Implementing Database Access Control Policy from Unconstrained Natural Language Text

John Slankas
Department of Computer Science
North Carolina State University
Raleigh, USA
John.Slankas@ncsu.edu

***Abstract*—Although software can and does implement access control at the application layer, failure to enforce data access at the data layer often allows uncontrolled data access when individuals bypass application controls. *The goal of this research is to improve security and compliance by ensuring access controls rules explicitly and implicitly defined within unconstrained natural language texts are appropriately enforced within a system's relational database.* Access control implemented in both the application and data layers strongly supports a defense in depth strategy. We propose a tool-based process to 1) parse existing, unaltered natural language documents; 2) classify whether or not a statement implies access control and whether or not the statement implies database design; and, as appropriate, 3) extract policy elements; 4) extract database design; 5) map data objects found in the text to a database schema; and 6) automatically generate the necessary SQL commands to enable the database to enforce access control. Our initial studies of the first three steps indicate that we can effectively identify access control sentences and extract the relevant policy elements.**

***Index Terms*— Security, persistence, access control, role based access control, classification, natural language parsing, policy**

I. Technical Problem

Organizations face data security issues on a daily basis through both a need to protect information from unauthorized users as well as to comply with applicable regulations and standards. Many software systems establish access control only at the application level. If those controls are breached due to malicious activity or defective implementation, no additional layers of security exist to protect data. Insider threats exist if individuals can bypass application level security and access the database directly. Additionally, programmers can intentionally or inadvertently implement functionality that violates access control policies. A defense in depth security approach requires access control at *both* the application and data layers. However, creating and defining access control policies for databases can be a tedious, time-consuming, and error-prone endeavor. Developers must sift through existing documentation, application code, and database implementations.

The goal of this research is to improve security and compliance by ensuring access controls rules explicitly and implicitly defined within unconstrained natural language texts are appropriately enforced within a system's relational database.

To successful establish access control based from project documentation and other natural language texts, a number of issues must be solved:

1) Identifying natural language text that contain access control policies and database objects;
2) Extracting the various access control elements (subject, resources, actions) correctly from the text;
3) Extracting database design correctly from the text; and
4) Mapping resources found in the natural language text to specific database tables and columns.

II. Background and Related Work

This section reviews the work related to our proposed process.

A. Natural Language and Access Control

He and Antón [1] proposed an approach to generate access control policies from natural language based upon available project documents, database design, and existing policies. Utilizing a series of heuristics, humans would analyze the documents to find access control policies. In addition to heuristics to find the elements within the typical access control tuple (subject, resource, action), they created heuristics to identify policy constraints (temporal, location, relationship, privacy, etc.) and obligations. More recently, Xiao et al. [2] presented an approach, Text2Policy, where they parsed use cases to create eXtensible Access Control Markup Language[1] (XACML) policies. Their approach required use cases and the ability to match sentences to one of four specific sentence patterns to extract the subject, action, and resource for an access control policy. Using XACML as a target simplifies their process in that they did not have to map objects in the natural language text to existing application elements or database entities. Our process utilizes machine learning to make a decision as to whether or a sentence involves access to control and then to discover sentence patterns from which the access control policies can be extracted. We then implement those policies within an application's database(s).

[1] http://www.oasis-open.org/committees/xacml

978-1-4673-3076-3/13 © 2013 IEEE

ICSE 2013, San Francisco, CA, USA
Doctoral Symposium

B. Controlled Natural Language

Other researchers have resolved converting natural language to and from policies through a controlled natural language (CNL). Schwitter [3] defines CNLs as "engineered subsets of natural languages whose grammar and vocabulary have been restricted in a systematic way in order to reduce both ambiguity and complexity of full natural languages." While CNLs provide consistent, semantic interpretations, CNLs limit authors and typically require language-specific tools to stay within the language constraints. Project documents previously created cannot be used as inputs without processing the documents manually into the tools. Policy authored outside of tools must confirm to strict limited grammars to be automatically parsed. Brodie et al. [4] used this approach in the SPARCLE Policy Workbench. By using their own natural language parser and a controlled grammar, they were effectively able to translate from natural language into formal policy. Inglesant et al. [5] demonstrated similar success with their tool, PERMIS, which utilized a RBAC authorization model. Recently, Shi and Chadwick [6] presented their results of an application to author access control policies using a CNL. While they showed the improved usability of CNL interface, they were limited in the complexity of the policies that could be created as the interface did not support conditions or obligations. Our process removes the CNL constraint, working against original, unconstrained texts.

C. Schema and Ontology Matching

One challenge with the approach is to map resources found in the natural language text to database tables and columns. This problem is similar to the challenges of matching one database schema to another database schema as well as matching different ontologies. Multiple survey papers have been written on database schema matching [7, 8] and ontology matching [9, 10]. Research continues in this field with such recent work as Po and Sorrentino [11] who utilized probabilistic techniques to derive lexical relationships and disambiguate word sense across different ontologies. Our initial solution utilizes an approach based upon word similarity. We have no plans for further research in this area.

D. Databases from Natural Language

Different approaches exist to discover database entities from natural language. In 1983, Chen [12] published a series of heuristics to create entity relationship diagrams from natural language. Hartman and Link [13] extended this work in 2007 through additional heuristics and refinements. Both approaches require humans to analyze the text. Omar [14] applied semantic role labeling techniques to automatically generated database design from a series of heuristics. Another approach [15, 16] comes from research in natural language queries of databases. To effectively perform natural language queries, a system must first translate the user's request to a symbolic representation, map the representation to the physical database, and then determine the requested data and query conditions. Our will use Hartman's and Link's heuristics to bootstrap a supervised machine learning approach to extract a database design from the natural language documents.

E. Semantic Relationship Extraction

To extract policy information from the natural language text, we will utilize semantic relationship extraction. A number of different ways exist to identify semantic relationships within text. Snow et al. [17] utilized word patterns and grammatical relationships within a sentence to discover "is-a" type relationships among words. Alan Akbik and Jürgen Broß [18] used a similar process to extract a wide range of sematic relations with the goal to extract arbitrary relations for semantic search. We plan to utilize word patterns and grammatical relationships to extract the access control policy elements from sentences as well as to extract the database design. We will incorporate machine learning algorithms such that the process can learn and apply new patterns as more and more natural language text has been evaluated.

III. Proposed Solution

We propose a six-step process, Role Extraction and Database Enforcement (REDE), to extract roles, database objects, and other security elements from natural language text and generate database-enforced access control policies for applications that utilize relational databases.

For input, the process takes natural language text, a physical database schema, existing database role names, and, optionally, a list of domain-specific words. Any number of existing project documents such as requirements, use cases, designs, test scripts, and training material can serve as the natural language text. The process outputs the SQL commands to establish role-based access control in a database. Additionally, the process can generate consistency and traceability reports.

The REDE process consists of six primary steps:

1. *Parse natural language*. In this step, sentences are converted into an internal representation and analyzed for domain specific keywords and elements found in sentences already processed. We will use Stanford's Natural Language Parser[2], but then add additional analysis to discover various attributes related to access control policy such as negativity. Additionally, we apply a compact document grammar to identify key features such as section headings and lists within texts.
2. *Classify sentence*. Next, we use a classifier to examine whether the sentence relates to access control or database design.
3. *Extract access control elements*. The process finds subjects, actions, and objects based upon learned semantic patterns. Additionally, if specific conditions exist for the access, the necessary elements to enforce those constraints will be extracted.

[2] http://nlp.stanford.edu/software/

4. *Extract database design*. This step runs concurrently with the previous step and looks for semantic patterns based to extract database entities, attributes, and relationships.
5. *Map resources to database objects*. This steps maps the database design extracted from the previous step to the actual objects used in the application's database(s).
6. *Generate SQL Commands*. Based upon the extracted access control elements, the system generates the necessary commands to generate role-based access control within a relational database. As necessary, for some conditional-based access, views will be created and utilized to ensure the access control policy is appropriately implemented.

The process produces ancillary benefits as well. Access control policies are traceable back to their document sources. Policy conflicts in which a subject has contradictory permissions can be detected within the documentation. Current access control policies can be validated against those produced by the process.

IV. Research Questions

A. Does the process correctly find natural language sentences indicating access control policy or database design?

The process relies upon the ability to successfully identify relevant sentences that contain access control policies or database design. Only those relevant sentences need additional processing to extract the access control policy elements. Additionally, an organization benefits immediately from the identification as the relevant document sections can be extracted for further manual analysis as needed.

B. Does the process correctly extract access control policy elements from the natural language text with semantic relation extraction?

The process must reliably and automatically extract subjects, actions, and resources from sentences to create access control policies.

C. Does the process correctly extract database design from the natural language text with semantic relation extraction?

Just extracting the access policies is not sufficient as the policies must be implemented in the target environment. This question evaluates how well we identify and reconstruct that environment from the natural language text.

D. Does the process correctly implement access control within a system's relational database(s)?

Finally, we need to evaluate how effectively the access control is implemented within the application. Most importantly, does this process prevent users from gaining unauthorized access to data if application controls are bypassed? Do all of the databases tables have access control policies? Does the application correctly function? How much effort was required to alter the application to use the access control policies generated by this process?

V. Methodology and Evaluation

To address the research questions, we will perform three studies. We will initially focus on a single problem domain (electronic health records), but then address other domains to examine the generalizability of the REDE process.

A. Identification Study

In this first study, our goal is to research the ability to classify natural language sentences for access control and database design. We will first gather relevant documents for open source health care systems. We will also utilize other documents such as regulatory texts and industry guidelines. Next, we will label each sentence as to whether or not it contains access control or data design. If either one of those elements are present, we will also label the relevant words that form the access control tuple or imply database design. From this labeled corpus, we will train and test a variety of machine learning classifiers. We will also study the impacts of different attributes affect classification performance. We measure the classifier performance based upon precision and recall values.

B. Database Design Extraction Study

In our next study, we research how database design can be effectively extracted with word patterns and grammatical relationships. As a starting point, we will use the heuristics developed by Chen, Hartman, and Link [12, 13] to discover additional patterns and relationships for database design. Once the set of patterns has been developed, we will then use those patterns to extract the database design from the natural language text. For evaluation, we will measure how effectively we extract tables, relationships among tables, and attributes from the text as compared to both the text and the system's actual database schema.

C. Access Control Extraction Study

In our final study, we evaluate how the access control can be effectively extracted from the natural language text with word patterns and grammatical relationship. As with the previous study, we will utilize a bootstrapping process from an initial set of access control patterns to discover additional patterns throughout the evaluated documents. The study will culminate in the complete implementation and subsequent test of the generated role based access control within a system's relational database.

VI. Current Progress

We have implemented an initial version of the process in our work [19]. Analyzing a public requirements specification, our classifier correctly predicted 78% of the access control statements while recalling 92% of the applicable statements. The requirements consist of 40 use cases plus additional non-functional requirements, constraints, and a glossary. The version we used contained 1114 sentences with 409 (36.7%) of those sentences classified as access control. We utilized a stratified 10-fold cross validation in this work as we only evaluated the specification itself.

VII. Proposed Work

Our next step in our work is to utilize the word patterns and grammatical relationships to extract the database design from natural language text. We plan to perform this on the system we have already examined as well as utilizing the documents Chen, Hartman, and Link have used in their work. Then, we will revisit our policy extraction process to use the word patterns and grammatical relationships to identify subjects, actions, resources, and conditions. We will then complete the process with generating the necessary SQL commands and re-evaluate our initial system. From there, we will evaluate the process against another system in the same domain, followed by system(s) in other domains.

VIII. Expected Contributions

We expect this work to have the follow contributions:

- Process to classify natural language statements as access-control related
- Process to extract access control elements via semantic relation extraction with machine learning
- Process to extract database design from natural language text via semantic relation extraction.
- Publically-available labeled corpus

For practitioners, we plan to develop an open-source tool to incorporate into any software engineering methodology our process to generate database access control from unconstrained natural language text.

Acknowledgment

This work was supported by the U.S. Army Research Office (ARO) under grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI). Thank you to the North Carolina State University Realsearch group and the anonymous reviews for their helpful comments on the paper. Most especially, I would like to thank my advisor, Laurie Williams, for her guidance and support of my work.

References

[1] Q. He and A. I. Antón, "Requirements-based Access Control Analysis and Policy Specification (ReCAPS)," *Information and Software Technology,* vol. 51, pp. 993-1009, 2009.

[2] X. Xiao, A. Paradkar, S. Thummalapenta, and T. Xie, "Automated Extraction of Security Policies from Natural-Language Software Documents," in *International Symposium on the Foundations of Software Engineering (FSE)*, Cary, North Carolina, USA, 2012.

[3] R. Schwitter, "Controlled Natural Languages for Knowledge Representation," in *Proceedings of the 23rd International Conference on Computational Linguistics*, ed. Beijing, China: Association for Computational Linguistics, 2010, pp. 1113-1121.

[4] C. a. Brodie, C.-M. Karat, and J. Karat, "An Empirical Study of Natural Language Parsing of Privacy Policy Rules Using the SPARCLE Policy Workbench," *Proceedings of the second symposium on Usable privacy and security - SOUPS '06,* p. 8, 2006.

[5] P. Inglesant, M. A. Sasse, D. Chadwick, and L. L. Shi, "Expressions of Expertness: The Virtuous Circle of Natural Language for Access Control Policy Specification," in *Proceedings of the 4th symposium on Usable privacy and security*, ed: ACM, 2008, pp. 77-88.

[6] L. Shi and D. Chadwick, "A Controlled Natural Language Interface for Authoring Access Control Policies," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ed. TaiChung, Taiwan, 2011, pp. 1524-1530.

[7] E. Rahm and P. a. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal,* vol. 10, pp. 334-350, 2001.

[8] A. H. Doan and A. Y. Halevy, "Semantic Integration Research in the Database Community : A Brief Survey," *AI magazine,* vol. 26, p. 83, 2005.

[9] N. Choi, I. Y. Song, and H. Han, "A Survey on Ontology Mapping," *ACM Sigmod Record,* vol. 35, pp. 34-41, 2006.

[10] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: the state of the art," *The Knowledge Engineering Review,* vol. 18, pp. 1-31, 2003.

[11] L. Po and S. Sorrentino, "Automatic generation of probabilistic relationships for improving schema matching," *Information Systems,* vol. 36, pp. 192-208, 2011.

[12] P. P.-S. Chen, "English Sentence Structure and ER Diagrams," *Information Sciences,* pp. 127-149, 1983.

[13] S. Hartmann and S. Link, "English Sentence Structures and EER Modeling," in *Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling* vol. 67, ed. Ballarat, Australia: Australian Computer Society, Inc., 2007, pp. 27-35.

[14] N. Omar, R. Hassan, and H. Arshad, "Automation of database design through semantic analysis," in *Proceedings of the 7th WSEAS International Conference on Computational Intelligence, Man-machine Systems and Cybernetics*, ed. Cairo, Egypt, 2008, pp. 71-76.

[15] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, "Natural Language Interfaces to Databases - An Introduction," *Journal of Natural Language Engineering,* vol. 1, pp. 29--81, 1995.

[16] A. Popescu, A. Armanasu, O. Etzioni, D. Ko, and A. Yates, "Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability," in *Proceedings of the 20th international conference on Computational Linguistics*, ed. Geneva, Switzerland, 2004, pp. 141-148.

[17] R. Snow, D. Jurafsky, and A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," presented at the Advances in Neural Information Processing Systems (NIPS 2004), Vancouver, British Columbia, 2004.

[18] A. Akbik and J. Broß, "Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns," presented at the Workshop on Semantic Search, Madrid, Spain, 2009.

[19] J. Slankas and L. Williams, "Extracting Database Role Based Access Control from Unconstrained Natural Language Text," North Carolina State University, Raleigh, North Carolina, USA, 2012.