Position Paper: Towards a Moving Target Defense Approach for Attribute-based Access Control

Carlos E. Rubio-Medrano, Josephine Lamp, Marthony Taguinod,
Adam Doupé, Ziming Zhao and Gail-Joon Ahn
Arizona State University
{crubiome, jalamp, mtaguino, doupe, zzhao30, gahn}@asu.edu

ABSTRACT

In recent years, *attribute-based access control* has been recognized as a convenient way to specify access mediation policies that leverage attributes originating from different security domains, e.g., independently-run organizations or supporting platforms. However, this new paradigm, while allowing for enhanced flexibility and convenience, may also open the door to new kinds of attacks based on forging or impersonating attributes, thus potentially allowing for attackers to gain unintended access to protected resources. In order to alleviate this problem, we present an ongoing effort based on *moving target defense*, an emerging technique for proactively providing security measurements: we aim to analyze attribute-based data obtained at runtime in order to dynamically change policy configurations over time. We present our approach by leveraging a case study based in *electronic health records*, another trending methodology widely used in practice for mediating access to sensitive healthcare information in mission-critical applications.

Keywords

Attribute-based Access Control; Moving Target Defense; Electronic Health Records; Policy Mutation

1. INTRODUCTION

Recently, *attribute-based access control* (ABAC) [1], has attracted the interest of both academia and industry as a convenient means of protecting computer systems from security-related incidents. As ABAC evolves into a mature paradigm and various implementations are successfully deployed in practice, attributes originating from different sources may be leveraged for expressing rich policies that better meet the specific needs of customized environments [4]. Such a paradigm, while allowing for enhanced flexibility and convenience, may also introduce non-trivial security vulnerabilities. As an example, consider an ABAC policy managed by an organization A that leverages attributes from an outside independently-run organization B, in such a way that end-users holding attributes issued by B can safely access the resources being shared by A. In such a setting, the policy makers in A should somehow *trust* the way attributes are created and assigned to end-users in the context of B. However, as time evolves, such an assumption may not always hold in practice, as organization B may be the subject of security incidents itself, e.g., hacking, or may not have a strict control on the way its attributes are created and distributed. This would potentially allow for malicious third parties to *compromise* them, for instance, by means of a well-crafted forgery process.

In this paper, we describe an ongoing effort to alleviate this problem by leveraging an approach inspired from *moving target defense* (MTD) [2], a promising paradigm based on the idea of proactively changing, e.g., *moving*, system configurations in an effort to deter potential future attacks. In our approach, we aim to analyze attribute information collected from runtime traces of mission-critical applications, a.k.a., the *attribute bag*. Taking an ABAC policy as an input, our approach first obtains the list of *original* attributes listed in such policy and subsequently inspects the attribute bag in an effort to locate attributes that are *correlated* to the *original* ones. Later on, these *newly-extracted* attributes are used to enhance the original policy, producing a new policy that is forwarded to the access mediation infrastructure for enforcement. The intuition behind our approach is that the entities involved in a given access request, e.g., end-users and protected resources, typically exhibit additional trusted attributes besides the ones listed in the original policy. This way, if the original attributes are compromised, the newly-extracted ones, which we assume stay uncompromised, may still deter the unintended exploitation of the original policy. In addition, we aim to mitigate the harm to usability, e.g., end-users no longer able to access previously-available resources, by striving to obtain a high degree of correlation between the original attributes and the newly-extracted ones.

We present our approach in the context of an emerging application domain: *electronic health records* (EHRs) [3], which has also become the focus of many implementations in practice as well as in research endeavors due to its notable benefits of providing better quality of patient care. We show how attributes belonging to both patients and healthcare providers, e.g., doctors and nurses, collected from both EHRs and access logs can be leveraged to provide stronger security guarantees by means of our approach, as it allows for correlated attributes to be discovered and later used to provide enhanced policies that can prevent future attacks,

ABAC'16, March 11 2016, New Orleans, LA, USA
© 2016 ACM. ISBN 978-1-4503-4079-3/16/03...$15.00
DOI: http://dx.doi.org/10.1145/2875491.2875499

without affecting the overall usability of a given EHR system.

This paper is organized as follows: we start by briefly reviewing some important background topics, along with a running example and some other key considerations for our approach in Section 2. Our proposal is later described in Section 3, and we finalize this position paper by outlining the status of our research as well as future work in Section 4.

2. BACKGROUND

Moving target defense (MTD) [2] is an emerging paradigm for providing security guarantees by proactively changing, e.g., *moving*, the configurations of a protected system. Opposed to traditional approaches, which assume security configurations remain *immutable*, MTD strives to reduce the possibility of a successful attack by negating any advantages the attacker may have. For instance, complicating the *reconnaissance* process in which an attacker gathers information about the current configurations of the victim system; or by deterring ongoing attacks that were crafted based on previously-discovered (and later changed) configurations. In addition, effects to the *usability* of the protected system, e.g., response time and end-user access patterns, should be minimized, in an effort to prevent runtime inconveniences that may complicate the adoption of MTD-based techniques.

Attribute-based access control (ABAC) [1] is also a trending technique for mediating access to sensitive resources within a computer system. Fig. 1 presents a depiction of the model considered for the purposes of this paper. Besides the traditional sets of attributes, permissions and access entities that have been previously discussed in the literature, our approach includes the concept of *attribute sources* and *policy makers*. The former are in charge of defining, creating and assigning attributes to access entities, whereas the latter are in charge of crafting policies by establishing a relationship between attributes and permissions. Our model depicts an *open world* scenario where attributes originating from different sources may become relevant under the security domain that is defined by the policy makers. However, despite being represented as different sets in Fig. 1, the sets of attribute sources and policy makers may not necessarily be mutually exclusive in practice, as a policy maker may also play the role of an attribute source when managing attributes defined within a certain security domain. In order to safely leverage attributes, policy makers should somehow trust their corresponding sources. As an example, an attribute whose value can be deliberately modified by the assigned entity may not provide strong security guarantees, as such an entity may be allowed to modify the attribute's value at will to meet the requirements defined in a given policy. Therefore, policy makers should have confidence on the attribute creation and assignment process carried out by the sources. In our model, we assume such *attribute trust* score can be modeled as a binary value in the set {0,1}. This way, only attributes depicting a trust score of 1 may be safely used for policy crafting.

Electronic health records (EHRs) [3] increase the efficiency of healthcare organizations by improving communication between clinicians and other health institutions leading to improved continuity and coordination of care. They also support clinician decision making by providing comprehensive information about patients, thereby increasing quality of patient care. Complete and relevant information must be accessible to clinicians in a timely manner on a need-to-know only basis within the set of privileges allowed by the patient, while unauthorized accesses to private data must be prevented.

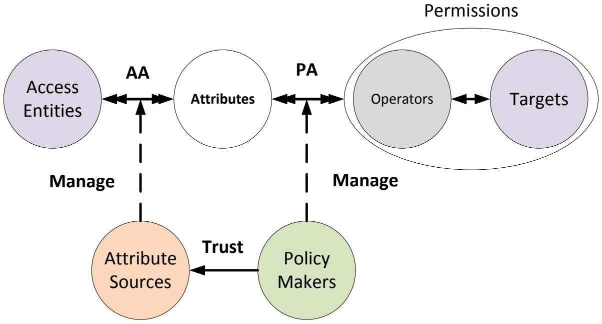

Figure 1: A depiction of an ABAC model: attributes are related to access entities (e.g., end-users and protected resources) by means of the *attribute assignment* (AA) relation. Access rights (permissions) are in turn related to access entities by the *permission assignment* (PA) relation. Policy makers are in charge of establishing the PA relation by leveraging the attributes provided by the attribute sources, who are in charge of managing the AA relation.

Table 1 shows an extract of attribute-based and access log data depicting EHRs from different users. As an example, the entry depicted in the first row shows an access request to the EHR belonging to a user identified by the attribute *PatientID* with a value of 11234. Such an access request was denied, as shown by the value of the *Decision* attribute set to *False* in the last column. Other attributes are shown in Table 1 for illustrative purposes, and will be further discussed, along with their corresponding coloring scheme, in Section 3.

3. OUR APPROACH

As described in Section 1, we aim to develop an approach based on MTD theory in such a way attacks to attribute-based policies can be effectively prevented. With this in mind, we first describe the attack model we take into consideration, followed by a description of our proposed approach and finalize with a short discussion on how our solution meets the goals for MTD as described in Section 2.

3.1 Attack Model and Assumptions

In this paper, we assume an attack model where the attributes listed in a given ABAC policy, e.g., in a policy rule consisting of one or more constraints, become *compromised* by an attacker, thus creating an unintended attribute-access entity assignment, as depicted by the AA relation discussed in Section 2. Different ways an attacker may be able to compromise a given attribute may include, but may not be limited to the following: an unintended software error, forgery or a hacking incident compromising the infrastructure where attributes are created and assigned to entities, e.g., a remote credential server.

In addition, we also assume the following: first, the data containing attribute-based information can be properly collected and is available for analysis. In the context of EHRs, the application domain used for our running example, data collection may include a preprocessing step in which data

Table 1: A sample set of attribute-based access log data depicting EHRs.

EHR ID	Patient ID	Patient Loc	Personnel Loc	Role	Certification	Decision
A11234	A11234	Surgery	General	Surgeon	MD	FALSE
A43452	A43452	ER	ER	Nurse	RA	TRUE
A83422	A83422	General	General	Physician	MD	TRUE
A56102	A56112	Pediatrics	Lab	Lab Tech	CLIA	FALSE
A23108	A23108	ER	ER	Physician	MD	TRUE
A76313	A77777	General	General	Nurse	RN	FALSE
A89736	A89736	Radiology	Radiology	Nurse	RA	TRUE
A24912	A24912	Surgery	Surgery	Surgeon	MD	TRUE
A87632	A87632	Radiology	Radiology	Physician	MD	TRUE
A34028	A34028	General	Pharmacy	Physician	MD	TRUE

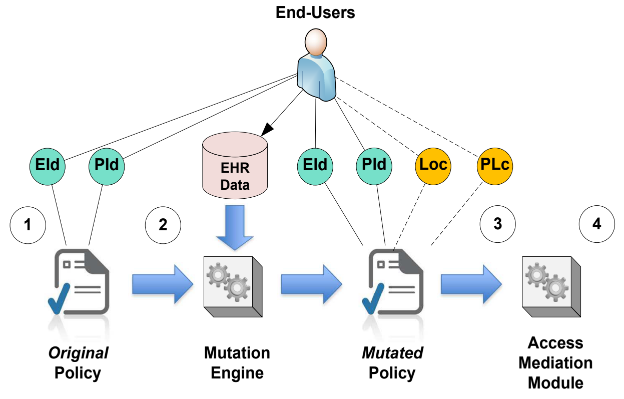

Figure 2: A graphical depiction of our approach: the original policy, defining a set of *original* attributes (1), is fed to the mutation engine along with data depicting an *attribute bag* (2). Such an engine identifies new attributes from users that are correlated to the original ones (3), producing a *mutated* policy that is later used for access mediation (4).

from the access logs is combined with information extracted from EHRs themselves. As an example, Table 1 shows data collected for each access request made in the context of an EHR. For each request, the following items are shown: first, the resource being accessed, the result of evaluating the request, and a description of the attributes (included values) shown at request time, a.k.a., the *attribute bag*. Second, we assume that the software framework handling the specification and runtime evaluation of ABAC policies, as well as the software modules implementing our approach (including the collection procedure described above), are out of reach for an attacker. As an example, even when a given ABAC policy, along with its listed attributes, may be known to the attacker, he/she has no way to deliberately change its contents, either by removing the policy as a whole or by adding or removing attribute-based rules at will.

3.2 Correlation-based Policy Mutation

A graphical depiction of our approach is shown in Fig. 2. Initially, we model an ABAC policy P as a set of constraint-based *rules* $\mathrm{R} = \{\mathrm{r}_1, r_2, r_3, ..., r_n\}$ for some $\mathrm{n} > 0$. For each rule $r \in \mathrm{R}$, we introduce the set S_r of attributes that are listed in it. In addition, we also model the *attribute bag* as described before as a set of attributes A such that $\mathrm{A} \cap \mathrm{S}_r \neq \emptyset$ for all $r \in \mathrm{R}$.

Given the original policy P, our approach then aims to produce a *mutated* policy P' as follows: for each rule $r \in \mathrm{R}$, we locate the set of attributes $\mathrm{C}_r = \{\mathrm{c}_1, \mathrm{c}_2, ... \ \mathrm{c}_p\} \subseteq \mathrm{A}$, $\mathrm{C}_r \neq \mathrm{S}_r$, that are *correlated* to the set of attributes S_r. Then, r may become a new rule r' by randomly choosing an attribute $c \in \mathrm{C}_r$ such that $\mathrm{S}_{r'} = \mathrm{S}_r \cup c$ [1]. Later, the set of modified rules $\mathrm{R}' = \{\mathrm{r}'_1, \mathrm{r}'_2, ... \ \mathrm{r}'_n\}$ is combined together to create the new mutated policy P'. As shown in Fig. 2, P' is forwarded to a policy evaluation module for further enforcement. We repeat the above procedure periodically in an effort to produce many different policy mutations. For such a purpose, an interactive approach may randomly produce modified rules as shown above by selecting only a subset of the set C_r of correlated attributes each time, in such a way that the resulting rules may vary from time to time. In addition, as time evolves, new correlated attributes may be collected in the attribute bag, thus possibly producing different mutated policies as a result.

We reiterate that finding the set C_r of correlated attributes is core to our approach. For such a purpose, we aim to find patterns relating the attributes in the attribute bag with the ones contained in the set S_r of original attributes. For illustrative purposes, assume a sample original policy based on the data shown in Table 1, which contains a single rule granting access to an EHR if the value of attribute *EHR ID* is equal to the value depicted by the *Patient ID* one.

Our attribute correlation process can be then described as follows: we start by first finding the relationship between the attributes in the original set S_r, e.g., *EHR ID* and *Patient ID*, and the access decision with a value of *true*, in an effort to identify within the data records depicting the attribute bag, the ones that belong to the requested access being granted according to our original policy. In Table 1, such relation is represented by the cells colored in *green*.

Next, we strive to find relationships between the original set S_r (*green*), as identified by the previous step, and some other attributes in the attribute bag. As an example, in Table 1, the values of attributes *Patient Loc* and *Personnel Loc* are the same when the values of the attributes *Patient ID*, *EHR ID* are equal as well, and the access decision depicts the value of *true*. Such a relationship is displayed in Table 1 in the *orange* color. In order for this step to be meaningful for our approach purposes, this relationship should be as strong as possible, that is, the vast majority of the records depicting the original attributes should also depict the newly-correlated ones. Referring back to Table 1,

[1] In case $\mathrm{C}_r = \emptyset$, then $r' = r$.

the number of cells colored in *green* and the ones colored in *orange* should be the same or stay within a close margin.

In a subsequent step, we also obtain the relationship between the original *green* attributes, the *true* access value, and the *inverse* of the attributes depicted in *orange* obtained from the previous step, e.g., the cells where the values for the *Patient Loc* and *Personnel Loc* are not the same. Such a relation is shown in the cells colored in *purple* in Table 1, and represents the entities having legitimate access according to the original policy but not holding the correlated attributes depicted in *orange*. Following the intuition described for the *orange* attributes, the number of cells colored in *purple* should be minimal with respect to the number of cells in *green* and *orange*, as a large number would imply a potential impact to the usability of our approach, e.g., entities getting previously-granted access denied as a consequence of implementing our solution.

Next, we strive to identify the relationship between the candidate *orange* attributes and the *false* access decision value, in an effort to make sure these newly-discovered correlated attributes are not shared by entities getting the *false* access decision in the attribute bag data. The intuition behind this is that the *orange* attributes should only be assigned to the entities getting legitimate access according to our original policy. Such a relationship is represented by cells depicting the yellow coloring in Table 1. Ideally, the number of cells in *yellow* should be minimal in respect to the number of cells depicting the *green* and *orange* colorings, e.g., close to zero, as a large number of such *yellow* cells would imply a potential security vulnerability.

With all this in mind, our approach should identify the candidate *orange* attributes in such a way that their relation to the original ones (*green*) is strong, whereas the relation with both the *yellow* and *purple* ones is kept to a minimum for safety and usability purposes, respectively. If such conditions are met, the *orange* attributes are said to depict the set C_r as described before, and can be then used to create mutations of the original policy. Following our running example, the newly mutated policy may include a new rule adding location of requirement of the values of *Patient Loc* and *Personnel Loc* to be equal along with the previous constraint relating the values of *EHR ID* and *Patient ID.*

3.3 Discussion

Following the discussion on MTD presented in Section 2, our approach strives to reduce the probability of carrying on a successful attack based on the model described in Section 3.1, by limiting the amount of time available for an attacker to exploit a compromised attribute. For such a purpose, we continuously mutate policies that leverage correlated attributes, such as the *orange* ones discussed above. This way, even when an attacker may be able to compromise an attribute in the original policy, the newly-correlated ones may be able to deter the attack. Moreover, our approach is also intended to avoid considerable impact to the *usability* of the system being protected. As mentioned before, end-users should not experience the rejection of previously-granted access requests as a result of the modifications made following the MTD paradigm. We achieve this goal by calculating the *purple* relation as describe above, and requiring it to be considerably less than the relation represented by the *orange* one. Not enforcing such requirement may deviate in mutated policies that may reject previously-granted requests, thus harming usability in a considerable way. Finally, even when an attacker may be aware of our proposed approach, we believe the continuous mutation of policies over time, as described in Fig. 2, as well as by randomly selecting a subset of *orange* attributes to appear on each mutation, may introduce a significant level of deterrence against possible attacks, e.g., predicting the next policy mutation. For such a purpose, we also assume the subset of *orange* attributes cannot by compromised by an attacker, at least until the next policy mutation. We base such assumption on the fact that in case an attacker can potentially modify any attribute at will at any time (including both the *green* and *orange* ones), not only our approach can be circumvented, but also the original attribute-based policy (and any other policies) that may be in place for access mediation purposes.

4. CONCLUSIONS

In this position paper, we have presented an on-going approach for leveraging MTD in the context of attribute-based policies. As of today, we are working towards refining the approach presented in Section 3. Concretely, we are formalizing our intuitions into a series of algorithms that leverage well-established techniques such as *association analysis* [5]. In addition, we have started the codification and evaluation process on custom-designed synthetic data based on previous examples found in the literature. We also plan to expand such a process by incorporating data obtained from a real-life EHR through a partnership with a healthcare organization. Finally, for the sake of efficiency, we plan to perform different performance measurements that include a variety of attack scenarios.

Acknowledgments

This work was partially supported by a grant from the National Science Foundation (NSF-SFS-1129561), by a grant from the Department of Energy (DE-SC0004308) and by a grant from the Center for Cybersecurity and Digital Forensics at Arizona State University.

5. REFERENCES

[1] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone. Guide to attribute based access control (abac) definition and considerations. *NIST Special Publication*, 800:162, 2014.

[2] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang. *Moving target defense: creating asymmetric uncertainty for cyber threats*, volume 54. Springer Science & Business Media, 2011.

[3] J. Jin, G.-J. Ahn, H. Hu, M. J. Covington, and X. Zhang. Patient-centric authorization framework for electronic healthcare services. *Computers & Security*, 30(2):116–127, 2011.

[4] C. E. Rubio-Medrano, Z. Zhao, A. Doupe, and G.-J. Ahn. Federated access management for collaborative network environments: Framework and case study. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies*, SACMAT '15, pages 125–134. ACM, 2015.

[5] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *ACM SIGMOD Record*, volume 25, pages 1–12. ACM, 1996.