Identity as a Platform Service

Sam Hartman, Painless Security LLC.

Josh Howlett, JANET(UK).

April 27, 2011

Focus on Value to Applications and Users

Security mechanisms are most often discussed and advocated according to the cryptographic properties that they confer on particular applications. This kind of analysis is often based on assumptions about the key stakeholders, such as users, deployers and developers, that are often demanding and not infrequently unrealistic. For example, in the case of the Web the actual security of authentication is unlikely to be improved until users are more circumspect about their passwords; theoretical notions such as password entropy are largely moot in an environment where the plaintext is often readily obtained by simply asking the user directly for it, as we observe with phishing. It is, therefore, tempting to conclude that an authentication mechanism is not valuable unless it avoids passwords or otherwise successfully discourages users from disclosing their password.

However, experience suggests that adoption of the most widely deployed identity mechanisms have been driven by improvements to usability rather than security. Consider the Enterprise identity landscape prior to the widespread adoption of Kerberos. For users, selecting the correct credential for a particular a service was a major obstacle. For administrators, implementing centralized management of authorization was also very challenging. With the advent of Kerberos, usability was improved. Although users still had to present a password when logging into their workstation at the start of the day, they were generally not prompted for the rest of the working day. Administrators were able to centrally manage access to resources across the entire Enterprise using a single identity and authorization management system. While Kerberos did provide significant security benefits, usability drove adoption.

Similarly, identity federation technologies such as SAML have been motivated by the requirement for easier sharing of user information, such as attributes and authorizations, between business partners. This reduces costs (e.g., because a user is enrolled at a single organization), increases convenience (e.g., single sign-on) and facilitates compliance with data protection regulation (e.g., by providing necessary controls for obtaining and signaling consent). Again, security benefits are present, but were not typically the primary adoption driver.

The adoption of SSL appears to have been motivated by security. However, even there, the value is in the usability improvement. SSL provided merchants with the necessary confidence to use the Internet as a means for commerce. With SSL, new uses were enabled.

This paper argues that instead of focusing on security properties as the primary metric for evaluating identity mechanisms, we should consider the broader requirements of the stakeholders and the marketplace. This does not mean that security and other important architectural qualities such as extensibility are unimportant. Instead, having identified a market need for a particular extension, it should be provided in a manner that facilitates security and other design goals. Typically, security involves tradeoffs; an Enterprise with strong security requirements may be less willing to expose information to a web application than a consumer. Our extensions should encourage browsers to make these tradeoffs primarily at the browser configuration level with additional restrictions possibly provided by an individual web application.

How the Platform Helps Identity

The platform can provide identity services that the application cannot provide because the platform has access to information that cannot be made available to the applications. In addition, as with all other services, providing re-use of the mechanisms between applications reduces development cost and encourages incleased availability of standardized functionality.

Take the example of RFC 4559, which describes Negotiate (and thus Kerberos) authentication in HTTP and is implemented in most browsers. The Negotiate authentication scheme allows an Enterprise's authorization and identity management infrastructure to be used for web authentication. This not only enables the use of the same identity across a number of websites within the enterprise; it also allows the use of the same credentials with applications outside of the web platform such as typical e-mail and file server applications. To accomplish this, a number of decisions must be made:

- Which identity should be used for this web resource?
- Can the existing credential be used for this web resource?
- Should the platform delegate the ability to act on behalf of this user to this web resource?

The application is often not best positioned to address these questions. First, the application may be unaware of the complete set of identities possessed by the user; exposing the complete set to the application may have significant privacy considerations. Secondly, an application that prompted the user to select an identity would defeat the advantage of single sign-on in a number of cases. Finally, the decisions about credentials and delegation are policy decisions: the application may like to have access to these resources, but should it be trusted? The platform, however, already has the necessary infrastructure for addressing these issues as a result of Kerberos' existing integration into the platform. Leveraging this same infrastructure for the Web therefore improves usability because they can re-use existing configuration and policy. The fact that security is also enhanced by exploiting the platform is a fortuitous outcome, and not a primary driver.

While Negotiate demonstrates that integrating identity services into the platform is valuable, experience highlights that significantly more work is required on these services. Implementations of Negotiate often have many of the same challenges as HTTP Basic authentication, such as presenting browser dialogues to the user rather than dialogues from the web application. Significant configuration is also required to enable the mechanism; this is reasonable in the context of a single Enterprise, where the identity infrastructure is run by the Enterprise that also administers the browser; but less tenable in scenarios where the same identity needs access to resources from other security domains.

In summary, while Negotiate provides sufficiently significant usability advantages today for it to be widely deployed within the Enterprise, there is room for improvement. The next sections will consider ways to achieve this.

Identity Selection

Although innovations such as Kerberos and platform-based identity services have reduced the number of identities issued by Enterprises to their users, users are nonetheless acquiring more identities; contributing factors include:

- Enterprise software-as-a-service or out-sourcing relationships that use service-provisioned identities, rather than identities managed by the user's Enterprise.
- Users opting to use a service that has not been formally sanctioned by their Enterprise but is nonetheless useful for the purposes of their employment.

- Users wishing to use any number of the multitude of services available on the Internet, for purposes such as entertainment, social networking, banking, interactions with government and other public services, and so forth.

Therefore users typically need to choose between a number of identities if required to authenticate. Some identities may need to belong to a specific application. However, as we have seen it is often more convenient to apply the same identity to multiple services using identity federation. In a federation, service providers delegate identity management and authentication to organizations that have a long-term relationship with the user, such as the user's employer, preferred public web portal or government (these are sometimes called 'identity providers'). Technologies that use this strategy include PKI, Kerberos, SAML, WS-Federation, OpenID and OAuth.

These technologies have been used to great effect within well-bounded communities where a user is typically assumes a single role. In this case, the user only has a single identity to select from. However as a community becomes increasingly diverse, the relationships between its users and organizations become more nuanced and numerous; and the question of which organization is authorized to make claims about the user for a given service becomes more ambiguous. This is perhaps most visible within the education sector, where a user may be a member of staff at a particular organization, a student at another and a parent of children attending school. It is clearly necessary that the service provider is able to unambiguously differentiate between the user's different affiliations, and in most federations this generally resolved by requiring the user to select the appropriate organization. Unfortunately, this often presents users with perhaps hundreds of possible identity providers to use; this is sometimes known as the NASCAR problem. The problem may also be aggravated by applications that cache previously selected identity state, in an effort to reduce the number of identity selection prompts, requiring users with multiple identities to remove the cached information so that an alternate identity can be selected.

However, the platform probably knows what identities the user already has. Consequently, rather than trying to narrow down the number of identity providers that the application will accept (potentially a large if not unbounded number), the platform can start with those that are relevant to the user. If the application can provide information concerning which identity providers it accepts, the platform may be able to narrow these down further.

Enterprise Configuration

It is common for an Enterprise or the identity provider to have valuable configuration information that concerns the scope of applicability of the identity or on other important properties. For example, an Enterprise directory could easily provide the platform with information that controls which external vendors should be given a user's enterprise identity.

This type of configuration is not limited to the enterprise. Most identity provider can provide cross-application information that describes where their issued identities are likely to be useful. The platform is in a position to know many of the available identities and to coordinate this configuration information between applications.

Sharing Identity outside of the Web Platform

Most environments have applications outside the web platform. Desktops and mobile platforms typically also support writing native applications. The web platform can share identity management services with the rest of the platform. This extends all of the benefits that make identity management valuable to the web platform across all applications. In addition, this will typically reduce costs of adding or updating identity management mechanisms because mechanisms present outside the browser can be reused.

Channel Bindings

RFC 5056 defines a concept called channel binding, which describes the act of cryptographically binding different acts of authentication together (perhaps at different architectural layers, for example). For example, if a mechanism to verify passwords is used within TLS, then channel binding confirms that the authentication endpoints are the same for both the password and TLS authentications. Channel binding can provide significant defense against malicious websites, especially in the case when there is an existing relationship between two parties. Channel binding can also be used to protect transitions from one resource to another where the first resource knows the expected authentication of the second resource. When used properly, a channel binding failure indicates the presence of some sort of attack with very high reliability.

Channel binding requires support from the platform.

Identity Management Beyond the Web

Even the most casual observer of identity management could hardly fail to notice the multiplicity of technologies within the space. Each has advantages and disadvantages: no one solution is appropriate for everyone. The architectural challenge is to understand how these diverse technologies are best composed in ways that meets the stakeholders' requirements.

One of the goals of the IETF's Common Authentication Technology Next Generation (KITTEN) working group is to standardize abstract interfaces to identity management, called the Generic Security Services API (GSS-API). The goal of the GSS-API is to abstract the conversation between the application and the identity mechanism, so that applications can focus on identity details that are important to their requirements but ignore others. Some applications are completely independent of the identity mechanism and can work with all the mechanisms listed below. Other applications take special advantage of features of an identity mechanism but still find the abstraction useful.

There is significant experience using this style of infrastructure on the existing desktop platforms with OS integration for Windows, Mac and Linux among others. This architecture has been very successful at capturing the existing identity management systems. Today, within the IETF, it supports:

- OpenID
- OAUTH
- SAML both using the traditional Web SSO profile and a richer enhanced client profile
- Kerberos
- PKI

A new system, called ABFAB, has recently joined this repertoire of systems.

Project Moonshot and ABFAB

Currently, the IETF is undertaking an effort to standardize a new identity management mechanism in the ABFAB working group. This architecture is designed to provide a system that will extend the application of contemporary identity management techniques to existing applications and platforms that support the GSS-API (and SASL also, by virtue of the GS2 mechanism).

JANET(UK) is leading an effort called Project Moonshot to implement the ABFAB technology and integrate it into desktop platforms. These technologies have several advantages:

- Both the application and platform can contribute to the identity selection. This manages the NASCAR problem while allowing the user to choose any acceptable identity
- Preferences can be shared between applications and initial hints provided by identity providers or enterprise administrators. This improves the usability of the system
- Identity providers can use whatever authentication mechanism best meets their needs; supported technologies can be anything from username/password to strongly authenticated public-key credentials. The conversation between the user and identity provider is standardized to minimize the risk of phishing.

Project Moonshot chose to take an aggressively platform-orientated approach, on the assumption that a tight level of platform integration would yield substantial benefits for application developers and deployers of the technologies who already extract substantial value from these platforms' system and administrative services respectively. On the basis of initial results, this approach has been vindicated; the technology has been successfully demonstrated with a number of applications, including Apache and Firefox.

Recommendations

Any effort to improve the browser's use of identity management technologies should be based on concrete market requirements that will drive adoption, rather than obsessing on security properties that may exceed these requirements and impede uptake. Experience suggests that the key driver for security technologies is improved system usability.

We believe that work within the IETF and by developers of traditional operating system platforms clearly demonstrates the usability advantages of providing identity management as a platform service. We recommend that the web platform provides identity management functionality that includes interfaces allowing applications to select identities, provide credentials and perform authentication.

The operating system platform itself should assist with the credentialing and identity selection process when possible. While support for HTTP authentication would be nice, much of the usability benefit will come from providing these mechanisms as a service that can be used within web pages and for XML requests.

Such work should enable applications and users to select an appropriate tradeoff for issues such as privacy and security. This platform service should be abstract and allow the platform to provide a variety of concrete identity management mechanisms. We would be very interested in working to define requirements and make sure that web platform aligns with other platform work going on in the IETF and elsewhere.