Open Problems for Usable and Secure Open Systems

Adam J. Lee and Marianne Winslett
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, USA 61801
{adamlee, winslett}@cs.uiuc.edu

ABSTRACT

In open computing systems, resources are shared across organizational boundaries in an effort to allow for greater access to information and easier collaboration between geographically and administratively dispersed groups. Designing adequate access control solutions for these types of systems is a challenging task, as traditional solutions tend to exhibit failures or other undesirable behaviors in the face of an ever-growing and constantly evolving user base. To address these types of shortcomings, security researchers have proposed the notion of attribute-based access control (ABAC). Though the theoretical and systems issues associated with ABAC are currently being investigated, the human side of ABAC is relatively unexplored. In this paper, we discuss several usability and human factors challenges related to a promising ABAC solution known as trust negotiation.

ACM Classification Keywords

D.4.6. Operating Systems: Security and Protection—access controls, authentication; H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous; K.6.5. Management of Computing and Information Systems: Security and Protection.

Author Keywords

Attribute-based access control, open systems, trust negotiation, usability

INTRODUCTION

The notion of open systems, in which resources are shared between multiple organizations, has been gaining popularity for a number of years. Example open systems include scientific grid computing systems, ad-hoc networks used in disaster management scenarios, peer-to-peer networks for the exchange of multimedia data, supply chain management systems based on web services, and joint military task forces. Useful cyberinfrastructure for these types of systems must ensure that timely access to information and resources is granted to any and all individuals who need them, provided that these individuals are *authorized*. Traditional methods for determining user authorization, often based on user identities, fail to provide an adequate solution for use in open systems in which a dynamic and potentially unbounded set of users wish to interact with a large number of distributed services. Users have no basis for determining which services are trustworthy and services have no way to know the identities of each authorized user. To address this problem, the security community has proposed a number of *attribute-based* access control (ABAC) solutions.

ABAC systems (e.g., [2, 3, 5–7, 9, 10, 12, 16]) allow resource administrators to move away from maintaining complicated ACLs by allowing access policies to be written as declarative specifications of the attributes that must be possessed by authorized users. Many proposals for ABAC systems also give users the ability to control the disclosure and dissemination of their sensitive credentials. These systems have been shown to have solid theoretical underpinnings and to be feasible to deploy, however the human factors challenges associated with ABAC systems have been largely unexplored.

In this paper, we describe a popular ABAC solution known as trust negotiation. Using trust negotiation as a basis for discussion, we highlight a number of human factors challenges that arise in the context of ABAC systems. Many of these problems are multidisciplinary in nature and solving them correctly is likely to require cooperation between researchers in the fields of access control theory, software systems, networking, and computer-human interaction. Addressing these concerns prior to the deployment of ABAC systems is critical to ensuring the success of these systems and the open computing environments which they can enable.

TRUST NEGOTIATION

Having recognized the problems associated with performing access control in open systems, trust negotiation has been proposed as a potential solution [15]. In trust negotiation, the access policy for a resource is written as a declarative specification of the attributes that an authorized entity must possess to access the resource. In these systems, the credentials used to certify user attributes are also considered resources, so sensitive credentials can be protected by disclosure policies of their own. In this way, an access request leads to a bilateral and iterative disclosure of credentials and policies between the user and resource provider. Trust is established incrementally, as more and more sensitive credentials are disclosed between the user and resource provider.

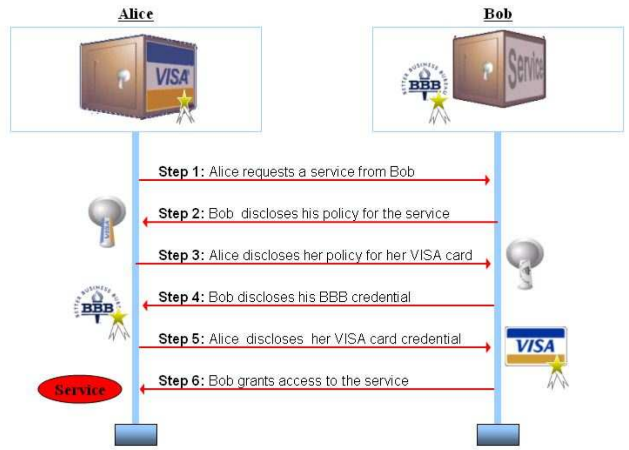
Alice
Bob
VISA
BBB
Service
Step 1: Alice requests a service from Bob
Step 2: Bob discloses his policy for the service
Step 3: Alice discloses her policy for her VISA card
Step 4: Bob discloses his BBB credential
Step 5: Alice discloses her VISA card credential
Step 6: Bob grants access to the service
VISA
Service

Figure 1. An example trust negotiation session

Figure 1 shows an example trust negotiation session between a user, Alice, and a resource provider, Bob. In the first step of this example, Alice requests access to some service provided by Bob. In step 2, Bob discloses the access policy for his service, which states that Alice must disclose her digital Visa cardholder credential, indicating that she has a Visa-issued credit card. In step 3, Alice discloses the policy protecting her Visa cardholder credential to Bob. This policy states that if Bob can prove that he is a member of the Better Business Bureau (BBB), Alice will disclose her Visa cardholder credential. Bob is in fact a member of the BBB and will disclose this credential to anyone. This satisfies Alice, who discloses her digital Visa cardholder credential in step 5 and is then granted access to the service in step 6. In this example, we see the bilateral and iterative nature of trust negotiation—not only does Alice disclose credentials to Bob, but also Bob discloses credentials to Alice. In this way trust is established incrementally, starting with the Alice's request, advancing through the disclosure of Bob's public BBB credential and Alice's sensitive Visa cardholder credential, and culminating with Bob allowing Alice access to his service.

Since trust negotiation is an automated process, negotiation strategies play an important part in determining the way that a particular negotiation proceeds. Prior to each round of disclosures, a negotiation strategy is used to determine which credentials and policies to disclose based on the policies and credentials received in the previous round. Strategies can be designed to optimize various aspects of the negotiation process. The *eager* strategy is used to optimize the speed with which negotiations take place by disclosing every credential whose disclosure policy is satisfied, regardless of its apparent relevance at the current stage of the negotiation. The *informed* strategy is optimal with respect to privacy preservation. This strategy discloses only those credentials whose disclosure policies are satisfied and are relevant at the current stage of the negotiation. A variety of other strategies have also been developed [18].

Trust negotiation addresses many of the shortcomings that traditional access control mechanisms exhibit when used in large-scale open systems. First and foremost, there is no closed-world assumption. Since authorization decisions are made based upon the attributes of potential accessors, there is no need for a predefined list of authorized users. Thus, access policies can remain more or less static, despite a changing user-base. Trust negotiation also allows users to discover resource access policies at runtime, which is important in environments where users and resources belong to different security domains. The bilateral and iterative nature of trust negotiation allows users to establish trust in resource providers as a part of the access request process. This protects clients from disclosing sensitive credentials to unknown resource providers. Additionally, many trust negotiation systems have desirable properties regarding the termination, soundness, and completeness of the interactions that take place between two participants.

OPEN PROBLEMS

It should be apparent from the preceding discussion of trust negotiation that the flexibility afforded by this and other ABAC systems comes at the cost of increased system complexity. At a recent workshop, Matt Bishop made the claim that while technology can certainly support security solutions, security itself is not a technological problem. He asserted that, in the end, *people* are both the problem and solution with respect to system security [4]. Viewed in this light, the adoption and acceptance of access control solutions such as trust negotiation, and ultimately the open systems that they enable, relies on the ability of humans to comprehend and manage their complexity. In this section, we discuss several important human factors challenges related to the comprehension of the access control process, the management of required technology, and the specification and maintenance of access control policies.

Access Control Comprehension

Traditional methods of access control are easy for users to understand. In the end, the decision made by the system depends simply on whether the requesting user's name appears in the access control list for a given resource. In trust negotiation, however, access control decisions are made without ever knowing the identity of a requesting user. Although automating the process of trust negotiation hides much of its complexity from the users in a system, it does so at the cost of making the process seem somewhat magical or arbitrary. After learning the results of a negotiation, the user requesting access will likely not know why her access was granted or denied. In current trust negotiation prototypes, technically savvy users can parse through log files and debugger output to determine the path taken by a negotiation, but this method of examination is far from accessible to an average user.

In [17], the authors present a system for visualizing the logs of previously-executed trust negotiation sessions carried out using the trust target graph (TTG) protocol [14]. While this is an excellent first step towards making the process of trust negotiation more accessible to everyday users, there is still considerable room for improvement. Although the tool presented can visualize the steps of a negotiation, these steps are labeled using the logical policy and credential expressions exchanged between the two parties. As an average user will not likely have a sufficient background in mathematical logic to fully understand these types of diagrams, it is imperative that the diagrams generated by visualization tools contain human-readable abstractions of the negotiation process, preferably with the level of abstraction tunable by the human requesting the visualization. In addition, rather than examining negotiation logs, the ability to visualize negotiations on the fly and interact with this largely automated process will be essential for users who wish to maintain greater control over their credentials. Designing and evaluating such an interface is likely to be a challenging task due to the wide range of expertise possessed by users of trust negotiation systems and the possibly complex interactions required with the logical engines used by trust negotiation agents.

In addition to abstracting access control policies from individual users, some trust negotiation policy languages allow access control policies to be context-dependent. The end result of this context dependence is that a particular user may be allowed access to a resource at one point in time and denied at another. Trust negotiation is already a complicated process to understand and these seemingly contradictory results could further confuse naive users. In [8], the authors explore how to provide context-sensitive feedback to users in the event that an access control decision is denied in a pervasive computing system; as with the trust negotiation visualizations previously discussed, this feedback is in a logical format. This work also requires that feedback policies be manually generated in conjunction with the access control policies themselves. It would be interesting to see work like this adapted to generate feedback policies automatically and allow variable levels of abstraction to make the feedback accessible to users of all levels of expertise.

Technology Management

In addition to understanding the operation of trust negotiation systems, users must also be prepared to manage the software, credentials, and policies required to operate these systems. History has shown time and again that people do a poor job of managing even simple passwords; most security administrators can provide anecdotal evidence of users sharing passwords, using extremely weak passwords, or even writing passwords down next to their terminal. In ABAC systems, users must manage cryptographic credentials (e.g., X.509 certificates) attesting to their various attributes. The management of these credentials is considerably more complex than the management of passwords which can lead to a variety of security and usability problems.

In [13], the authors show that most users had extreme difficulty using PGP 5.0 to manage a single certificate and securely encrypt and sign email messages. Users in attribute based access control systems must manage many such credentials, which adds further opportunities for security and usability problems to arise, particularly in the case in which a user wishes to access his credentials from multiple computers (e.g., an office computer, a laptop, and a home computer). The MyProxy [1] and Thor [11] projects seek to address the credential management problem for users in grid computing and trust negotiation systems, respectively. Both of these studies propose mechanisms for credential management that satisfy the security requirements for their particular domain, though they do not evaluate the ease of use of their solutions or analyze whether improper use of these mechanisms could lead to security violations. In the future, it will be important to study the requirements for secure identity management in ABAC systems and design credential management solutions which are both secure and easy to use.

In addition to the complexities of identity management, users must also manage and configure the other aspects of trust negotiation. In particular, trust negotiation sessions are strategy-driven interactions. It remains to be studied whether typical users of these systems will have the necessary skills and understanding to adequately choose trust negotiation strategies. Balancing the trade-offs between correctness, privacy-preservation, and negotiation speed is difficult and this difficulty will only increase as multiparty negotiation strategies begin to emerge. It is imperative that the capacity of users to make the strategic decisions necessary to effectively manage the configuration of trust negotiation software be evaluated so that appropriate interfaces for the management of these strategies can be designed.

Policy Specification and Maintenance

Trust negotiation systems require the specification and maintenance of complex policies by not only system administrators but also ordinary users. While administrators are often trained to specify and manage complex policies, end users typically have no such training; this leads to several prominent problems. One major concern is that users will have difficulties creating proper release policies for their sensitive credentials and personal information. A related concern relates to the ability of users (and trust negotiation strategies) to distinguish between authorization and need-to-know. For instance, though a bank may satisfy the release policy protecting a user's driving record, this information is not pertinent to the process of acquiring a home mortgage and thus should not be released to the bank. The widespread success of phishing attacks in recent years indicates that many users do not adequately protect their sensitive information and are thus likely to fall prey to policy specification and information protection pitfalls. Perhaps by studying user comprehension of threats to private information, we can adequately design systems in which it is difficult for users to make bad decisions regarding personal information protection.

In addition to problems surrounding information protection, policy specification is another area that requires attention. Current trust negotiation implementations require that users specify policies in either complicated XML or datalog-based formats. Writing policies in these formats is akin to writing computer programs in assembly language: the low-level of abstraction makes it difficult for an untrained user to map their mental model of the system onto the underlying logical policy specification model. Consider a user who wishes to allow firefighters access to a building protected by digital locks. A naive specification of the term "firefighter" might include "fire safety officers" employed at a local restaurant;

it is unlikely that the user meant for these restaurant workers to have unconditional access to their building! While this example is in some senses contrived, it does serve to illustrate the underlying problem. Investigating human comprehension of ontologies for use in constrained access-control environments could lead to the development of policy specification and analysis tools that would allow users without advanced knowledge of mathematical logic to specify realistic access control and release policies. These tools might also be used to allow users to quantify both the obvious and non-obvious effects of changes to these policies in a way that is consistent with the their own mental models of the system.

CONCLUSIONS

Creating and maintaining access control policies in open distributed systems is a complicated task. Although attribute-based access control systems such as trust negotiation have desirable theoretical properties and have demonstrated utility when used by security-conscious researchers with background in mathematical logic, the problem of providing an adequate open system access control solution for the average user is far from solved. In this paper, we indicated several broad areas of research in which the human aspects of trust negotiation and ABAC require advancement in order for these systems to be successfully deployed. Cooperation between researchers in the fields of access control theory, software systems, networking, and computer-human interaction could lead to important advancements in the areas of access control comprehension, identity management and system configuration, and policy specification and maintenance. These advancements are a critical part of enabling the successful deployment of flexible access control solutions such as trust negotiation and the ensuring the sustainability of the open systems that they enable.

Acknowledgments

The authors were supported by the NSF under grants IIS-0331707, CNS-0325951, and CNS-0524695. Lee was also supported by a Motorola Center for Communications graduate fellowship.

REFERENCES

1. J. Basney, M. Humphrey, and V. Welch. The MyProxy online credential repository. *Software: Practice and Experience*, 35(9):801–816, Jul. 2005.
2. M. Y. Becker and P. Sewell. Cassandra: Distributed access control policies with tunable expressiveness. In *Proceedings of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY '04)*, pages 159–168, 2004.
3. E. Bertino, E. Ferrari, and A. C. Squicciarini. Trust-X: A peer-to-peer framework for trust establishment. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):827–842, Jul. 2004.
4. M. Bishop. Panel Discussion, ITI Workshop on Dependability and Security, Dec. 2005.
5. M. Blaze, J. Feigenbaum, and A. D. Keromytis. KeyNote: Trust management for public-key infrastructures (position paper). *Lecture Notes in Computer Science*, 1550:59–63, 1999.
6. M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *IEEE Conference on Security and Privacy*, May 1996.
7. P. Bonatti and P. Samarati. Regulating service access and information release on the web. In *7th ACM Conference on Computer and Communications Security*, pages 134–143, 2000.
8. A. Kapadia, G. Sampemane, and R. H. Campbell. KNOW why your access was denied: Regulating feedback for usable security. In *11th ACM Conference on Computer and Communications Security (CCS04)*, pages 52–61, Oct. 2004.
9. H. Koshutanski and F. Massacci. An interactive trust management and negotiation scheme. In *2nd International Workshop on Formal Aspects in Security and Trust (FAST)*, pages 139–152, Aug. 2004.
10. N. Li and J. Mitchell. RT: A role-based trust-management framework. In *Third DARPA Information Survivability Conference and Exposition*, Apr. 2003.
11. T. W. van der Horst and K. E. Seamons. Short paper: Thor—the hybrid online repository. In *First IEEE International Conference on Security and Privacy for Emerging Areas in Communications Networks*, Sept. 2005.
12. L. Wang, D. Wijesekera, and S. Jajodia. A logic-based framework for attribute based access control. In *2nd ACM Workshop on Formal Methods in Security Engineering (FMSE 2004)*, pages 45–55, Oct. 2004.
13. A. Whitten and J. D. Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *8th USENIX Security Symposium*, pages 169–183, Aug. 1999.
14. W. H. Winsborough and N. Li. Towards practical automated trust negotiation. In *Third IEEE International Workshop on Policies for Distributed Systems and Networks*, Jun. 2002.
15. W. H. Winsborough, K. E. Seamons, and V. E. Jones. Automated trust negotiation. In *DARPA Information Survivability Conference and Exposition*, Jan. 2000.
16. M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu. The TrustBuilder architecture for trust negotiation. *IEEE Internet Computing*, 6(6):30–37, Nov./Dec. 2002.
17. D. Yao, M. Shin, R. Tamassia, and W. Winsborough. Visualization of automated trust negotiation. In *Workshop on Visualization for Computer Security (VizSEC'05)*, Oct. 2005.
18. T. Yu, M. Winslett, and K. E. Seamons. Interoperable strategies in automated trust negotiation. In *ACM Conference on Computer and Communication Security*, Nov. 2001.