Towards a Hybrid Attribute-based Access Control (ABAC) Engineering Framework

Manar Alohaly
University of North Texas, Denton, TX, USA
ManarAlohaly@my.unt.edu

Daniel Takabi
Georgia State University, Atlanta, GA, USA
Takabi@gsu.edu

Abstract

The development of an attribute-based access control (ABAC) system requires a substantial degree of manual effort to derive a set of appropriate machine-readable policies. To reduce the development costs, two primary approaches have been proposed, namely bottom-up policy mining and automated top-down policy engineering. Major shortcomings of these approaches, respectively, are generating policies irrelevant to organization needs and overlooking information of existing access control system. In this work, we propose a hybrid ABAC policy engineering approach to combine the benefits and address the shortcomings of bottom-up and top-down policy engineering in a systematic framework. The novelty of our approach is in generating a machine-readable ABAC policy that conforms with the existing access control of the system and is consistent with its authorization requirements as originally expressed in natural language form.

Problem Statement

Manually crafting ABAC rules can be difficult [1] and certainly expensive task [2]. Therefore, a major challenge is to effectively automate this process and reduce costs associated with the manual execution. Two primary approaches have been studied to help with cost reduction goal, namely automated top-down and bottom-up policy mining ABAC policy engineering.

Top-down approaches focus on the analysis of the organizational business processes, system's use-cases and authorization requirements, which are often written in natural language, in order to work them into machine readable ABAC policies [3-5]. However, the difficulty of the problem and the limitations of current natural language processing techniques restrict the capabilities of this approach.

The bottom-up approaches, on the other hand, utilize information about existing access control model and attribute data to partially automate the construction of ABAC policies. While this approach has the advantage of constructing a fully-specified formal policy, it still faces the challenge of generating semantically meaningful and noise-free ABAC policy. That is, the mined rules can be irrelevant to the actual requirements and unintuitive for security administrators who have to post-process the resulting rules.

Motivation

Since full automation of ABAC policy development is beyond the current state of the art, it is worthwhile to study how to aggregate the benefits of current approaches.

Contributions

The main research contributions of this work are: (1) Designing a hybrid ABAC policy engineering framework. At the core of our proposed framework is a natural language policy analysis component interacting with a bottom-up policy miner for ABAC rules generation and refinements. (2) Designing a similarity metric that matches mined rules with natural language policy sentences they implement.

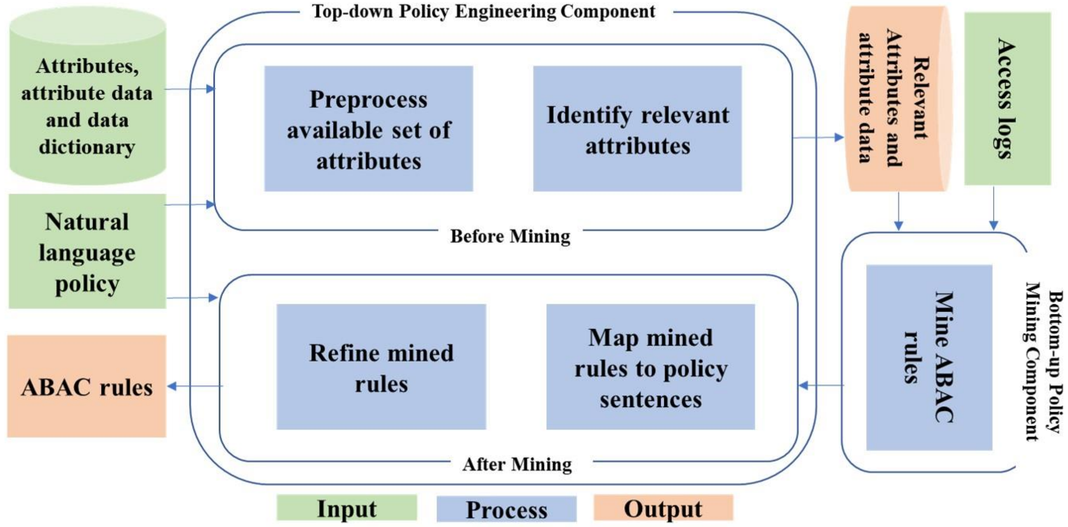

Figure1. High level overview of the proposed hybrid ABAC engineering framework

Framework Design

The core idea of our proposed framework centers around using the organizational policy as a point of reference to determine the proper input to the miner and to enhance its output. Particularly, as illustrated in Figure 1, the proposed process begins with analyzing organizational policy along with the attribute data from existing databases to determine attributes necessary for the construction of ABAC rules. After identifying the set of relevant attributes, it is fed to the miner to produce the first draft of formal ABAC rules. Next, the process proceeds to improve the mined rules to be as close as possible to the actual organizational policy. This entails mapping mined rules to policy sentences they likely implement, comparing both representation and refining the mined rules accordingly.

As input, our hybrid ABAC policy engineering framework accepts the tuple $<U,R,O,A_u,A_r,d_u,d_r>$, where U, R, O, A_u, A_r, d_u and d_r denotes the set of users, resources, operations, user attributes, resource attributes, user attribute data, and resource attribute data, respectively.

Besides, it also requires the data dictionary DD defining the meanings of attribute data, access log L and the organizational policy N LPs in its natural language form. The goal is to produce ABAC rules that conform with the organizational policy using a combination of low-level information of the existing system and a high-level organizational policy.

As illustrated in Figure 1, the pipeline of our framework consists of five modules. It begins with a preprocessing step to establish the link between the low-level and high-level information. Then, it proceeds towards four other tasks of identifying relevant attributes, mining ABAC rules, mapping mined rules to natural language policy and ends with a rules refining step.

Research Question

How well are the rules, generated by our hybrid approach, in terms of their similarity to the ground truth?

Results

We run the framework on 68 policy sentences of IBMApp [6] dataset using 29 attributes with 334 distinct attributes values and 62 rules. Of the 29 attributes, only 14 attributes are relevant to the expression of rules.

To evaluate rules produced by the proposed framework, we employ Jaccard Similarity Metric to capture the similarity between generated rules and the truth as in [7]. Generally, Jaccard similarity, denoted as J , of the sets S_1 and S_2 is defined as:

$$J(S_1,S_2)=\frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

The Jaccard similarity of rules $(e_u;e_r;O;c)$ and $(e'_u;e'_r;O';c')$ is the average of the similarities of their components, i.e., the average $J(e_u, e'_u)$, $J(e_r, e'_r)$, $J(O, O')$ and $J(c, c')$.

Our hybrid approach achieves a similarity score of 0.71. While this is not ideal, it is a significant improvement over the 0.40 achieved using pure bottom-up miner. This represents an improvement rate of 77.5% over a pure bottom-up policy engineering approach.

References

[1] Matthias Beckerle and Leonardo A Martucci. 2013. Formal definitions for usable access control rule sets from goals to metrics. In Proceedingsof the Ninth Symposium on Usable Privacy and Security. ACM, 2

[2] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J. Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, 2013. Guide to attribute-based access control(ABAC) definition and considerations (draft).NIST special publication800, 162 (2013)

[3] Alohaly, M., Takabi, H., Blanco, E.: A deep learning approach for extracting attributes of abac policies. In: Proceedings of the 23rd ACM Symposium on Access Control Models and Technologies (2018)

[4] Alohaly, M., Takabi, H., Blanco, E.: Towards an automated extraction of abac constraints from natural language policies. In: IFIP International Information Security Conference (2019)

[5] Alohaly, M., Takabi, H., Blanco, E.: Automated extraction of attributes from natural language attribute-based access control (ABAC) Policies. In: Cybersecurity 2 (1), 2 (2019)

[6] IBM. 2004. Course Registration Requirements. https://khanhn.files.wordpress.com/2016/08/vidu-ibm.pdf

[7] Zhongyuan Xu and Scott D Stoller. 2014. Mining attribute-based access control policies from logs. In IFIP Annual Conference on Data and Applications Security and Privacy. Springer, 276–291