What Does Logout Mean?

Michael B. Jones, Identity Standards Architect, Microsoft
Brock Allen, Software Security Consultant, Solliance

OAuth Security Workshop, March 2018, Trento, Italy

“Logout” can mean many different things

- This session is a discussion on what “logout” means in different contexts and what the usability, application, and security implications of the different meanings and mechanisms are
- Intent is to capture what we learn and write a summary document
  - Satisfying https://bitbucket.org/openid/connect/issues/984
  - *Create a document explaining "single logout" semantics*
- Who will be our note-taker?

Background

- Digital identity systems support end-users logging into applications
- Many systems also support logging out
- Different semantics for “logout” apply in different use cases
- Different mechanisms for achieving “logout” are used

- It’s telling that OpenID Connect supports three logout mechanisms:
  - OpenID Connect Session Management 1.0
  - OpenID Connect Front-Channel Logout 1.0
  - OpenID Connect Back-Channel Logout 1.0
- SAML 2.0 also had multiple logout mechanisms

Differences in logout mechanisms include:

- Whether logout is reliable or best-effort
- Whether only application is logged out or also the identity provider
- Whether only web applications are logged out or also native apps
- Which state is revoked/cleared by logout and which is not
  - cookies
  - access tokens
  - refresh tokens
  - HTML5 local state
  - etc.

Reasons for performing logout include:

- End-User action
- Application time-out
- Identity Provider time-out
- Detection of anomalous behavior or account compromise
- Account termination

Kinds of Logout Messages in Federated Systems

- Request from RP to IdP to log out end-user
- Request from IdP to RP to log out end-user
  - May be sent in parallel to all logged-in RPs known to the IdP
- Chained request to sequentially log out series of RPs (used in SAML)
- Logout confirmation message from RP to IdP
- Logout confirmation message from IdP to RP

- Note that hierarchies of federated systems may result in an RP with one IdP also being an IdP to another set of RPs

Communication mechanisms for logout messages

- Browser-based message delivery methods:
  - Redirect from RP to IdP
  - GET at RP iframe
  - GET at tiny/hidden RP image
  - postMessage between RP and IdP frames
  - JavaScript invocation on iframe load
  - iframe/image loaded notifications within browser
  - Redirect from IdP to RP
  - Redirection chain initiated at IdP through all RPs to be logged out
- Backchannel message delivery methods:
  - GET or POST from IdP to RP

Possible state clean-ups at RPs

- User Session State
  - Cookies
  - Browser-based storage (e.g. HTML5 local storage, index dB, etc.)
    - Requires JavaScript notification
  - Storage in native client (platform-specific and no spec for this)
- Token Revocation
  - Access Tokens
  - Refresh Tokens
  - ID Tokens

Possible state clean-ups at IdPs

- User session state
  - Cookies
  - Tokens
  - Server database entries
  - List of logged-in RPs

Logout and Auditing Information

- IdPs may keep a log of when & where end-users logged in and out
- May be used for service operator logging and auditing
- May be used by end-user to log out undesired sessions

Problems Delivering Browser-based Logout Messages

- User may close tab/browser before message delivered or acted upon
- User may navigate to a different tab
  - May suspend activity in intended tab
- iframe and/or image page loads can take time
- May not know when iframe and/or image loads have completed
  - Problem especially acute in nested scenarios
- IE/Edge zones sometimes prevent session cookie from being passed to check session endpoint
  - Can result in spurious “changed” notifications and false positives for logouts
- User disabling 3rd party cookies can interfere with state management
- Network timeouts may occur

Problems Delivering Back-Channel Logout Messages

- Requires IdP to be able to reach RP's back-channel logout URL
  - Some network deployment topologies may preclude this
- Book-keeping in RP required if cookie is the only artifact tracking the end-user's session
- In distributed/load-balanced environments, state updates need to propagate between replicas

Drilling into Logout Scenarios

(mostly courtesy of Brock Allen)

Monolithic Application (No Federation)

- Easy case
  - No distributed state
- Either logged in or logged out

Single IdP, Web App RPs

- Single company owns IdP, users' identities, and all RPs
- From user's perspective, it makes sense to logout of all active RPs

Single IdP, Native Client with Embedded Browser

- Not applicable:
  - Recall that SSO requires shared execution context, so SLO would only apply to that same shared execution context
  - Ergo, native apps possibly don’t participate in normal SSO context when using an embedded browser, thus don’t need/want to participate in SLO process
  - Somewhat moot since embedded browsers are discouraged
- Logout just means cleaning client’s tokens, and possibly using revocation endpoint

Single IdP, Native Client with System Browser

- If using the system browser for login, then it's unclear if a native client should participate in SLO and/or receive a logout notification
- Login with system browser would normally be rare/one-time, so the user's session might be timed out in the browser if the user triggers SLO in the native app
- This might only be needed if:
  - the client does not use offline access and requires the user to reauthorize on each use of the client, or
  - the cookie at the IdP is persistent and logging out of the native client wishes to also log out of the persistent cookie
- These scenarios require more thought

Federation Scenario with B2B

- Should we trigger logout of upstream business partner IdP?
  - Possibly not, since the federation gateway set of apps are logically distinct from the IdP ecosystem
  - Would also depend on each scenario/trust relationship
- Also, it’s possible to prompt user to ask if they want to trigger SLO upstream at IdP
  - But adds to end-user complexity

Federation Scenario with Social Login

- Should we trigger logout of upstream social provider?
  - (if they even support logout)
- Doubtful, as consumers are used to always being logged into them

Federated Logout (Notification to Federation Gateway from Upstream IdP)

- Should logout notification from upstream IdP trigger SLO at federation gateway and to federation gateway's clients?
- Brock would lean towards yes, but might depend on logical boundary between upstream IdP and local IdP
  - Similar to item for federated SLO with B2B

Public Kiosk

- SLO everywhere would be requirement
  - Difficult if upstream IdP is social and/or doesn't support SLO
- This would be a scenario for "close the browser", reboot the machine, install a new copy of the OS, etc. :-)
- Very hard to guarantee without a lot of control over the host OS tailored to SSO/SLO scenario
- Brock: "Personally, given what I know about security I'd never use a public computer or kiosk to authenticate (unless perhaps my credentials were some smart card style authN device)."

User Inactivity in Browser

- At RP
  - Brock doesn’t think this should trigger SLO
  - Consider a user is in two RPs in two different tabs in the browser
  - The RP in the inactive tab should not trigger SLO when the user is deemed inactive, as it will then trigger logout notifications to other RPs, including the one that is in the active tab in the user’s browser
- At IdP
  - Brock doesn’t think this should trigger logout, for the same reason as user inactivity in the RP
- Inactivity, in general, is very difficult to coordinate across all RPs and IdP
  - Nothing (that Brock is aware of) provides a coordinated “ping” to notify that the user is still active somewhere in application ecosystem

Account Termination

- Should certainly prevent new tokens from being created
- Should also revoke any revocable tokens stored at IdP (reference/refresh)
- Introspection should honor account termination as well
  - Might solve lack of revocation for JWTs, but of course requires RP to use introspection for JWTs – not likely
- Front-channel is very difficult to trigger; back-channel could be used if immediate notification in RP is required

Account Compromise

- Shares initial characteristics with Account Termination?
- Resume normal status after account recovery process & login

What Does Logout Mean? – Conclusions

- Hopefully our discussions have added information to our understanding of what logout means

Logout clearly isn't "one size fits all"!

- Volunteers to work on write-up for OpenID Connect issue welcomed

- Send additional information to:
  - Mike Jones, mbj@microsoft.com, https://self-issued.info/, @selfissued
  - Brock Allen, brockallen@gmail.com, https://brockallen.com/, @BrockLAllen