Applying Context to Web Authentication

John Linn, Burt Kaliski, and Moti Yung, RSA Laboratories; Magnus Nyström, RSA Security Inc.

Prepared for W3C Workshop on Transparency and Usability of Web Authentication, New York, March 2006

Abstract

Current practice for authentication between web-based users and sites is vulnerable to many forms of deception that can mislead sites about the identities of their accessing users and mislead users about the identities of the sites they access. In this environment, attackers can often obtain and reuse authenticator data. Enhancements to user interfaces and trustworthiness of underlying browser processing can help to improve many aspects of assurance. To further mitigate relevant threats, it is also important to constrain the context within which authenticator data elements can be reused, narrowing their scope consistent with their intended purpose. In this paper, we introduce our position relative to web authentication and the workshop, describe a general taxonomy characterizing the reusability of intercepted authentication data, and specifically discuss methods applicable to constraining the reuse of intercepted passwords or One-Time Password (OTP) values used for web-based authentication. This discussion provides a vehicle for consideration of user interface concerns related to security generally, and to authentication enhancements in particular.

Position Statement

We believe that improved mutual authentication between web users and sites is an important and complex problem, and one that can benefit from multiple complementary approaches. Browsers should provide their users with clear and accurate information to support informed judgments and assessments of the sites they access. Authentication-related data should be processed in a fashion that limits the potential for it to become available to hostile parties. As some authentication data is likely to fall into attackers' hands nonetheless, technical mechanisms should limit the ability for those attackers to employ it and impersonate users. In each of these areas, trustworthy interfaces and modules are important, and use of standard approaches and best practices will be important to achieve interoperable deployment. Recognizing these factors, we hope and believe that the workshop's discussions will contribute to enhanced usability and trustworthiness for authentication in web environments.

Taxonomy

In contemporary web environments, authentication data can be subject to interception by attackers seeking to reuse the data and impersonate its authorized holders. Facilities to constrain the context within which intercepted data can be effectively reused can help to limit attackers' capabilities. In this section, we introduce a structure to characterize authentication methods' resistance against data reuse by attackers.

To perform authentication, a user presents evidence, whose form varies for different methods; some examples include passwords, OTPs, and biometrics. An agent process on the user's computer constructs and transmits authenticator data based on that evidence; different methods perform different (null or non-null) transformations to construct authenticator data from input evidence. In our taxonomy, we distinguish three classes of reusability constraints that can be offered for transmitted authenticator data:

- Fully Reusable (FR). In this class, no protection is provided against reuse of authenticator data.
- Partially Reusable (PR). In this class, an attacker's ability to reuse intercepted authenticator data is limited by one or more constraints, comprising a context for the data's successful use. Many dimensions of context are possible; subsequent discussion will provide representative examples. In some cases, but not all, the context within which authenticator data can be reused is established by the original authentication event that generated the authenticator data.
- Not Reusable (NR). In this class, authenticator data is not reusable except by entities holding secret quantities that are not obtainable from the authenticator data. For example, an authenticator data object encrypted in the context of a particular session would normally not provide reusable information to an attacker lacking that session's encryption key.

Within the PR class, several types of constraints can be distinguished, and can operate independently or in combination with one another. Constraints may be provided within the authenticator data representation, as aspects of its underlying method, and/or through other measures within an overall system or its operational environment. The following list is representative:

- Destination: Authenticator data is represented in a form that is specific to a particular destination, and which would not be acceptable if presented to a different verifier.
- Origin: Authenticator data is acceptable only if presented from a certain source or class of sources. An example in this area would be a system that accepts and authenticates users only if their access attempts can be confirmed as originating within a specified perimeter.
- Time: Authenticator data is acceptable only if presented within a particular time window. This type of protection may be achieved, e.g., as an aspect of a time-based OTP method, or by incorporating timestamps within protocol messages where authenticator data is carried.
- Limited-use semantics: A particular authenticator data element will be accepted by a verifier only for a specified number of occurrences (often only one); if a further usage attempt is detected, it will be rejected. Many OTP methods provide this type of assurance; it is worth noting, however, that some of its effectiveness depends on an assumption that an attacker cannot suppress a user's legitimate authentication request from being received and processed at a verifier.

- Transaction: Authenticator data is generated or represented in a form that is bound to one or more elements of a transaction; as such, it cannot be effectively applied to authenticate transactions having different values for those elements.

Password Protection

Within the scope of this workshop and the taxonomy presented above, we recognize that most web user authentication is currently based on transfer of reusable passwords, falling into the FR class. We also recognize that the pervasive deployment and use of this approach and its associated user experience makes it unlikely for qualitatively different methods (e.g., user-level cryptographic credentials integrated with protocol transactions) to be adopted quickly. In order for today's predominant form of authentication to be reasonably secure, the following properties must hold:

- A user must not transfer her password until and unless she can authenticate the entity to which the password is to be transferred, and can determine that that entity correctly corresponds to the user's intended site
- The channel carrying the transferred password must provide confidentiality protection against attackers
- A user must employ different passwords on a per-site basis, to limit the potential for compromise at one site to impact others

The second property is commonly achieved using TLS, but the first property depends today on a complex combination of user awareness and caution, site name recognition, DNS resolution, and browser trust roots, among other elements. If TLS-protected channels are established to carry password data to inappropriate destinations, the fact of TLS usage cannot protect against misuse of the data by those destinations. The third property conflicts with user convenience. Given these concerns, even as means to enable users to authenticate the sites they access are enhanced, we believe that it remains important to limit the ability for attackers to make use of authentication data if that data is obtained inappropriately.

In this section, we present an approach to achieve the PR class, by constraining the context within which existing authenticator data based on passwords or OTPs can be used. Specifically, we tailor the authenticator data to the context of a particular destination; when used in conjunction with OTP methods, this approach adds a destination constraint to the time and/or limited-use semantic constraints that those methods may offer. This approach (unlike many alternatives in the NR class) preserves much of the operational and protocol flow associated with current practice. Further, it allows users mobility, in the sense of being able to enter their authentication evidence at multiple clients, and does not require that clients distinguish trustworthy from untrusted sites. Some aspects of the overall method have been exemplified in the PwdHash approach[1] developed at Stanford University; contemporaneous work in this area has been undertaken by RSA Security[2] and is the subject of a pending patent application.

[1] B. Ross, et al., "Stronger Password Authentication Using Browser Extensions", Proceedings, 14th Usenix Security Symposium, 2005.

[2] RSA Security, "Enhancing One-Time Passwords for Protection against Real-Time Phishing Attacks", 16 January 2006. Available at http://www.rsasecurity.com/rsalabs/node.asp?id=2991.

Today, users entering authentication evidence cannot be confident that it is being sent to an intended destination, and only to that destination. For best security, it is important to ensure that a user enters passwords or OTPs only into a trustworthy module that can be relied upon to transform them in a target-specific fashion before they are transmitted. It should not be possible for an attacker to mimic the appearance and behavior of this module's interface, thereby attracting input of evidence that could be misappropriated. Desirably, a user would invoke a secure attention sequence to establish a trusted path to this module when evidence is to be provided; for highest assurance, operating system support below the browser level appears valuable. The module would accept the user's data, and would generate an appropriate transformed authentication data representation to be sent to the destination. Further discussion of the importance of trusted paths in a web authentication context can be found in a 2005 report by Aaron Emigh[3].

One means for transforming passwords or OTPs in a destination-specific fashion involves hashing a value as provided by a user or OTP device with an identifier specific to the target site before transferring the result. The target site can be identified using one or more of its domain name, IP address, URL, and/or its public key; use of a public key identifier offers the advantage of resistance against DNS compromise. With this approach, the hashed value presented to site A would differ unpredictably from that presented to site B, even if the underlying password as entered by the user was the same for both sites. If an attacker obtains a password as hashed for site A, it cannot be transformed into a form suitable for authentication at B without either reversing the hash function or exhaustively searching the input password space to find a result that matches the hashed value as received. Further, a malefactor site M simulating the appearance of A or B and soliciting a user's evidence would receive a password transformed for M, which would be unusable if forwarded to A or B. Computational measures can be applied to the construction to increase the work factor for exhaustive searching attacks. Standardization of a specific hash-based construction would be helpful as a basis for interoperability, as would methods (e.g., embedded <object> tags) for sites to indicate support for specific transforms and algorithms.

As a further facility, it is also possible for the destination to employ its knowledge of a secret password or OTP shared with the user to generate and return a confirming result, which the module could check in order to achieve mutual authentication. Here, the module acts as a trusted agent on behalf of the user, interpreting the received result value. Two design alternatives are possible, with different integration and user interface implications: if mutual authentication fails, the user can be warned; here, it is necessary that a trustworthy user interface be provided. Alternately, further interaction on the session can be blocked; this reduces the direct user interface requirement, but may appear coarsely abrupt or arbitrary to users. If workshop discussion can lead to consensus on best practices for action when security exceptions are detected, this could provide valuable guidance for implementers.

[3] A. Emigh, "Online Identity Theft: Phishing Technology, Chokepoints, and Countermeasures", 3 October 2005. Retrieved 16 January 2006 from http://www.antiphishing.org/Phishing-dhs-report.pdf.

Conclusions

Web authentication must evolve to serve its users, by better informing and protecting them. It must aid users in determining which modules and sites can be considered trustworthy, in a tailorable and well-known fashion. It must serve a range of users, varying in their awareness and attention to security practice. As such, the technology should support sophisticated user behavior but must not rely on it. It remains necessary to construct authentication mechanisms in a manner that is resilient to user error, and that constrains the impact of compromise. We look forward to productive and collaborative workshop discussion that can help to advance each of these goals.