HTCondor/Vault Integration

Dave Dykstra, Fermilab

October 18, 2021

Workshop on Token-Based Authorization & Authentication

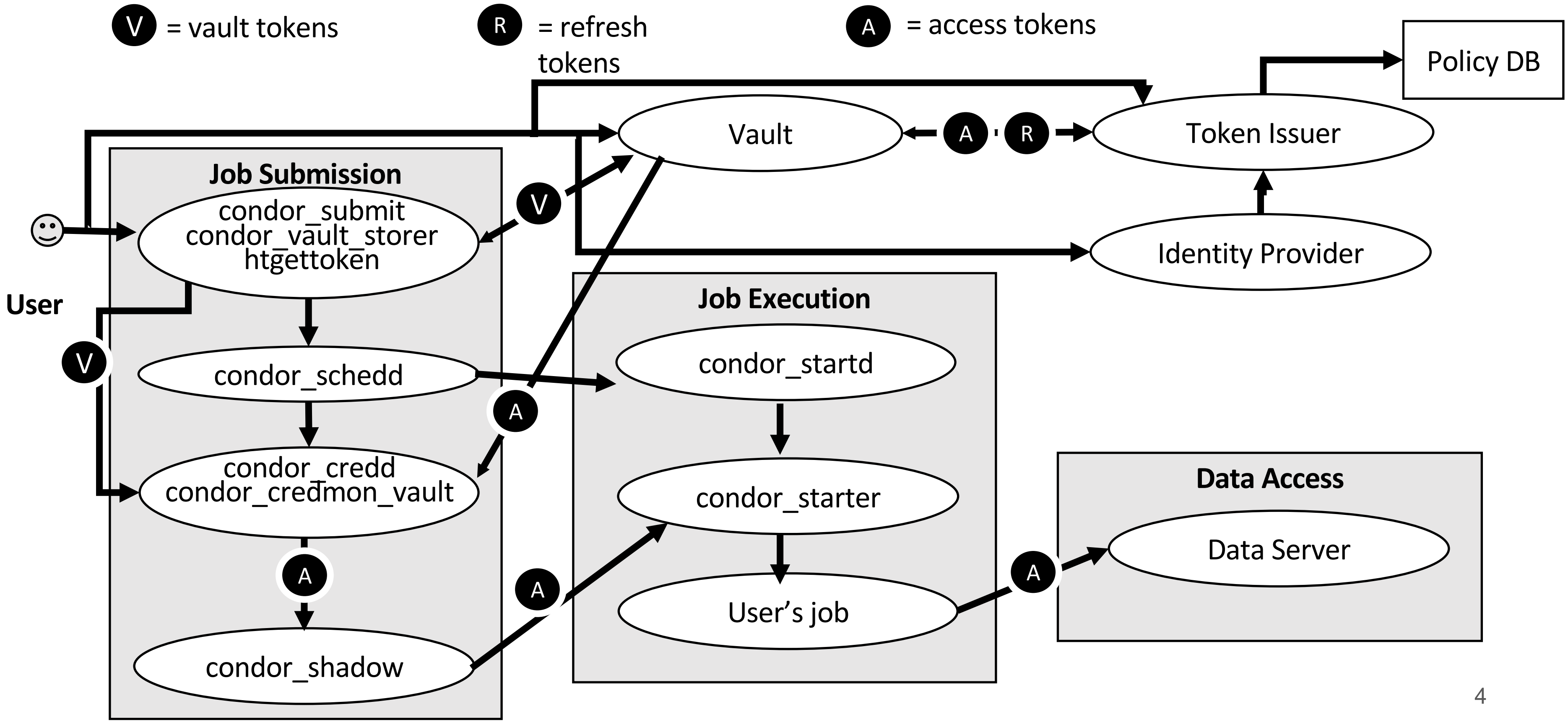
Background

- At the previous WoTBAn&Az, FNAL introduced htgettoken & Vault
    - Vault stores Oauth refresh tokens, and issues its own tokens to access them
    - Vault takes the place of MyProxy in our job submission architecture, and is also the Oauth client
    - htgettoken is the Vault client
- We use htgettoken for both submitting jobs to HTCondor and for direct file transfers
    - We want to share the token, so the same authentication can be used for both
- We have integrated the use of htgettoken & Vault into HTCondor

HTCondor Vault integration

- condor_submit can be configured to call a plugin that automatically invokes htgettoken as needed and stores a vault token in a condor_credd service
  - Vault token used by condor_credmon_vault (a plugin to condor_credd) to get new short-lived access tokens pushed to jobs
  - Vault token is extra long, 4 weeks, in order to work with jobs that are queued for a long time
    - Corresponds to time we stored proxies in MyProxy
- Submit file specifies issuer, optional role, and optionally can choose reduced audience and/or scopes
  - May obtain more than one token for a job
  - Based on previous implementation of Oauth2 credential support in HTCondor
- Vault token is stored on submit machine and in condor_credd with an extension indicating the VO & role, so can keep multiple on same machine
- Available in all current HTCondor versions

Token flow with HTCondor and Vault



HTCondor configuration

- System admin:
  - Install condor-credmon-vault rpm and set for example:

    ```
    SEC_CREDENTIAL_GETTOKEN_OPTS = -a fermicloud543.fnal.gov
    ```
- User submit file for example:

  ```
  use_oauth_services = dune
  dune_oauth_permissions = storage.read:/ #optional
  dune_oauth_resource = https://eos.cern.ch #optional
  ```
- Service names may include role, such as `cms_production`
- Handles may appended to store multiple variations for each service:

  ```
  dune_oauth_permissions_readonly = storage.read:/
  dune_oauth_permissions_write = storage.write:/
  ```
- All tokens end up in `$_CONDOR_CREDS`

Links

- Vault: https://www.vaultproject.io/
- htvault-config: https://github.com/fermitools/htvault-config
- htgettoken: https://github.com/fermitools/htgettoken
- HTCondor docs (search for vault): https://htcondor.readthedocs.io