Interaction Provenance Model for Unified Authentication Factors in Service Oriented Computing

Ragib Hasan and Rasib Khan
SECRETLab, Department of Computer and Information Sciences
University of Alabama at Birmingham, AL, USA
{ragib, rasib}@cis.uab.edu

ABSTRACT

Authentication is one of the most fundamental security problems. To date, various distinct authentication factors such as passwords, tokens, certificates, and biometrics have been designed for authentication. In this paper, we propose using the history or provenance of previous interactions and events as the generic platform for all authentication challenges. In this paradigm, provenance of past interactions with the authenticating principle or a third party is used to authenticate a user. We show that the interaction provenance paradigm is generic and can be used to represent existing authentication factors, yet allow the use of newer methods. We also discuss how authentication based on interactions can allow very flexible but complex authentication and access control policies that are not easily possible with current authentication models.

Categories and Subject Descriptors

K.6.5 [**MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS**]: Security and Protection (D.4.6, K.4.2): Authentication

Keywords

Authentication, Events, Interaction, Provenance, Security

1. INTRODUCTION

In our everyday lives, we frequently face the need to prove our identities to others. A valid claim of identity allows a service provider to link the claimed user to the available services. In service oriented computing, authentication refers to proving one's identity to a challenging authority, and subsequently, avail the offered services from the provider. Authentication is the most critical part in ensuring security in any service oriented architecture. Service providers incorporate different authentication mechanisms according to their need and purpose. Usually, authentication services are dependent on three prime factors, that is, what the user knows, what the user has, and what the user is. However, most authentication mechanisms still remain proprietary and pose as a challenge in ensuring a completely secure process.

We have identified an inherent similarity among all the authentication factors, in their singular form or in any combination of factors. A system can authenticate a user only based on past events where the user interacted with the system. For example, a password can be used for authentication to a system only if the user has registered an account in the system and created (or was assigned) a password before. This is similar to authentication in social contexts. People recognize each other based on their previous interactions, events, and actions. Additionally, this information is not presented only at the time of recognition, but rather exists as a string of events over the particular subject's timeline.

Based on this observation, we believe that these apparently disjoint authentication factors can be fundamentally merged under a common root. In this paper, we use this notion to propose a model to unify all existing authentication factors into a single interaction provenance verification scheme. Our proposed model delivers a concept of using past interactions between various entities to validate the entities involved in authentication. We refer to the term interaction provenance to represent the set of events in a user's history of interactions with various systems. We claim that all authentication factors can be represented in terms of interaction provenance.

2. MOTIVATION

In everyday practices, the more secure a system is, the less usable it becomes for its users. For authentication, three major factors have been developed and deployed over the years [2]. Authentication challenges are considered to be fundamentally designed around one or more of the following factors:

Knowledge: In this case, authentication is done by the subject by presenting a secret shared between the subject and the system at an earlier time. This can be a password, which the subject has established when setting up her account with the system. This can also be a shared knowledge about the user (e.g., the amount of a few transactions posted to a user's bank account in the last week). This factor is perhaps the most commonly used factor in authentication, yet the most attacked. Passwords are vulnerable to simple guessing attacks using a dictionary-based or a brute force approach. Strong passwords that are hard to break are also hard to remember, making them difficult to enforce in practice.

Possession: Here, authentication is done by presenting a physical or digital object that the subject holds. For example, it can be a badge or a token or a X.509 digital certificate held by the subject. There are also hardware devices which a user must use to generate one-time-pads and present to the provider. The system can verify the token and therefore the identity of the subject. In each of these cases, the authentication in dependent on a certain possession of the user at the time of validation, and has been registered as a valid item.

CODASPY'14, March 3–5, 2014, San Antonio, Texas, USA.
ACM 978-1-4503-2278-2/14/03.
http://dx.doi.org/10.1145/2557547.2557578 .

Biometrics: The physical characteristics of the subject are also used for authentication. The most common and widely used biometric authentication schemes include fingerprints, voice recognition, iris recognition, and face recognition. Bodily attributes can be considered as the most unique authentication features. However, in case biometric information, such as fingerprints, is forged, the subject loses the ability to use fingerprints ever again for authentication purposes.

Contextual Information: There can be other authentication factors used in the validation process. In general, they are referred to as contextual information, and include the location of the subject, background or network oriented data, and recommendation chains from other users [3]. However, this information is not secure and self-sufficient, and therefore, acts as a reinforcement factor for other authentication mechanisms.

Therefore, we can see that each of the factors is prone to multiple vulnerabilities when considered on their own. Multi-factor authentication mechanisms have been incorporated into systems which require higher security. However, the usability of such multi-factor authentication mechanisms are greatly reduced, as users are required to memorize passwords, save certificates, and carry around one-time-pad generating devices. Another problem with all existing authentication mechanisms is that they rely on credentials presented at only the given time. As a result, any lost credential (username/password, certificate) results in authentication fraud and identity theft. Identity thefts as such are very easy, since an attacker only needs to provide the information (knowledge, possession, or biometrics) only at the time of authentication.

Access control and authorization of resources are also complicated in terms of such authentication credentials. The main security problem is to determine the rights (e.g., read, read-write, etc.) a user has over a given resource or object [2]. However, common access control mechanisms are unable to impose such policies based on authentication credentials. Furthermore, cross-platform compatibility of authentication and authorization are always a critical problem among service providers, when it comes to agreeing on a common protocol for supporting user transition from one provider to the other.

3. UNIFICATION OF AUTHENTICATION FACTORS

In this section, we present the concept of interaction provenance, and its applicability for authentication and other security domains.

3.1 Interaction Provenance

An interaction is an event or a record of a user action with one or more other entities. Therefore, an interaction entry is a log of the protocol execution, and is maintained in an ordered set of messages or actions performed by two or more entities. A principal is a participant in an interaction provenance entry, if it had sent or received at least one message, or had initiated at least one action. We define an event as a particular action or a record of a protocol execution that has taken place in the past. Interaction provenance of a principal is a chronologically ordered sequence of interaction entries, in each of which, the principal was a participant for a particular event.

From the definition of interaction, it immediately follows that interactions are always about events. Since time is linear, interactions for a user form a chronologically ordered chain, with no cycles. Interactions are strongly attached to the user and cannot be transferred. Interactions are also considered mutual. That is, both parties in an interaction will always have a record of the same event. For example, when Alice registers an account with a service provider, she interacts with it by following its new account registration protocol, accessing the system, and then setting a user name and a password. The provider also sees Alice's actions and is a participant in the registration protocol. Later, whenever Alice logs into the service, she runs different protocols and has various interactions with the system and/or other external or internal users.

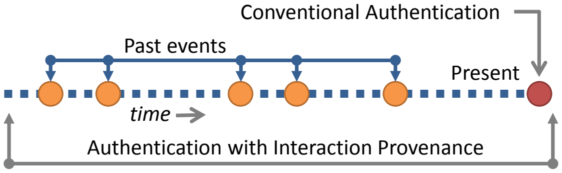
Conventional Authentication
Past events
Present
time
Authentication with Interaction Provenance

Figure 1: Interaction Provenance for Authentication

Additionally, ordering of the interaction entries are chained, such that, the chronological sequence cannot be altered with respect to each other. To use interaction provenance for authenticating users, it is essential that the data can be verified and any tampering should be detected. We can adapt techniques from secure data provenance for designing tamper-proof individual interactions and the order of interaction provenance chains [8, 9].

3.2 Interaction Provenance for Authentication

The various conventional authentication factors depend on some previous events or interactions that the user had with the server, or some other entity trusted by the server. Authentication is dependent on a past event when the user created or registered the secure credentials with the provider. From this perspective, we can say that all the authentication factors can simply be represented as the use of past interactions. Based on this observation, we propose using interaction provenance as the only generic authentication factor, and thus unifying all the conventional authentication factors. Users can prove k out of n recorded past events and corresponding interactions, and authenticate themselves to a system. The provider verifies that the claimed interaction indeed occurred and satisfies the authentication policy of the system. Upon successful validation, the system can verify the identity of the principal subject in the interaction, and map that to a known user in the system. Therefore, as shown in Figure 1, authentication using interaction provenance enforces a validity check on the timeline of past events. On the contrary, conventional authentication procedures only rely on presenting authentication credentials at the present time.

We used certain models of data block composition to represent events and interactions, and also for user-provider interaction at later times, as shown below:

Interaction [UserID, ProviderID, EventType [Key|Value, ..]]

To illustrate how current authentication factors can be represented in the form of events and interactions, we can consider the individual authentication types. In the case of password or shared-secret based authentication systems, we can model a 'registration' event as a past interaction. Therefore, this event is presented by the user during authentication at a later time as follows:

Sender: *UserA*, **Receiver**: *ProviderB*
Interaction: [*UserA*, *ProviderB*, Registration [UserID: *UserA*,
ProviderID: *ProviderB*, Password: *password*,
RegTime: *timestamp*]]

Authentication systems based on certificate or token possession can also be modeled as above. According to our scheme of interaction provenance, we can consider the issuance of the token by a trusted certification authority as an event. Therefore, the user can present this interaction to any other provider for the purpose of authentication as follows:

Sender: *UserA*, **Receiver**: *ProviderB*
Interaction: [*UserA*, *TrustedPartyID*, Issue_Credential [
UserID: *UserA*, ProviderID: *TrustedPartyID*,
Credential: *token*, CreateTime: *timestamp*]]

In the same way shown for the above cases, biometric authentication can also be represented using interaction provenance. As we know that the user had a past event when the biometric information was registered with the provider, this interaction can be presented during authentication as follows:

Sender: *UserA*, **Receiver**: *ProviderB*
Interaction: [*UserA*, *ProviderB*, Register [UserID: *UserA*,
ProviderID: *ProviderB*, Biometric: *data*,
CreateTime: *timestamp*]]

Additional factors, such as, contextual information and recommendation chains can also be presented using interaction provenance [3]. The representation of information may vary based on the type of factor in use. All authentication factors represented as past events can thus be unified under a single paradigm of authentication using interaction provenance items.

3.3 Extension of Interaction Provenance in Secure Systems

Interaction provenance can be extended to serve other security services. We can augment access control and authorization to make it more flexible and dynamic via the use of interaction provenance. Access policies can be written in terms of past interactions and events. Therefore, a user will be allowed access to a resource if a specific type of interaction provenance can be presented. In an example scenario, an airline traveler will require to present interaction records with the ticketing system, which implies that, a successful purchase of tickets. Next, the passenger is required to present successful verification of documents and passing through the airport security checkpoints. Another example can be to share contents on social network, only with people who had previous interactions with the user. Interaction provenance can be used to implement path-based access control, where access to a resource depends on the physical (or logical) path of the user or data item. Past interactions with secure systems and other users can be used to leverage assertion and contextual information based admission to resources.

4. MODEL ANALYSIS

Unifying different authentication factors into a single authentication paradigm based on provenance is beneficial in many respects. First, the proposed model unifies all existing authentication factors into a common representation model. This unification can be leveraged to implement cross-platform and common authentication mechanism among service providers. Second, interaction provenance can be used to enforce authentication mechanisms based on a string of past events, in contrary to only presenting credentials at the present moment. This allows an improved level of security for the domain of service oriented computing. Third, the proposed model allows newer methods or factors of authentication, such as, knowledge belonging to group interactions. This also introduces increased flexibility in authentication and brings the authentication process closer to real life trust establishment. Fourth, interaction provenance can allow mutual authentication of users and providers, where both of which should present a previous record of interaction. Fifth, we can utilize interactions to allow anonymous authentications by creating an authentication event the first time, and validating the provenance in the subsequent occasions. Finally, extending the idea of interaction provenance to other security problems introduces significant benefits. Authorization and access control can be defined using richer and simpler semantics, which will allow writing complex and innovative security policies based on past interactions.

5. RELATED WORK

A lot of research in recent years has focused on securing provenance information against illicit tampering and confidentiality or information flow violations [2, 4–6]. However, our proposed primitive looks at the opposite problem: how provenance can be used to solve security problems such as authentication and access control. New innovative methods of authentication such as using recommendations from other validated users have been proposed recently [3]. However, to the best of our knowledge, no attempts have been made to unify all authentication factors. In this work, we propose using interaction provenance as the only generalized authentication factor. The use of the history of a user or an application for access control has been explored by some researchers. Edjlali et al. discussed the use of the history of mobile code to determine access control [7]. Abadi et al. presented an access control model based on application execution history [1]. Krukow et al. extended the idea to provide a logical framework for history based access control [10]. In our paper, we propose making history or provenance of interactions as the only factor in access control.

6. CONCLUSION

This paper introduces interaction provenance as a generalized factor for authentication. We showed that existing authentication factors can be represented via interactions, and new authentication methods can be introduced through the use of interactions. We posit that the notion of interaction provenance as a fundamental security primitive can be successfully used in many areas of security and has the potential of bringing flexibility and introducing novel applications that are not currently possible with existing approaches.

Acknowledgment

This research was supported by a Google Faculty Research Award, the Office of Naval Research Grant #N000141210217, the DHS Grant #FA8750-12-2- 0254, and under NSF Grant #0937060 to the Computing Research Association for the CIFellows Project.

References

[1] M. Abadi and C. Fournet. Access control based on execution history. In *Proceedings of the 10th Annual Network and Distributed System Security Symposium*, pages 107–121, 2003.

[2] M. A. Bishop. *The Art and Science of Computer Security*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[3] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung. Fourth-factor authentication: somebody you know. In *Proceedings of the 13th ACM conference on Computer and Communications Security*, CCS '06, pages 168–178, New York, NY, USA, 2006. ACM.

[4] U. Braun, A. Shinnar, and M. Seltzer. Securing provenance. In *Proceedings of The 3rd USENIX Workshop on Hot Topics in Security (HotSec)*, July 2008.

[5] J. Cheney. A formal framework for provenance security. In *Computer Security Foundations Symposium (CSF), 2011 IEEE 24th*, pages 281–293, 2011.

[6] S. Chong. Towards semantics for provenance security. In *First workshop on on Theory and practice of provenance*, TAPP'09, pages 2:1–2:5, Berkeley, CA, USA, 2009. USENIX Association.

[7] G. Edjlali, A. Acharya, and V. Chaudhary. History-based access control for mobile code. In *Proceedings of The 5th ACM Conference on Computer and Communications Security*, CCS '98, pages 38–48, New York, NY, USA, 1998. ACM.

[8] R. Hasan, R. Sion, and M. Winslett. Introducing secure provenance: problems and challenges. In *Proceedings of The ACM Workshop on Storage security and survivability (StorageSS)*, pages 13–18, New York, NY, USA, 2007. ACM.

[9] R. Hasan, R. Sion, and M. Winslett. Preventing history forgery with secure provenance. *ACM Transactions on Storage (TOS)*, 5(4):12:1–12:43, Dec. 2009.

[10] K. Krukow, M. Nielsen, and V. Sassone. A logical framework for history-based access control and reputation systems. *Journal of Computer Security*, 16(1):63–101, Jan. 2008.