Position Paper—Medina: Combining Evidence to Build Trust

Johannes Helander and Benjamin Zorn
Microsoft Research
{jvh,zorn}@microsoft.com

April 2007

Abstract

Security mechanisms require flexibility to accommodate the frailties of the imperfect people that use them. For example, password systems typically allow users who forget their passwords to reset their password after passing some other test. More generally, many human decisions of trust are based on weighing a preponderance of evidence in an ad hoc fashion. We present Medina, an authentication system based on combining various forms of evidence in a computational framework. Medina assumes that all authorization decisions are based on weighing a variety of evidence and brings elements of security (such as what happens when someone forgets their password) into a computational framework. Medina also allows for a range of access control policies that are less strict and/or more flexible than traditional security mechanisms.

1 Introduction

Ideally, security mechanisms are simple and we can reason directly about their effectiveness using mathematical tools. For example, we can directly reason about the security of password protection by analyzing the number of possible combinations that an attacker would have to attempt in order to guess the password. Passwords are almost never used in isolation, however, due to the possibility that a user might forget one. Systems that allow a forgotten password to be emailed to a specified mail account are widely used on the Web. While these external mechanisms provide greater convenience, they also significantly reduce our ability to reason about the security the overall system provides (e.g. see Adams et al. [2]). When password security is viewed at this higher level, we observe that authentication requires a combination of different forms of evidence: either you know the password or you have access to the correct email account. We observe that many security systems involve such layering and use ad hoc mechanisms for convenience.

Medina is a framework in which the ad hoc solution described above can be systematically generalized, reasoned about, and extended. We observe that many non-computation forms of security involve gathering a body of evidence and weighing the evidence against the access privilege being granted. We seek to bring such a system into the computational domain by systematizing the definition of evidence, the system for combining evidence, and the policy with which access privileges are granted based on evidence.

2 Passwords in Medina

In Medina, access control is granted based on evidence, which represents a set of facts ($E = \{e_i\}$), and an access control policy, P, which maps evidence to a boolean access control decision $allow = P(E)$. For example, a password system that allows you to have your password emailed to a specific email address could be expressed in Medina as follows. First, there are three facts: e_1 = "knows password", e_2 = "has an email address registered with the account", and e_3 = "can read email sent to that address". Thus, $allow = P_1(e_1, e_2, e_3) = e_1|(e_2 \& e_3)$.

With this formulation, we see immediately that P is strictly weaker than than e_1 alone (which we can reason about), and depends on the security policy associated with e_3, which might also involve having a password but which may have different policies for protecting it (e.g., whether it is ever sent over a network in clear text, how strong the password has to be, etc.). Taking the example a bit further, e_2 implicitly implies another fact, e_4 = "entity requesting access is human", due to the assumption that authorities granting mailboxes attempt through some means (such as a human interaction proofs HIP [6], perhaps in the form of a CAPTCHA [16, 17]) to verify that the mailboxes are being assigned to human beings.

We can improve the security of a password pro-

tection system by changing the policy as follows: $P_2(e_1, \ldots, e_4) = e_4 \& P_1(e_1, e_2, e_3)$. This new policy requires the entity requesting access to not only know the password, but explicitly verify that they are human (e.g., through a use of a HIP). From this example, we have shown that expressing parts of a security system that are typically not expressed (e.g., things like requesting a user to know a fact, such as her mother's maiden name before granting access), and sometimes only implicitly present (such as proof of humanity), allow clearer reasoning about the effectiveness of the system.

3 Implications of Medina

By explicitly incorporating a variety of evidence in making access control decisions, Medina allows different previously defined forms of evidence to be incorporated into a unified framework. As we have seen, combining HIPs with passwords results in strictly stronger security than passwords alone. Likewise, one can imagine using Medina to achieve the opposite effect. For example, if a more convenient, but weaker form of access control is desired (for example, for sharing photos, a scenario we describe below), weaker evidence, such as proof of humanity alone, might be appropriate. Many forms of evidence have already been proposed in the literature, including HIPs, client puzzles [7, 10, 18], biometrics [14], proximity [12], peer rating [15], and simple tests of knowledge. We envision incorporating all of these forms into Medina evidence, and creating new forms of evidence based on additional usage scenarios.

We have introduced Medina using a familiar traditional access control problem, password-based authentication, and showed that it is flexible enough to express this problem in its more general structure. We believe the true strength of Medina is in expressing access control based on a variety non-traditional evidence as compared to traditional password-based access control. Many informal human interactions take place without employing heavyweight authentication procedures. The inconvenience of remembering and entering passwords is known to be a burden that is typically reserved for enforcing only important access control. However, as computers become a fundamental part of our everyday interactions with other humans, our access control mechanisms must also evolve to more closely mirror known social mechanisms for interpersonal trust. We also seek to implement a computational basis for such trust mechanisms that can integrate with the trust relationships of the human users.

We have developed several scenarios, one of which we discuss below, that we believe represent challenges to existing security frameworks. These scenarios do not easily fit into a traditional onion-model of access control (e.g., as in the military model of computer security described by the Orange Book [8]), but are more amenable to being expressed in terms of Medina. While space prevents us from fully outlining how Medina can be used for these scenarios, we feel that they represent challenges that future security systems to address and therefore are an appropriate contribution to a workshop.

Scenario: Soccer Pictures at Starbucks
Consider two acquaintances sitting at a cafe next to each other, each with a laptop. If one wants to give the other a picture of her kid's soccer team, there are few good options. She could put the picture on a memory stick but subject both machines to transmission of viruses. She could email or instant message the pictures but those systems are not suited to large file transmission, require disclosure of the email address, and is unnecessarily clumsy considering both people and laptops are next to each other. She could put the page on a web server somewhere, but would have to either make the picture available to everybody or setup an account with a password.

Direct file access over wireless is even harder due to the complexity of setting up accounts across domains, usually the prerogative of the domain administrator, especially on a work laptop. What the memory stick does is to prove physical proximity, humanity, and human-to-human trust by the hand-to-hand exchange. The ideal solution would be to use the built-in wireless adapters to do essentially what the memory stick does but without the physical medium and risk of directly accessing the file system. It seems intuitive to most users that that this should "just work".

A systematic solution to this problem based on Medina could express the properties of a virtual USB flash drive. When you share this device with another person, for example, by telling them its name, their computer would have access to all the files you dragged to it. In this scenario, the evidence used to determine access would be proximity (based on the local wireless transfer protocol), humanity (based on a HIP built into the sharing interface), and interpersonal trust, based on the fact that they know the word you used). A touch based key exchange could also be used for additional strength [12].

Scenario: Wiki
Consider a multi-user multi-editor collaborative online project, such as a Wiki. It might be desirable

to let the best experts of a field be in charge of editing a particular page. The expertise can be measured through a quiz in that field of knowledge and through reader and peer ratings. The editor can then gain access to the cached knowledge score by using a password. Different pages will require different quizzes, for instance if one page is about 16 bit assembly programming and the other one about horse grooming, there would be little in common.

The following example illustrates an access policy, which specifies whether a user can read or edit different parts of the wiki, where `peer` signifies a user's peer rating, and `quiz1` and `quiz2` represent their score on knowledge quizzes:

```
edit1 = ((quiz1>70% & peer>50%) | passwdA) & HIP
edit2 = ((quiz2>90% & peer>75%) | passwdB) & HIP

read1 = anybody
read2 = (peer>20%) & HIP
```

In this example, to edit page 1 on assembly programming the editor would have needed to score reasonably well on the programming quiz and have a favorable peer rating. However, anybody including web crawlers can read the page.

The standards for horse grooming are stricter, however. The quiz2 score needs to be almost perfect and peer acceptance higher. To even read the page at least a modest peer approval, such as a recommendation by two friends, is required, in addition to proof of humanity.

4 Adaptive Trust Evaluation

Access control decisions can be computed by combining boolean evidence using functions and propositional logic as seen above. This model can be extended in multiple ways. We are particularly interested in computing trust based on diverse and complex forms of evidence. In this section, we consider novel opportunities in this area.

Time-based
Trust changes over time. For instance login sessions or passwords expire after a while. To keep working the user may be required to enter additional evidence. For instance some Web email services require new HIPs after the user has sent a large number of emails before even more emails can be sent. Password systems often refuse to allow continued attempts after some number of failed attempts. Expressing policy as a process rather than a static function may be one way to incorporate time and temporal change to the system.

Probabilistic
Often evidence is not clearly right or wrong. A user may have answered seven out of ten questions correctly. Setting a fixed threshold for what constitutes valid evidence (e.g., getting 7 or more out of ten questions right) may result in a rigid mechanism which fails to have the desired flexibility. It would be more natural to conclude that this constitutes e.g. 70% evidence. Rather than applying a confidence test at each evidence separately to turn the result into a boolean and then combining the booleans using discreet logic it may be advantageous to first combine the evidence and then apply the confidence test to that. After all the final result must be boolean (allow/deny) but the intermediate calculation can be determined, for example, using Bayesian reasoning.

Stochastic process
Combining a process that changes over time with a probabilistic evaluation function that incorporates a feedback loop constitutes a stochastic process. There are many formal ways to reason about such processes, especially if one considers specific subsets, such as Markov processes. An application of simple stochastic processes to distributed real-time scheduling is presented in [11] and shows that simple stochastic processes can successfully be used on embedded microcontrollers. A probabilistic evaluation can easily be adapted over time, including gradual decay functions and introduction of additional evidence.

Gray areas between trust and mistrust
When a trust evaluation results in a confidence that is very close to the threshold between trust and mistrust we introduce a "gray area". In the gray area the system continues to allow operations to go forward but does so with suspicion. The system will then pay extra close attention to user activities and if suspicious activity is detected the trust will be lowered below the threshold, to deny.

Negative evidence
Evidence can exist in both positive and negative forms, and the quality of evidence can be re-evaluated. One approach to forming negative evidence is taken by anomaly detection systems (e.g., [13]), which build negative evidence against a principal based on behavior. In these systems, normal behavior is modelled and observed activities are compared to the model. If the two deviate the result is negative evidence. As with positive evidence, the strength of negative evidence might decay when either the model adapts to the reality or the suspicious activity dissipates.

5 Related Work

The issue of trust is a broad one, typically integrating social considerations (such as what organization you belong to and who your friends are) with access control mechanisms and policies [3]. Trust management systems [4] attack the complex problem of providing security in network services by formalizing the definition of credentials and defining policies that determine whether a particular set of credentials is sufficient to allow access. Two challenging aspects of trust management research are developing approaches that work in a distributed, multi-organizational setting and defining formal policy languages that allow policies to be specified clearly and concisely.

Chapin et al. [5] describe the state of the art in trust management research, and conclude that formally well-founded approaches offer the greatest promise for meeting the design goals of such systems. They distinguish trust management systems from simpler access control systems by assuming that in simple access control systems all principals are known a priori, and access decisions are based on the identity of the principals. In trust management systems, by the nature of their being distributed, it is assumed that there is no central authority that knows all principals. Our work differs from existing work on trust management, in that we are initially not focusing on the broader problems of a distributed environment. Instead, our interest is in developing a richer definition of the kinds of information used to make access control decisions and building flexible policies around that rich information.

In addition to work on trust management systems, researchers have investigated the use of logic in specifying access control policies. Abadi presents a survey of the logics and languages used for this purpose [1]. Binder is an example of one such security language, which encodes security policies in datalog, a subset of the Prolog language [9]. Binder allows complex policies, that might otherwise be represented as a large access control matrix, to be specified concisely and precisely as a logic formula. Our work differs from previous work in this area in that we are more interested in understanding the domain of data that can be used in Binder-like formalisms. We also see weaknesses in building a security language based on logic in the presence of authorization data that is imprecise.

6 Summary

We have outlined Medina, an authentication framework based on combining evidence from multiple sources. Medina allows existing ad hoc security mechanisms (such as asking personal information) to be incorporated into the description of a security policy and reasoned about more systematically. Medina is especially useful when different, possibly non-traditional forms of evidence, such as HIPs, proximity, biometrics, client puzzles, knowledge tests, etc. are combined to reach an access control decision. Our vision is that Medina can combine evidence discreetly or probabilistically and adapt to newly introduced evidence as well as modify decisions based on existing evidence as a function of time. Medina is designed for expressing and computing access control decisions based on weaker forms of trust (e.g., trust between social acquaintances, such as the soccer parents), that balance permissive access with convenience. We envision that as computers become an essential part of our lives, policies related to trust and access control must adapt to more closely support social practices.

References

[1] M. Abadi, "Logic in access control," in *LICS*. IEEE Computer Society, 2003, p. 228. [Online]. Available: http://csdl.computer.org/comp/proceedings/lics/2003/1884/00/18840228abs.htm

[2] A. Adams and M. A. Sasse, "Users are not the enemy," *Commun. ACM*, vol. 42, no. 12, pp. 40–46, 1999.

[3] T. Berners-Lee, "Trust," Jan. 2006. [Online]. Available: http://www.w3.org/2000/10/swap/doc/Trust

[4] Blaze, Feigenbaum, and Lacy, "Decentralized trust management," in *RSP: 17th IEEE Computer Society Symposium on Research in Security and Privacy*, 1996.

[5] P. Chapin, C. Skalka, and X. S. Wang, "Trust management: Features and foundations," 2006, submitted for publication.

[6] M. Chew and H. Baird, "Baffletext: A human interactive proof," in *Proceedings of SPIE-IS&T Electronic Imaging, Document Recognition and Retrieval X*, Jan. 2003, pp. 305–316. [Online]. Available: citeseer.ist.psu.edu/chew03baffletext.html

[7] D. Dean and A. Stubblefield, "Using client puzzles to protect TLS," in *Proceedings of the 10th USENIX Security Symposium*. Washington, D.C.: USENIX, Aug. 2001. [Online]. Available: http://www.cs.rice.edu/ astubble/tls-usenix.pdf

[8] *Department of Defense Trusted Computer System Evaluation Criteria*, Department Of Defense Computer Security Center, Aug. 1983.

[9] J. DeTreville, "Binder, a logic-based security language," in *IEEE Symposium on Security and Privacy*, 2002, pp. 105–113. [Online]. Available: http://computer.org/proceedings/sp/1543/15430105abs.htm

[10] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1993, pp. 139–147.

[11] J. Helander and S. B. Sigurdsson, "Self-tuning planned actions: Time to make real-time SOAP real," in *ISORC*. IEEE Computer Society, 2005, pp. 80–89. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ISORC.2005.51

[12] J. Helander and Y. Xiong, "Secure web services for low-cost devices," *isorc*, vol. 00, pp. 130–139, 2005.

[13] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 3, pp. 295–331, 1999.

[14] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001. [Online]. Available: http://www.research.ibm.com/journal/sj/403/ratha.html, http://www.research.ibm.com/journal/sj/403/ratha.html

[15] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Commun. ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[16] von Ahn, Blum, Hopper, and Langford, "CAPTCHA: Using hard AI problems for security," in *EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT*, 2003.

[17] L. von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Commun. ACM*, vol. 47, no. 2, pp. 56–60, 2004.

[18] B. Waters, A. Juels, J. A. Halderman, and E. W. Felten, "New client puzzle outsourcing techniques for dos resistance," in *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2004, pp. 246–256.