# Position Paper for the W3C Workshop on Transparency and Usability of Web Authentication

15/16 March 2006

Terry Hayes, AOL

## Abstract

There several barriers to deploying better authentication methods that would be more resistant to "phishing" and other attacks, and more convenient for users. Web services that supported federated identity attributes would provide an infrastructure that allows deployment of stronger methods.

There has been some recent interest in providing web services using REST-style interaction – in particular, protocols placed directly on HTTP rather than including SOAP. HTTP does not provide an Authorization header value that is compatible with a federated environment. Development of a suitable method would benefit REST-style interaction.

## Introduction

AOL believes that strong authentication and authorization are foundational elements to support identity-based services.

The current web environment generally requires that a user who wishes to consume some service must establish an account or profile with each provider. This structure prevents the deployment of better authentication methods in several ways.

## Barriers to Better Authentication

First, the number of separately maintained accounts requires that over the course of a day credentials are entered as each web service is used. Even though we try to educate users to be skeptical of requests for credentials, they are quickly trained to fill in passwords (and other credentials) in order to view the desired resource. This action becomes second nature, and allows "phishing" attacks to succeed.
The number of requests for credentials needs to be reduced (within policy requirements) so that each individual action can be considered properly by the user, rather than being viewed as an annoying obstacle to accessing resources.

Second, stronger credentials such as one-time-password (OTP) and PKI (certificate) based authentication typically require significant investment in software development and support by the issuer of these credentials. Service providers (especially smaller ones) will never expend the effort or funds necessary to deploy these capabilities to their users. These sites will continue to use less secure methods such as passwords.

Third, the inconvenience to the user of establishing and maintaining multiple accounts and profiles means that the associated identity data will very frequently be incorrect (address, credit card, telephone etc). The credentials used to access these accounts will also be weaker, since users will typically select similar names and passwords at each site. Even if a good password is selected, its strength is reduced if it is repeated at many different service providers.

## Federation as a Solution

By allowing the user to use a single account or profile across many different web sites, these barriers can be removed.

In a federated environment the user will generally deal with a single user interface for establishing their identity using their credentials. Since the identity provider's site will be one of a handful that is used, requests from other sites will be obviously invalid. The user will expect to interact with the site in certain way (say on a daily basis). They will be more likely to notice requests that outside of this normal pattern. In addition, users may make the site the "first stop" in a web session, by using a preset bookmark or manually entered URL. By using a path completely controlled by the user and the user's software agent to access the site, the possibility of "phishing" is reduced.

Identity providers in a federated environment are much more likely to provide stronger credentials such as OTP or public key methods. The economics of developing and deploying solutions such as these are much more favorable when a direct relationship to the business (providing identity) is apparent. These stronger credentials will then result in stronger forms of identity data that can be released to services, which in turn will result in a wider variety of services that can rely on this data.

Finally, federation reduces the level of inconvenience to the user, allowing them to update identity data in a single place. The increased correctness of the data will be a benefit to service providers. Since there are fewer sites managing identity, the strength of the credentials (even passwords) used to access them will improve.

## HTTP Authorization

Recently, there has been a fair amount of interest in REST-style interfaces to web services. This includes both services accessed by web browsers and those intended to be used by client applications and server-to-server transactions. We would like to apply federation to this style of web service.

Web service protocols with layers built above HTTP provide a way to send authorization information along with service protocol message content. SOAP allows data to be added to the message header, typically a security token of some kind. The Liberty browser profile for federation uses a sequence of HTTP redirections to send messages between the service and the identity provider. After this sequence is complete, the service generally retains the authorization information in the form of a cookie associated with the web site.

For a web service using the REST model, we would like to use only HTTP.
HTTP provides an Authorization header, but there is not currently a standard method defined for that header to use it in a federated environment.

The current methods defined for HTTP Authorization (such as basic and digest) transmit an identifier and appropriate credentials for authenticating that identity. That is, the method performs *authentication* not *authorization*. The web service uses the resulting authenticated identifier to authorize a request based on other local policy information. In effect, this authorization method assumes that the identity provider is also the provider of the resource.

HTTP should have an authorization method that does not require a user identifier or user credentials. Instead, the method should define a way to transmit identity information from another source (a third party identity provider). This data should be verified by the service relying on it, and then used to grant access according to local policy.

The user agent acquires the data needed for this new authorization method by using other web services. The data could be in many forms, such as a WS Trust token, or a SAML assertion. Depending on the policy of the service holding the resource, any of the SAML assertion types could be accepted (authentication, attribute or authorization).

The authorization data might not include a specific identifier. Instead, it could include more general attributes (such as age or association membership) that might authorize the request. This would allow access to resources on a service without identifying the particular user making the request.

Standardization around such an authorization method would be welcome.