Shibboleth and Application Logout Best Practices
Stephanie Obodda - 2020-11-19 - 0 Comments - in Shibboleth Single Sign On

Overview
The notion of a log out feature in a Shibboleth single sign on (SSO) services is one that is often misunderstood to be feasible. The concept (often called Single Log Out, or SLO) is that by clicking a link in one of several service providers using an identity provider (IDP), the user logs out of all SPs served by the same IDP. After a great deal of thought by a great number of highly intelligent and informed people throughout the world, the current consensus is that implementing what's known as SLO in Shibboleth is not advised. Furthermore, implementing a 'log out' button in a Shibboleth-protected application can create a false sense of security, because of the numerous ways in which a user's Shibboleth session may not in fact be destroyed by the Shibboleth log out process. Instead of rehashing an involved discussion on the matter, I'll refer interested commentators and kibitzers to Scott Cantor's article on why Shibboleth does not support SLO: [https://spaces.internet2.edu/display/SHIB2/SLOIssues](https://spaces.internet2.edu/display/SHIB2/SLOIssues).

The bottom line is that while implementing the logout functionality available in Brown's Shibboleth infrastructure may successfully log a user out of their Shibboleth sessions on the IDP and the SP, it is not guaranteed because of the numerous complexities of a distributed SSO, particularly in the case of a user logged into multiple federated services, inside and outside of Brown. The only way to ensure that a user's Shibboleth session has been terminated is for the user to close all browser tabs and windows and quit the browser, provided that the user is not using certain browsers that offer a 'feature' to deliberately maintain session state across browser sessions (that's insane!).

Why This Matters
In an SSO context, it's easy for users to forget (or never realize) that when they log into one website protected by an SSO like Shibboleth, they are also logged into other sites protected by the same SSO. If a user leaves a computer unattended and unlocked, a malicious user with access to the desktop can hijack an open browser session to masquerade as the authenticated user on any site protected by the same IDP. One of the biggest risks for all Brown users is myAccount, where a hijacked session could be used to reset the password...
and hijack services. But in theory, any SSO-protected website or service using the same IDP to which you have access could be hijacked in this way, including federated services at other institutions. Some Brown users may have a cultural predilection to rely on the manual logout feature in Brown's current SSO service, WebAuth. For those users, there will be some retraining needed to inform and remind users that Shibboleth works the same way that many online banking applications work; they remind users that the only way to ensure that the authentication session is terminated is to close all browser windows and tabs and quit the browser.

**Automatic Session Timeouts and Other Forced Re-authentication**

A notable difference between Shibboleth and Brown's legacy WebAuth SSO is that some Brown users may be accustomed to using the Logout link in the WebAuth pop-up. As discussed previously, Shibboleth does not support such a feature. There are, however, several times a Shibboleth user will be required to re-authenticate:

- The IDPs sessions timeout in 8 hours; users will be required to re-authenticate at least every 8 hours.
- By default, SP sessions timeout in 8 hours, but the session length may be shortened upon request from the SP owner. Session length cannot exceed the IDP maximum session length.
- By default, SP sessions time out after 3 hours of inactivity on that SP; SP owners or administrators can request different timeout periods if necessary. Note that this is similar to the timeouts in WebAuth, except the user does not have the choice of the timeout duration.
- If one of the two load balanced IDP servers goes offline, users authenticated on the offline IDP will need to re-authenticate on the other IDP the next time they make a request on the IDP—typically, if they go to another Shibboleth SP, or when their SP session times out. This no longer be the case if Brown adopts the Terracotta framework for load balanced IDP state management.
- If a user's IP address changes after authentication on the IDP, the user will need to re-authenticate. Examples of when this might happen:
  - A user moves from one network to another
  - A user moves from a wired network to a wireless network
  - A user moves from the Brown-secure wireless network to the Brown wireless network, or vice versa
  - A user signs into or is dropped from the VPN connection
  - A user's home internet connection changes its public internet address

**Session Management Best Practices**

Applications running under Shibboleth service providers are fundamentally responsible for managing their own application sessions. Different applications will have different security requirements, based on the application author's determination of the various risks, including those of hijacked desktop sessions. While application sessions can rely on the
Shibboleth IDP and SP sessions for authentication and attributes used by the application to authorize users, the Shibboleth sessions should be used as a guide, not the core user session. Typically, this means instantiating a session in your application based on the presence of attributes passed by Shibboleth. To determine if the SP has a valid Shibboleth session, you can look for "https://sso.brown.edu/idp/shibboleth" in the Shib-Identity-Provider attribute. Another useful attribute passed to Shibboleth SPs is the Shib-Authentication-Instant attribute. This indicates when the user last authenticated at the IDP. This is a convenient way to determine when a user last authenticated. If an application has strict security requirements, you can choose to only accept Shibboleth sessions that indicate recent authentication, where 'recent' is in the eye of the beholder.

For simple applications--like webpub apps or basic content-only websites, users can simply rely transparently on the Shibboleth session to manage state. That is, simply writing an .htaccess ACL to authorize users is sufficient. But more complex applications, particularly those with important security considerations, should use the Shibboleth session as a guide to managing their own application session state.

Tags  
Service Center