



www.i3sp.com
mattw@i3sp.com

Web Single Sign-On Technology

The Problem

Single sign-on is a problem that many corporate web sites and portals have to deal with. Typical web login solutions are provided using cookies, wherein a cookie transmits information to the web site with each request, enabling the site to keep cross-request state information, such as login details, shopping cart contents etc. The problem for many web-sites is that products, corporate partners, business names etc can have their own web-sites. These often use different domains and have different content, but the provider does not want to burden the user with having to login multiple times to the different sites, but instead wishes to present these multiple sites as one *Authentication Domain*. Using conventional login schemes involving cookies, this approach is impossible to engineer.

Limitations of Cookies

Web site programmers often use cookies to maintain state information regarding individual browser access to their web site. Cookies are set in the response to an http request and accepted by the browser dependent upon the user settings. Cookies carry certain pieces of information, being the domain or machine they are applicable to, the resource path (e.g. /cgi-bin) and when they expire, along with the cookie value itself. The cookie is then sent along with each request by the browser. Cookies, for security reasons were designed with certain limitations. No cookie can be set for a site or domain that it did not originate from. Thus a cookie set for domain foo.com can never be sent to any site in bar.com. Therefore it is not simple to use cookies to enable single sign-on across multiple domains.

The Solutions

URL Rewriting

One solution is to use URL rewriting. The web-server, upon responding to a request, rewrites every URL in the page to include a session identifier. Whenever the user navigates to a new URL via the page, the session identifier is transmitted and the web-site can obtain the users state information. This solution has many drawbacks. The user is only identified when navigating from the initial sign-on page. Typing the URL of a cooperating site into a new browser window will result in the user not being recognised and forced to sign-on again. When used in conjunction with cookies, this policy can operate efficiently dependent upon how much cross-cooperating site traffic occurs. If most navigation to cooperating sites is from other sources, bookmarks, search engines or such, then the policy in general fails.

Centralised Sign-On

The better solution is to provide a single sign-on that spans all web-sites involved in the same *authentication domain* (as distinct from Internet domain). To work across Internet domain namespaces it is required that the different domains co-operate with at least one web site in common. This site is responsible for collecting the user login information and for looking up

and storing user information on behalf of the other web sites.

The Details

Glossary of Terms

- *Id Validation*: Checking whether an id is valid. This prevents tampering, brute force attacks and makes snooping (where SSL is not used) more difficult. Typically, id's are generated by combining relevant information regarding the identity of the browser with a secret server side key, then using a message digest algorithm (such as HMAC), recombining the result with the server side secret, then using the digest algorithm again.
- *User Verification*: Checking with the central sign-on server that the session id passed to the web server is current and has been signed-on by the central sign-on server.
- *Server redirect*: When a server responds to a browser request with an instruction to visit a different URL instead.

The centralised sign-on technique utilises only widely used Internet technologies: Server side redirection and cookies.

A web-site, upon receiving a request requiring sign-on, generates a secure sign-on id¹ and embeds this id in a server side redirect URL to the central sign-on server. The URL that the user is redirected to can include context information about where the user is attempting to access, type of authentication required, etc (or the sign-on server can look up some of these details itself). The site can keep state information regarding the initial access, indexed against the sign-on id.

The central sign-on server then gathers the user credentials and authenticates them. When this process is complete, the central sign-on server generates a session id and sets it in a cookie for its own site and again does a server-side redirect back to the server that the user was originally attempting to access. The URL the user is redirected to contains parameters including the sign-on id and session id.

When a web site receives a request for a page that includes a sign-on id, the session id is checked for and extracted and then both ids are validated. If these checks pass, the site sets the session id as a cookie on the browser. The site can then contact the central sign-on server to verify the session id. Stored user information is looked up using the sign-on id (if necessary). Web-site content can then be customised for the authenticated user and the user allowed access to protected resources.

When a web site receives a request for a page that includes a session id cookie, the session id

¹ Generation of secure session-ids is best handled centrally by the site that will be managing user session information.

is validated and then the sign-on server contacted to verify the user and retrieve session information. If the session id is not valid, or the sign-on server does not verify the id, then the user must sign-on again.

If the sign-on server receives a request to sign-on a user that also has the session id cookie, the user has already been logged in. The sign-on server simply validates the id and then completes the redirect back to the originating server containing the sign-on id and session id.

Example Sign-on Transaction diagram

