Jiajie Li

CSE 300

30 October 2016

# Managing Computer Access with User Attributes

Access controlis an important topic in the field of computer security. People store and manage important data using computers, but without strong security data could potentially be stolen and manipulated for malicious purposes. Access control manages access of users, gives users permissions to perform actions on resources, and makesapproval or denial on access based on access records. It is important for companies and government agencies that manage large amounts of data, because data are accessed regularly, and it's crucial to ensure thatthe right people can perform right actions on the right resources. One concern with access control is its expense to implement and operate. When large amounts of data and users are involved, access control policies usually get complicated, and they become moredifficult to manage and more error-prone. The process to identify users and approveor deny access shouldn't take too much time, and this depends on the implementation of access control policy as well. There are many access control models developed in order to reduce the expense and increase the efficiency of the authorization. Attribute-Based Access Control (ABAC) is one of these recent models, and ithas gained popularity in research and study because of its unique properties and benefits.

The most popular access control policy is the traditional Access Control List (ACL). In ACL model, each resource has its own list of users and their privileges, and users are given permissions and access based on this list. Over the years Role-Based Access Control (RBAC) is developed as an improvement of ACL, and has gained some popularity as well. RBAC uses role

assignments of users to determine permissions, and ABAC uses attributes of users to determine permissions instead.

ABAC model was first proposed and recommended by the Federal Chief Information Officers Council (CIO) in FICAM Roadmap and Implementation Guidance Version 2.0.In this document, the CIO aimed to improve security and efficiency foridentity, credential, and data management (ICAM) of Federal Enterprise, which conducted electronic business between Federal agencies as well as general public (Federal Chief Information Officers Council3).It provided designs, models, and architecturesfor ICAM programs, and proposed them as solutions toseveral IT challenges such as cyber security and data management.The CIO discussed Access Control Models in Authorization (286-290), and suggested a list of models, each with own benefits and limitations, that it had recommended to government agencies. This list included the traditional ACL, its successor RBAC, and the newly developed ABAC. ABAC, as described by the CIO, focused on attributes, the characteristics and descriptions, of users and resources, and required no additional input or knowledge for creating rules. Therefore, the CIO considered ABAC to be highly adaptive and flexible, and especially efficient for applications with frequent user changes. According to the CIO, one of ABAC model's largest limitations was the time of implementation,since ABAC model correlated information and attributes from multiple sources to create rules for potential relating users, the process would be time-consuming (288). The CIO didn't state which model was the best at practice but it recommended agencies to use a hybrid approach to combine different models depending on situations. Despite being a recent model without many developments or studies, ABAC model was still recommended by the CIO, and this showed that the CIO had evaluated and consideredit to be effective, secured, and trustworthy for government agencies to utilize. However, one issue with ABAC, as the CIO pointed out, was

that it wasn't supported by common operating systems, so there weren't many existing implementations. At the same time, the CIO only listed these models with their conceptual approaches, each with their advantages and disadvantages,but didn't provide any guidance on their actual implementations or operations. Therefore, ABAC model still needed further research and developmentbefore it could be widely adopted and utilized.

Soon after the recommendation of ABAC model, National Institute of Standards and Technology (NIST) published the Guide to ABAC Definition and Considerations. In this document, Hu, et al provided the definition and considerations of ABAC.They defined that ABAC used different attributes to identify users, and in similar ways, ACL used the attribute of "identity" and RBAC used the attribute of "role". Although all 3 models had similarities, the main difference with ABAC was that it used only True orFalse to evaluate attributes. It also emphasizedone issuewith ACL and RBAC that wasn't present in ABAC policy, which was the difficultly to identify rules to update whenthere were changes (Hu, et al.5). This is mentioned in the FICAM Roadmap and Guidance, as ithighlighted the strength of ABAC for its flexibility to changes (Federal Chief Information Officers Council 288). According to Hu, et al, some other benefits of ABAC were ease of management and support of multi-factor decisions.They specifically stated that RBAC had limitations in supporting multi-factor decisions, as RBAC made decision solely based upon role assignments, or user associations with roles, and it had difficultly to consider other factors (Hu, et al. 6).Moreover, ABAC avoided the need of explicit rule between users and resources, and it enabled flexibility on enterprise level where management of rules were time consuming and complex.In the conclusion by Hu, et al,they concluded that ABAC could implement and migrate existing RBAC policies, but with the downsides of more costly implementation and maintenance due to its complexity (33). In this

document, the authorsdefined ABAC by comparing with ACL and RBAC, and it used them to present practical ABAC implementations. The comparisons help to emphasize the benefits of ABAC, and to better understand the operations of the model. Although ABAC brought flexibility and better performance, the authorsstillurgedorganizations to take serious consideration on the cost to develop and maintain, and evaluate all benefits and risks, before introducing ABAC policies into their systems. Theyacknowledged the difficulty to implement ABAC, the same issue that is stated in the FICAM Roadmap and Guidance as well. Due to its methods of operating, ABAC would become more complicated as users and resources grow, and this would be reflected in the implementation process.

Since the proposal of ABAC, it has been constantly compared with RBAC, which is an older access control model that has been widely adopted.In the article by Coyne and Weil, they discussed the possibilities of combining ABAC and RBAC to obtain a better model. Theystated the simplicity for ABAC, that users would be granted access if they had the attributes allowed by the resources they wanted to access. Comparing to RBAC, the identities or roles of the users must be evaluated first, then access was granted based on the result. However, they considered it wasn't practical for ABAC to audit user access to permissions, because all permissions were given based on attributes and attributes wouldn't define anything until they were associated with users and resources(Coyne and Weil 15). After comparing both models andstatingthe disadvantages of both models, Coyne and Weil suggested to combine both judiciously in order to obtain the flexibility of ABAC and auditing of RBAC. They gave the suggestion to use RBAC with attributes instead of ABAC with role name, because they believed ABAC couldn't audit user access to certain permissions, which RBAC would solve with its pre-defined role assignments. The combination of ABAC and RBAC isn’t mentioned in previous documents, but

the CIOhas suggested agencies to combine models to preserve their features for a better result (290).Coyne and Weil were able to suggest a combination of ABAC and RBAC with rationalreasons, but again they only provided details on a conceptual level without any implementation specifics. To evaluate whether this combination is indeed an effective model, it needs to be implemented and testedwith real data, which is already considered to be difficult because of ABAC's complexity.

Although ABAC comes with some great advantages, it stills faces issues during implementation.In both the FICAM Roadmap and Guidanceand the ABAC Definition and Consideration, the authorsemphasized the cost and expense to implement the ABAC model. Manual implementation of any access control models are always difficult and expensive, as it requires solid knowledge in subject area and long time to write each rule. Therefore, it's better to automate the process, with the help of policy mining. In the research paper by Stoller and Xu,they developed an algorithm for ABAC policy mining, which they hoped to overcome the disadvantages of implementing ABAC. The algorithm was able to mine ABAC policies from existing ACLs with attribute data, as well as from RBAC policy by expanding them into ACLs and adding "role" attribute for role assignments. Stoller and Xu first defined an ABAC policy language to use as the result of algorithm, then presented the algorithm that translated an ACL policy to ABAC policy. The algorithm was able to produce an ABAC policy from given inputs, by combining each constraint with users and resources from ACL, then generalizing and simplifying each result. They concluded that the algorithm was effective as it was able to producepolicies that were very similar to the input policies. According to Xu and Stoller, this was the first known ABAC mining algorithm (13).Although they provided examplesto showthe algorithm was able to automate the implementation of ABAC policy, it wasn't able to compare it

with others for efficiency. Another issue is that Stoller and Xu claimed that they didn't evaluate policies from real organizations, but instead created sample policies from real-world case studies. The sample policies had relative small sizes, but they stated this was intended in order to only resemble interesting rules in the policies (Stoller and Xu 7). Therefore, it will be more ideal to run the algorithm on real world policies, as well as on large size policies to test its efficiency on larger input.

ABAC has been treated as a solid model that can be widely adopted, but under some situation it isn't the ideal model to use. In the paper by Bogarerts, et al, they proposed the Entity-Based Access Control (EBAC), which they claimed could support more expressive policies. Theyclaimed that ABAC couldn't reflect relationships as attributes easily. ABAC determined permissions based on users' attributes, however, if the attributes involved user relationships, it could become complex and difficult to update (Bogarerts, et al. 291). Therefore, Entity-Based Access Control was proposed as a solution to this issue, because EBAC supported comparison of attributes, as well as traversed relationships of users to determine permission. Bogarerts, et alpresented a disadvantage of ABAC under a specific circumstance, which isn't discovered in other papers or documents. Although the model is expected to grow in popularity, it still faces some issues that can be solved by expanding or combining with other models.

Since ABAC was recommended by the Federal Chief Information Officers Council, it has becomea major trend in access control policy study and research. In some of the researches, ABAC is considered to be more flexible than some older models, and issuggested to replace or combine with them to create better access control policies.However, it is still not perfect and has its own disadvantages in different situations. Therefore, in order to fully utilize ABAC's distinguished benefits, it still requires more development to become a solid and efficient model,

and in the meantime, invent more advanced models that retain ABAC's advantages and overcome its disadvantages.

Works Cited

Federal Chief Information Officer Council and Federal Enterprise Architecture. “Federal Identity Credential and Access Management (FICAM) Roadmap and Implementation Guidance.”Version 2.0, 2 Dec. 2011.

Hu, Vincent, et al. “Guide to Attribute Based Access Control (ABAC) Definition and Considerations.” *NIST Special Publication 800-162*. National Institute of Standards and Technology, Jan. 2014.

Coyne, Ed and Timothy Weil. “ABAC and RBAC: Scalable, Flexible, and Auditable Access Management.”*IT Pro May/June 2013*.IEEE Computer Society, 2013, pp. 14-16.

Stoller, Scott and Zhongyuan Xu. “Mining Attribute-Based Access Control Policies.”*IEEE Transactions on Dependable and Secure Computing*.IEEE Computer Society, Sep. 2015.

Bogaerts, Jasper, et al.“Entity-Based Access Control: supporting more expressive access control policies.” *Proceedings of the 31st Annual Computer Security Applications Conference*. ACM, Dec. 2015, pp. 291-300.