JWT Authentication in MediaWiki

Jeffrey Wang

Non-MediaWiki authentication

- Organizations want to integrate MediaWiki with their SSO platform
- MediaWiki's auth and session management code allows extensions to implement their own auth mechanisms

Current approaches

- PluggableAuth is the primary SSO auth framework
  - Game changer for MediaWiki SSO
  - Used for SAML, OpenID Connect, OAuth, Shibboleth, LDAP, etc.
  - Easily adaptable to other SSO protocols
- Auth remoteuser [sic] used in cases where the session management is already being managed by the web server
  - Less common approach
  - Requires cooperation with the web server

What’s missing?

- What if we want to support IdP-initiated auth?
- What if we want to avoid using complicated SSO protocols?
- What if we want to reuse an existing authentication token?

JWTs

- JWT = JSON Web Token
- Token represents a unique auth session
  - Issued by the identity provider (IdP)
  - Consumed by the service provider (SP)
- Useful for persisting auth session throughout the token's life
  - Can reuse token across multiple services
  - Avoids needing to log in multiple times
- Best for
  - Systems where user will be logged in to several sites at once, including MediaWiki
  - Sites are accessed from a central login system

SSO protocols vs. JWT: analogy

- Analogy for authentication: getting pulled over
  - Department of Motor Vehicles = identity provider (IdP)
  - Driver = user
  - Law enforcement = service provider (SP) - e.g. MediaWiki
- SSO protocols (SAML, LDAP, OAuth, OIDC, Shibboleth, etc.)
  - Law enforcement declares there is a need for identification to occur
  - Police calls DMV to verify whether driver is who they claim to be
- JWT
  - Driver license is enough to convey the identity of the driver when they are pulled over by the police
  - Driver license = JSON web token

SSO protocols vs. JWT: analogy

- Analogy for authorization: permission to go on field trips
  - Parent = identity provider (IdP)
  - Child = user
  - School = service provider (SP) - e.g. MediaWiki
  - The permission slip = JSON web token
- SSO protocols (SAML, LDAP, OAuth, OIDC, Shibboleth, etc.)
  - School asks parents whether child can go on field trip
- JWT
  - Parents sign child's field trip permission slip
  - The permission slip = JSON web token

Using JWTs in MediaWiki

- Was not feasible until now...
- Introducing the JWTAuth extension!

JWTAuth extension

- Does not depend on other MediaWiki extensions
- Uses Firebase's PHP JWT library
- Uses SessionManager (but not AuthManager)
  - Avoids AuthManager's rigid auth process, which is not easily compatible with authenticating with JWT

How JWTAuth works

- Auth process
  - IdP sends a POST request containing the JWT to the wiki's Special:JWTLogin page
  - If the JWT is valid, the user is authenticated and redirected to the wiki's Main Page
  - If JWT contains authorization info, also assign user groups
- How do we know it's valid?
  - JWTs must be signed
  - Symmetric and asymmetric encryption algorithms both supported
    - Symmetric: uses passkey shared by IdP and MediaWiki
    - Asymmetric: IdP has private key to encrypt with, MediaWiki has public key to decrypt with
  - Use HS256, RS256, or EdDSA

More information

https://www.mediawiki.org/wiki/Extension:JWTAuth

Questions?