Simple Authentication for the Web [*]

Timothy W. van der Horst
Internet Security Research Lab
Brigham Young University
Provo, UT, USA
timv@cs.byu.edu

Kent E. Seamons
Internet Security Research Lab
Brigham Young University
Provo, UT, USA
seamons@cs.byu.edu

ABSTRACT

Automated email-based password reestablishment (EBPR) is an efficient, cost-effective means to deal with forgotten passwords. In this technique, email providers authenticate users on behalf of web sites. This method works because web sites *trust* email providers to deliver messages to their intended recipients. Simple Authentication for the Web (SAW) improves upon this basic approach to user authentication to create an alternative to password-based logins. SAW: 1) Removes the setup and management costs of passwords at sites that accept the risks of EBPR; 2) Provides single sign-on without a specialized identity provider; 3) Thwarts all passive attacks.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

General Terms

Security

Keywords

authentication, web single sign-on, password alternative

1. INTRODUCTION

Password logins are overused. Reusing a password across all web sites is risky, but managing multiple passwords results in frequently forgotten passwords. Many web sites handle forgotten passwords by emailing users a password or a hyperlink to a facility to reset the password, a technique we refer to as email-based password reestablishment (EBPR).

Simple Authentication for the Web (SAW) [6] is a new web site login approach that improves the security and convenience of EBPR. SAW utilizes email for all authentications and not just for recovering from forgotten passwords. SAW also provides single sign-on to web sites and can be fully automated so that login details are transparent to users.

1.1 Alternatives to Passwords

Password managers solve the problems associated with multiple passwords by remembering users' passwords, however, password managers generally lack portability and require significant account-level maintenance. Other alternatives to passwords require a specialized identity provider (e.g., Liberty [2], OpenID [3], Shibboleth [5]). As is evidenced by their lack of widespread adoption by web sites, finding a mutually trusted identity provider is difficult.

In 2003, Garfinkel [1] coined the term email-based identification and authentication (EBIA) to describe the general concept of using an email address as an identifier and the ability to receive email messages sent to that address as an authenticator. Garfinkel argued that EBIA's widespread use is evidence that the risks of this system are manageable, especially given that the alternatives are prohibitively expensive for many web sites.

2. SAW

At EBPR-enabled web sites, users prove their identity by using their password or by demonstrating ownership of their email address. If the ability to receive email messages is sufficient to circumvent users' passwords, why not make email the primary means of authentication and remove site-specific passwords? We identify four obstacles:

Latency In some cases, email message delivery and retrieval may require a relatively long period of time.

Lack of privacy Email messages are typically sent without cryptographic protection and are therefore susceptible to passive eavesdropping and active modification.

Convenience Password-based systems are pervasive and accepted by both users and web sites. Changing a web site's login system often requires significant time and resources as well as additional user training.

Reliance on a third party By involving an email provider in the authentication process, a dependency upon a third party is introduced. If the email provider is unavailable, the authentication process cannot succeed.

2.1 Protocol

The steps to authenticate using SAW (see Figure 1) are as follows:

Step-1. The user submits (e.g., via a webform) her email address to web server in the *Token Request* message. This request should be sent over HTTPS.

[*] This research was supported by funding from the National Science Foundation under grant no. CCR-0325951, prime cooperative agreement no. IIS-0331707, and The Regents of the University of California.

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

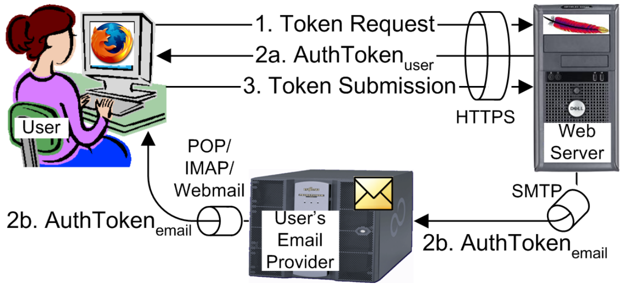

Figure 1: Based on the user's email address, specified in (1), a web site distributes two authentication tokens. AuthToken$_{user}$ (2a) is sent directly to the user while AuthToken$_{email}$ (2b) is emailed. Both tokens must be returned to the web site (3) to successfully authenticate. Each login attempt involves its own unique, short-lived, single-use tokens.

Step-2. The web site, based on the permissions of the email address, creates several short-lived, single-use *AuthTokens*.

If the address is authorized, a random secret is generated, $AuthToken_{complete}$, and split into two shares (using a conventional secret splitting scheme [4]) as follows:

$$AuthToken_{user} = AuthToken_{email} \oplus AuthToken_{complete}$$

where $AuthToken_{email}$ is another randomly generated value. $AuthToken_{user}$ is returned directly to the user over the secure link (HTTPS) used to initiate the authentication as an HTTP cookie while $AuthToken_{email}$ is emailed.

If the address is not authorized, a random $AuthToken_{user}$ is returned, and, in lieu of AuthToken$_{email}$, a human readable explanation of the failure is emailed. Always returning an $AuthToken_{user}$ prevents an impersonator from learning anything about the email address owner's permissions.

Step-3. The user returns both tokens to the site in the *Token Submission* message. If these values combine to equal the AuthToken$_{complete}$ for that particular user and token identifier, then the authentication is successful and the system uses a session-level trust preservation mechanism for the remainder of the session (e.g., a session cookie).

Due to the nature of XOR and since $AuthToken_{user}$ is sent over a secure link, passively observing $AuthToken_{email}$ is worthless.

2.2 Client-side Automation

Manually polling an email account for $AuthToken_{email}$ is inconvenient and unnecessary. Additionally, a carefully crafted phishing email that resembles a token message could lure unsuspecting users to a malicious site. By automating the process of retrieving and submitting AuthToken$_{email}$, the convenience and security of SAW is enhanced.

Ideally, web browsers would have native support for SAW to allow single sign-on to all SAW-enabled web sites.

2.3 Alternatives to Email

SAW provides personal messaging-based authentication. $AuthToken_{email}$ is easily delivered over a variety of personal messaging mediums (e.g., instant and text messaging). SAW provides a platform to explore the potential that these personal messaging systems or a hybrid combination of these mediums have for authentication.

Instant messaging, in addition to providing an attractive, low latency alternative for delivering $AuthToken_{email}$, facilitates the use of SAW by those who rely on free web mail accounts (e.g., Hotmail, Yahoo! Mail, Gmail) since the programmatic access (i.e., POP/IMAP) to these accounts necessary for client-side automation is often only available to "premium" accounts. As these providers associate free instant messenger accounts (e.g., MSN Messenger, Yahoo! Messenger, Google Talk) with each email address, users are able to leverage the same password as their email account to enjoy the benefits of SAW and client-side automation.

2.4 Prototype Implementation

We added server-side support for SAW to the Apache and Tomcat web servers as well as Wordpress, a popular web log platform. The client-side browser support is implemented for both Internet Explorer and Mozilla Firefox. The toolbar provides a recognizable, uniform interface for authenticating to SAW-enabled web sites, manages email account information, communicates with email providers via POP or IMAP, and also receives instant messages.

3. CONCLUSIONS

SAW is a simple concept. It builds on EBPR, which has already proved its utility for user authentication, and improves it by thwarting all passive attacks and raising the bar for active attacks. These enhancements make SAW a viable alternative for passwords at a substantial number of web sites. SAW has the potential to thrive because it does not require universal acceptance, modification of email providers, or significant changes to existing web site infrastructure.

SAW is an important step towards simplifying authentication and making it more convenient. It relieves both web sites and users from having to establish and manage passwords by off-loading user authentication to email providers. Although SAW does not eliminate passwords completely (users must still authenticate to their email providers), it should greatly reduce the number of passwords users need to manage. In addition, this system provides single sign-on to all SAW-enabled sites and can exploit client-side automation to speed up the login process as well as reduce the attack surface to phishing and social engineering attacks.

SAW is simple, but therein lies its strength. It is easy to understand and to implement; its benefits are significant. Its risks are clear and have already proven to be manageable.

4. REFERENCES

[1] S. L. Garfinkel. Email-based Identification and Authentication: An Alternative to PKI? *IEEE Security & Privacy*, pages 20 – 26, 2003.

[2] Liberty Alliance Project. http://projectliberty.org/.

[3] OpenID. http://openid.net/.

[4] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, pages 70-71, 1993.

[5] Shibboleth. http://shibboleth.internet2.edu/.

[6] T. W. van der Horst and K. E. Seamons. *Simple Authentication for the Web*. ISRL Technical Report 2007-1, Brigham Young University, January 2007, (http://isrl.cs.byu.edu/pubs/saw_techReport.pdf).