



www.i3sp.com
mattw@i3sp.com

Web Sign-On Security

The Purpose of Web Sign-on

Sign-on of users in a web server context is useful to the site provider in order to present customised information to the user. Today, many sites use this technique and certainly all that can afford to invest in the infrastructure. Some sites allow the user to set up their preferences for viewing the sites content (“my.XXX.com” sites) others to attempt to track who is downloading their software and yet others to attempt to tailor advertisements to the user, so that there site generates higher click-through rates.

Higher value web sites provide end user services that have intrinsic value, such as web based email providers and portfolio trackers. These sites contain information and access to services that it is highly undesirable that unauthorised personnel gain access to.

Still other sites allow the user to manage their finances, buy and sell stock, transfer money, manage large multi-corporation projects and access resources located on the web that have tangible monetary value.

Clearly, sign-on covers a range of services that have differing security requirements. A classifieds site with a “my” section for saving searches for the user requires low levels of security, while a site allowing a user to pay arbitrary bills from their account requires very high security. Some situations have the user desiring more security while in others the provider is more at risk.

Authentication, Authorisation and Security

These are 3 often-misused term amongst lay-people on the Internet. When considering web sign-on and access to privileged resources, all three concepts must be addressed.

Authentication is when a user is identified by the system as being a representative of a particular entity of which it has knowledge, e.g. a user is authenticated when they log into their web email account with their user name and password. Authorisation is when an authenticated entity is allowed to access certain specified resources, e.g. checking a bank balance for their account. Security addresses the concern that access to protected resources is gained only through the process of authentication and authorisation¹.

Sign-On models

Typical sign-on models involve the presentation of a username and password (the user *credentials*) to the authentication system. Rather than have this presented with every access, as in the BasicAuth HTTP model, most sites use *cookies*, wherein a “cookie” of information is returned back to the browser by the server and the browser re-presents this with every access. The server then uses this piece of tagged information to identify that this is the same user as logged on and thus a “session” is established over the top of the session-less HTTP

¹ Note that the breadth of the subject of security is not addressed here – only how it pertains to sign-on issues.

protocol. In subsequent requests, the presence of the cookie allows the identification of an authenticated user. Thus, whether the user has authorisation to access the different resources of the web site can be determined.

In this document, this is referred to as the *session identifier*.

This sign-on model is easily extended to other forms of authentication than username and password combinations. Checking using passwords, a smart card based token generator, round-robin password schemes and hardware devices all can be easily implemented.

Each scheme has its advantages and disadvantages but most suffer from some major security issues due to the nature of the sign-on schemes used.

Authentication Requirements

As seen above, different sites require different levels of authentication. These can fall into broad categories:

- *Low*: Sites where the sign-on is for presenting freely available information in a preferred format, tailoring advertisements, control of email notifications etc. Compromised sites of this nature will cause only nuisance factor to the site or user.
- *Medium*: Sites that present information that should be protected for the client for privacy reasons (e.g. stock portfolio trackers), or where the client can be misrepresented (e.g. unauthorised access to web mail accounts, online auctioning or sales). Unauthorised access to these systems may cause some harm to the client.
- *High*: Sites that allow control of a clients assets in a limited fashion (e.g. banking services where money can be transferred amongst accounts but not to others accounts) – Many may be services where the server provider is more at risk of fraud or discredit than the client – institutional service providers.
- *Extreme*: Sites that allow clients resources to be transferred to other parties (e.g. stockbroking systems, advanced banking systems)
- *Ridiculous*: Sites that should not be on the Internet: Military, government, institutional banking systems etc.

It is obvious that many sites will provide different services falling into different categories in this list. Sites can choose to implement one schema for authentication to satisfy all their requirements, or may adopt a more flexible (and expensive) approach requiring different levels or classes of authentication for authorising access to different resources.

In this model, the site provider may have different classes of users (e.g. corporate partners, institutional client and retail clients) requiring different authentication modes, or for simplifying a users access to the site, allow sign-on in graduated steps. Some services may require re-authentication with each transaction, while other a once-only sign-on. Managing this mixture of requirements can be problematic when attempting to present a coherent

interface to the user.

Basic Security Issues

Every access via the Internet passes over a (typically) unknown number of other systems, each of which potentially can monitor, intercept and/or subvert the data. Basic security on the Internet is performed using SSL or TLS². These public-key based communication layers allows a client to communicate with a server, (more or less) content that only the server is able to see the data that the client is sending.

Clearly, certain “levels” of authentication require that the user credentials be kept secure. Low and Medium level services can conceivably be implemented with standard HTTP, but higher levels of services require that a secure transport be used for presentation of the users credentials. Once the credentials have been presented and the user authenticated, it is imperative that any session identifiers sent as cookie data remain secure as well, otherwise the interloper can simply copy the cookie information and gain access to all the protected services. Thus, once a session is established over a secure transport, it must remain secure or the authentication level attained is compromised.

Similarly, if a user is authenticated over a non-secure transport, session identifiers resulting from this authentication are useless in a secure environment. Using the same identifier to authorise a user access to information returned via a secure transport is pointless if the session id has been exposed in a non-secure transport.

Resources requiring different classes of Authentication on a single web site

Many sites may wish to implement a scheme where users can have preference settings etc. in less secure sections of the site and have access to higher value resources in more restricted areas of the site. As discussed above (in *Authentication Requirements*), this can be presented to the user as a single sign-on process, or a graduated sign-on as different resources are accessed on the web-server. Irrespective, there are certain issues to be aware of.

There may be several different classes of authentication possible on a web-site, but there are only two domains that need to be separated – the secure domain and the insecure domain³. Each domain requires its own session identifier – within the secure domain (where all access is protected by a secure transport) a single session identifier is sufficient to identify all classes of authenticated user (even where a user has multiple authentication classes or levels). This identifier must not be exposed outside of the secure domain.

Correspondingly, a single session id is sufficient for identifying a user within the insecure domain. Again, note that this identifier must not be used for identifying an authenticated user

² Transport Layer Security is the name of the latest revision of the more familiar Secure Socket Layer

³ To ease management of resources and control of exposure of secure session identifiers, it is suggested that the secure areas of a web-site be grouped in one part of the web site.

within the secure domain, as it is inherently insecure.

If a single authentication policy is used for sign-on, two identifiers need to be set in cookies on the user browser – one for the secure domain and one for the insecure.

For graduated authentication policies, the insecure session id may be required for authentication to progress at the secure level, in which case, the insecure session id should be used for a superset of the web-site to the secure session identifier.

A further note: It is not entirely sufficient to simply re-present the user credentials to enter the secure domain of a web site. If these same credentials are used to authenticate the user in the insecure domain, these credentials may have been compromised. Additional credentials are required.

A Note regarding client side certificates

The technology behind client side public key certificates offers a seemingly ingenious method of securely identifying a user. The reality is depressingly different. The private part of a certificate is stored on a user computer, typically protected by a passphrase. If the encrypted certificate can be stolen, it is possible to gain the passphrase through a brute-force attack running on a separate machine. Thus in one light, client side certificates are actually less secure than a simple passphrase sent over a secure transport, since the passphrase can be ascertained without anyone being aware that a brute-force attack is in progress. On the other hand, if the computer system is secure, gaining access to the system may be difficult and use of a client side certificate for authentication, in addition to other methods, may be appropriate.