifications for privacy. *IEEE Intelligent Systems*, 31(5):8–15, September 2016.

[14] Özgür Kafalı, Nirav Ajmeri, and Munindar P. Singh. Kont: Computing tradeoffs in normative multiagent systems. In *Proceedings of the 31st Conference on Artificial Intelligence (AAAI)*, 2017.

[15] Özgür Kafalı, Munindar P. Singh, and Laurie Williams. Nane: Identifying misuse cases using temporal norm enactments. In *Proceedings of the 20th IEEE International Requirements Engineering Conference (RE)*, pages 136–145, Beijing, September 2016. IEEE Computer Society.

[16] Erik Kamsties. Understanding ambiguity in requirements engineering. In *Engineering and Managing Software Requirements*, pages 245–266. Springer Berlin Heidelberg, 2005.

[17] Peter Karpati, Andreas L. Opdahl, and Guttorm Sindre. Investigating security threats in architectural context: Experimental evaluations of misuse case maps. *Journal of Systems and Software*, 104:90–111, 2015.

[18] Abhay Kashyap, Lushan Han, Roberto Yus, Jennifer Sleeman, Taneeya Satyapanich, Sunil Gandhi, and Tim Finin. Robust semantic text similarity using LSA, machine learning, and linguistic resources. *Language Resources and Evaluation*, 50(1):125–161, 2016.

[19] Nadin Kökciyan and Pınar Yolum. Priguard: A semantic approach to detect privacy violations in online social networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2724–2737, Oct 2016.

[20] Divakaran Liginlal. HIPAA and human error: The role of enhanced situation awareness in protecting health information. In *Medical Data Privacy Handbook*, pages 679–696. Springer, 2015.

[21] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, pages 296–304, San Francisco, 1998. Morgan Kaufmann Publishers Inc.

[22] Yang Liu, Armin Sarabi, Jing Zhang, Parinaz Naghizadeh, Manish Karir, Michael Bailey, and Mingyan Liu. Cloudy with a chance of breach: Forecasting cyber security incidents. In *Proceedings of the 24th USENIX Conference on Security Symposium*, pages 1009–1024, 2015.

[23] Raimundas Matulevičius, Nicolas Mayer, and Patrick Heymans. Alignment of misuse cases with security risk management. In *Proceedings of the 3rd International Conference on Availability, Reliability and Security (ARES)*, pages 1397–1404, 2008.

[24] Einat Minkov and William W. Cohen. Graph based similarity measures for synonym extraction from parsed text. In *Proceedings of the 7th Workshop on Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 20–24. Association for Computational Linguistics, 2012.

[25] Sean Murphy. Is cybersecurity possible in healthcare? *National Cybersecurity Institute Journal*, 1(3):49–63, March 2015.

[26] Andreas L. Opdahl and Guttorm Sindre. Experimental comparison of attack trees and misuse cases for security threat identification. *Information and Software Technology*, 51(5):916–932, May 2009.

[27] Daniel Popescu, Spencer Rugaber, Nenad Medvidovic, and Daniel M. Berry. Reducing ambiguities in requirements specifications via automatically created object-oriented models. In *Proceedings of the 14th Workshop on Innovations for Requirement Analysis*, pages 103–124, 2007.

[28] Awais Rashid, Syed Asad Ali Naqvi, Rajiv Ramdhany, Matthew Edwards, Ruzanna Chitchyan, and M. Ali Babar. Discovering "unknown known" security requirements. In *Proceedings of the 38th International Conference on Software Engineering (ICSE)*, pages 866–876. ACM, 2016.

[29] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 448–453, 1995.

[30] Maria Riaz, Jonathan Stallings, Munindar P. Singh, John Slankas, and Laurie Williams. DIGS: A framework for discovering goals for security requirements engineering. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 35:1–35:10. ACM, 2016.

[31] M. Andrea Rodríguez and Max J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):442–456, 2003.

[32] K. Saruladha, G. Aghila, and S. Raj. A survey of semantic similarity methods for ontology based information retrieval. In *Proceedings of the Second International Conference on Machine Learning and Computing (ICMLC)*, pages 297–301, 2010.

[33] Pavel Shvaiko and Jérôme Euzenat. Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, Jan 2013.

[34] Munindar P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):21:1–21:23, December 2013.

[35] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D. Breaux, and Jianwei Niu. Toward a framework for detecting privacy policy violations in android application code. In *Proceedings of the 38th International Conference on Software Engineering (ICSE)*, pages 25–36. ACM, 2016.

[36] Amina Souag, Camille Salinesi, Raúl Mazo, and Isabelle Comyn-Wattiau. A security ontology for security requirements elicitation. In *Engineering Secure Software and Systems*, volume 8978 of *Lecture Notes in Computer Science*, pages 157–177. Springer, 2015.

[37] Jessica Staddon. Privacy incidents database: the data mining challenges and opportunities, November 2016. Cyber Security Practitioner.

[38] Verizon. Data breach investigations reports, 2016. http://www.verizonenterprise.com/verizon-insights-lab/dbir/.

[39] Andrew B. Williams, Anand Padmanabhan, and M. Brian Blake. Experimentation with local consensus ontologies with implications for automated service composition. *IEEE Transactions on Knowledge and Data Engineering*, 17(7):969–981, July 2005.

[40] Hui Yang, Alistair Willis, Anne De Roeck, and Bashar Nuseibeh. Automatic detection of nocuous coordination ambiguities in natural language requirements. In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*, pages 53–62. 2010.

[41] Koen Yskout, Riccardo Scandariato, and Wouter Joosen. Do security patterns really help designers? In *Proceedings of the 37th International Conference on Software Engineering (ICSE)*, pages 292–302, Florence, Italy, 2015. IEEE Press.