TRUSTED CI
THE NSF CYBERSECURITY CENTER OF EXCELLENCE

2020 NSF Cybersecurity Summit

Client Tools for Transitioning from X.509 to Oauth2 Access credentials

Dave Dykstra, dwd@fnal.gov

NSF Cybersecurity Summit
September 24, 2020

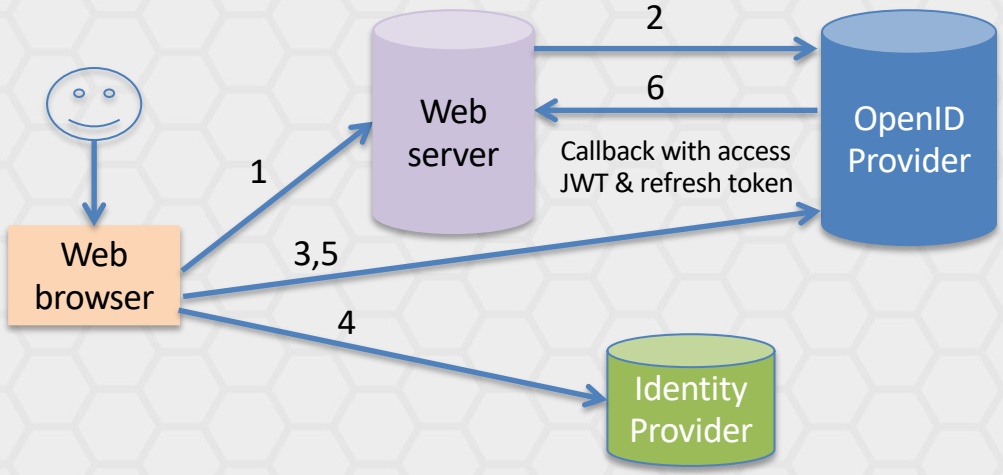
Fermilab

Open Science Grid

Background

- 2016 talk: X.509 user credentials from CILogon for Open Science Grid (OSG) and Worldwide LHC Computing Grid (WLCG)
  - Automated, hidden from users, using Kerberos to local Identity Provider (IdP) and an authentication protocol that requires no web browser (SAML ECP)
  - Store long-lived credentials in a secure server (MyProxy) to renew short-lived credentials for grid jobs
- Now moving to newer Oauth2/Open ID Connect (OIDC) Java Web Tokens (JWTs), which are good but introduce challenges:
  - OIDC assumes a web browser world, and our tools are based on command line
  - Need a new way to renew tokens for grid jobs
    - Need new secure storage for refresh tokens

Normal federated OIDC flow



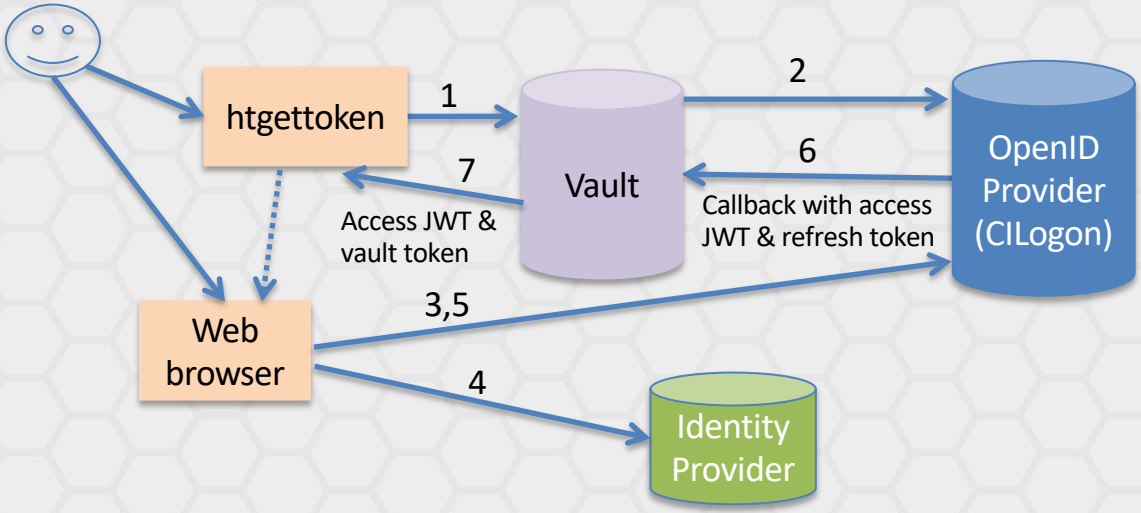
Option considered: oidc-agent

- oidc-agent plusses
  - Full-featured command line tool, good for small number of users
  - Securely stores refresh token in a per-user process and on disk
  - OIDC device flow support avoids need for callback
- oidc-agent minuses
  - User required to encrypt client details + refresh token with a personal passphrase, and type it when oidc-agent needs to restart
  - A separate OIDC client ID needs to be (dynamically) registered with the OpenID Provider for every end user
  - No good way to renew credentials sent with grid jobs, which may outlive oidc-agent processes

Option chosen: Vault with htgettoken

- Hashicorp Vault
  - Popular open source general purpose secure secret store
  - Very flexible plugin architecture and client/server API
  - Has existing OIDC and Kerberos plugins
    - Needed some extensions, submitted as pull requests
  - Manages access with its own tokens
  - We use it to store long-lived refresh tokens for many users
- htgettoken (ht from High Throughput Computing)
  - Command line Vault client to automate the flows
  - Initially authenticates via a web browser, then renews Vault access via Kerberos
  - Limited-life Vault tokens stored unencrypted

htgettoken with Vault initial flow



Conclusions

- Getting credentials almost as hidden as old system
  - Users with Kerberos only need to approve on web browser once
  - Should be able to extend Vault to support ssh-agent in addition to Kerberos
- All protocols are in common use
- JWTs are better supported and more secure than X.509 proxies
  - Can be much more purpose-specific
- Tools all open source, generally available

Links

- WLCG Authorization Working Group client tools investigation report
  - https://github.com/WLCG-AuthZ-WG/client-tools
- Vault & plugins
  - https://www.vaultproject.io/
  - https://github.com/hashicorp/vault-plugin-auth-jwt
  - https://github.com/puppetlabs/vault-plugin-secrets-oauthapp
- htgettoken
  - https://github.com/fermitools/htgettoken
- oidc-agent
  - https://indigo-dc.gitbook.io/oidc-agent/