Currently participating in the experiment to evaluate replacement of AskBot (AskLO) with Discourse.

Until authorised to submit tickets to Redmine, I only had access to Discourse (single session configuration).

I disconnected from the Discourse session by using the menu command associated with "avatar" at top right of screen (labeled Log Out). I was effectively disconnected but some token (cookie?) remained on my computer.

Next time I logged in, I did not need to enter manually my credentials as the token was automatically used (even if my computer was rebooted to make sure I started afresh).

This has security implication for me: anybody able to log into my computer account can then log into my Discourse (or SSO) account without the need to validate credentials. This is not really important for my desktop, but may be serious for my laptop which is sometimes left unattended in meeting rooms.

Guilhem Moulin provided me a link to fully disconnect from SSO, but this is inconvenient (I'll try to put it in a bookmark to have it at hand).

Is is possible to have a second menu command with another label (making clear it will cause a full and complete disconnection) for this purpose?

I understand there has been a long and dense debate about SSO and that my wish may cause service disruption (closing other simultaneously active sessions).

In case this is relevant, all my computers run under Fedora Linux and browser is Firefox.

History

#1 - 2020-09-01 19:45 - Guilhem Moulin
- Category set to Single Sign-On
- Assignee set to Guilhem Moulin

André Littoz wrote:

I disconnected from the Discourse session by using the menu command associated with "avatar" at top right of screen (labeled Log Out). I was effectively disconnected but some token (cookie?) remained on my computer.

Next time I logged in, I did not need to enter manually my credentials as the token was automatically used (even if my computer was rebooted to make sure I started afresh).

Oh! Didn't understand what you were saying earlier. The SSO system (and all our other services) should definitely expire sessions when the browser closes (ie not set an expiration date on the cookie). If it works otherwise I definitely need to investigate why. Thanks for the report!

This has security implication for me: anybody able to log into my computer account can then log into my Discourse (or SSO) account without the need to validate credentials. This is not really important for my desktop, but may be serious for my laptop which is sometimes left unattended in meeting rooms.

While not an excuse for us apparently keeping sessions active across browser restarts, protecting a machine unlocked and unattended is pretty much impossible IMHO.

Guilhem Moulin provided me a link to fully disconnect from SSO, but this is inconvenient (I'll try to put it in a bookmark to have it at hand).
FWIW you can also find it in the dropdown menu at https://auth.documentfoundation.org.

Is it possible to have a second menu command with another label (making clear it will cause a full and complete disconnection) for this purpose?

Quoting what I wrote on Discourse for posterity:

In your scenario, where you’re using TDF Single Sign-On platform for a single service (namely Discourse at the moment), I agree [making Discourse logout terminate the SSO session is more secure]. However I’m not convinced this is desired for those of us who are using multiple services. Logging out from SSO currently does not terminate sessions in services: if logging out from Discourse/Gerrit/whatever were to log out from SSO, one might incorrectly believe that active sessions from other services are terminated while this is not the case (log in to gerrit and discourse, then log out from discourse and SSO — your session on gerrit remains active so anyone with access to your browser can review patches on your behalf). It’d make sense to push logout from SSO to all active sessions, but AFAICT this would require changing the authentication protocol and/or patching.

Having two logout buttons would just create confusion IMHO. And for the reason spelled above forcing logout from SSO is desired but isn’t too useful if it doesn’t terminate active sessions on other services. I think we should strive to have a single button that does terminate active sessions.

#2 - 2020-09-01 21:26 - Guilhem Moulin

- Subject changed from Security concern with SSO to Single Sign-Out system

Guilhem Moulin wrote:

André Littoz wrote:

I disconnected from the Discourse session by using the menu command associated with “avatar” at top right of screen (labeled Log Out). I was effectively disconnected but some token (cookie?) remained on my computer.

Next time I logged in, I did not need to enter manually my credentials as the token was automatically used (even if my computer was rebooted to make sure I started afresh).

Oh! Didn't understand what you were saying earlier. The SSO system (and all our other services) should definitely expire sessions when the browser closes (ie not set an expiration date on the cookie). If it works otherwise I definitely need to investigate why. Thanks for the report!

So there are two issues in the present ticket. Moved the above to its own ticket #3342 and re-titling #3341 accordingly.

#3 - 2020-09-03 11:09 - André Littoz

- Subject changed from Single Sign-Out system to Single Sign-On system

#4 - 2020-09-03 12:29 - Guilhem Moulin

- Subject changed from Single Sign-On system to Implement Single Sign-Out system

The Subject is not a typo, please don’t change it back; what appears to be desired here is to have a single system to sign out from all services.

#5 - 2020-09-03 19:22 - André Littoz

Sorry, I really thought I made a typo when I initially entered the report.

IMHO, the issue stems from the definition of a "session". There are two aspects: client and server.

Client side

From a traditional point of view, client-oriented sessions are based on some cookie and the browser will erase this cookie when it is exited. I don't think cookie erasure occurs when a tab is closed.

With such a strategy, a session is not formally closed when used presses "disconnect".

Server side

Here the server is in full control and transmits a unique pseudo-random "session token" to the client as a cookie but keeps control data on its side.

I can imagine a reference count may be associated with this "control data" to measure how many applications are active. When a disconnect request is sent, the reference count is decremented. When it reaches zero, the "control data" can be wiped out, making the "session token" invalid.

Since all this is driven by scripts, both on the server and at the client, we can imagine that a periodic handshake can detect that the client has quit without disconnecting. After a certain number of attempts (to take into account transient network losses), the server can again wipe out its "control data" to force credential solicitation on next query.

Tell me if you need more information on my configuration.
Local (browser) configuration and setup is irrelevant, it's a limitation of authentication protocols and/or authentication plugins we're using. I don't see how the reference count would provide a single log out system: if you're logged in to services A, B, and C (all 3 through SSO), then AFAICT with the reference count you'd need to cycle through each service and click the logout button to really log out from everything. So to me this seems only useful when SSO is used to authenticate to a single service, while from a user perspective SSO is arguably only useful when used with multiple services. With a Single Sign-Out system, clicking the logout button on either A or B or C or the SSO portal terminates all active sessions: on A and B and C and the SSO portal.

That being said, not sure what you mean by "all this is driven by scripts". Writing authentication protocols/mechanisms ourselves is a recipe for disaster, we stick to open protocols that have been documented, audited, and standardized. With the protocol used currently, the application doesn't call back to the portal once you've been authenticated. In particular, the portal doesn't have a state of "active" applications along nor their logout endpoint to push to (if there was such an endpoint in the first place).

Right now we're using a mix of SAML2.0 and OAuth2. SAML2.0 does provide a logout endpoint, is stateful, and aside from that it's arguably better than OAuth2 for our use case since it's all part of the same infrastructure (there is no 3rd-party IdP). I see now that Discourse has a SAML2.0 plugin, probably something to try out. However note that not all applications are SAML2.0-capable, and The Document Foundation infrastructure team doesn't have the resources nor expertise to do that kind of development work.

All in all, we can (and should) probably solve that for some services by changing the authentication protocol, but I'm not sure there is an easy way to fix that for the entire fleet of services we maintain.

There has been a Firefox update today (2010-09-06) in my Fedora distro. Firefox is now 80.0.1.

When I close the browser and reopen it, I am now logged out as expected. The "Delete cookies and site data" is checked in the preferences (didn't change during the update). I'll have to see if this makes a difference when I uncheck it.

Did that now, that should partly solve the issue as far as Discourse is concerned. Clicking the logout button on Discourse now triggers a logout event on the authentication platform too, which solves the problem you describe in this ticket: One needs to authenticate again when clicking login.

"Partly" because there is a bug in our SAML Identity Provider (IdP); it was fixed upstream but we haven't upgraded yet. What's not working yet are IdP-issued SLOs that is, logout broadcast to other Service Providers (SP). If you're logged in to Nextcloud, Discourse, and the authentication platform, then clicking logout on either of these should ideally log you out of all of these (and any other SP using SAML 2.0).

Keeping the ticket open for now. We'll probably upgrade some time in fall which should fix this in full (and also #3434).