From Identity-Based Authorization to Capabilities: SciTokens, JWTs, and OAuth

Jim Basney
jbasney@ncsa.illinois.edu
HTCondor Workshop Autumn 2020
25 September 2020
Goals for an HTC Authorization System

- Enable access to HTC!

- Implement appropriate resource/data access policies
- Ease of use
- Manageability
- Distributed/Decentralized
Authentication & Authorization Standards

- X.509: Certificates
  - Grid Security Infrastructure (GSI)
  - Virtual Organization Membership Service (VOMS)
- SAML: Security Assertion Markup Language
  - Using XML
  - Single Sign-on for Higher Education: eduGAIN / InCommon / Shibboleth
- JWT: JSON Web Tokens
  - Using JavaScript Object Notation (JSON)
  - Pronounced "jot"
  - Digitally signed, self-describing security tokens
- OAuth: Authorization Framework
  - Optionally using JWTs
  - Tokens for limited access to resources
- OIDC: OpenID Connect
  - An identity layer on top of OAuth
  - Using JWTs
X.509

Certificate Authority
Policy
End Entity Certificate
Attribute Authority
Policy
Attribute Certificate
Trust
Submit Node
Policy
Execute Node
Policy
Data Node
Policy

4
SAML

Identity Provider

Submit Node

Trust

Data Node

Policy

Authentication Assertion

Attribute Assertion

Execute Node

Policy

Policy
JWT / OIDC / OAuth
# Credentials for Authentication / Authorization

<table>
<thead>
<tr>
<th></th>
<th>X.509</th>
<th>SAML</th>
<th>OIDC</th>
<th>OAuth / JWT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Credential Issuer</strong></td>
<td>Certificate Authority</td>
<td>Identity Provider</td>
<td>OpenID Provider</td>
<td>Authorization Server</td>
</tr>
<tr>
<td><strong>Credential Verifier</strong></td>
<td>Relying Party</td>
<td>Service Provider</td>
<td>Relying Party</td>
<td>Resource Server</td>
</tr>
<tr>
<td><strong>Credential</strong></td>
<td>Certificate</td>
<td>Assertion</td>
<td>ID Token</td>
<td>Access Token</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td>ASN.1</td>
<td>XML</td>
<td>JSON</td>
<td>JSON</td>
</tr>
<tr>
<td><strong>Credential Contents</strong></td>
<td>Distinguished Names / Fully Qualified Attribute Names</td>
<td>Attributes</td>
<td>Claims</td>
<td>Claims</td>
</tr>
<tr>
<td><strong>User Identifier</strong></td>
<td>Subject DN</td>
<td>NameID / eduPersonPrincipalName</td>
<td>Subject Identifier (sub) Claim</td>
<td>Subject (sub) Claim</td>
</tr>
<tr>
<td><strong>Managing Trust</strong></td>
<td>CA Certificate Bundle</td>
<td>SAML Metadata</td>
<td>OpenID Provider Metadata</td>
<td>Authorization Server Metadata</td>
</tr>
</tbody>
</table>
## Authorization / Access Control

<table>
<thead>
<tr>
<th>Authorization Method</th>
<th>Description</th>
<th>X.509</th>
<th>SAML</th>
<th>OIDC</th>
<th>OAuth</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identity-based</strong></td>
<td>User identifiers and access control lists</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td><strong>Attribute-based</strong></td>
<td>Access policies based on user attributes</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td><strong>Role-based</strong></td>
<td>Access controls based on group memberships and roles</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td><strong>Capability-based</strong></td>
<td>Tokens allow actions on resources</td>
<td></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>
OIDC JWT Demo

Log on to https://demo.cilogon.org/ with your campus identity provider or use your GitHub, Google, or ORCID account.
OIDC JWT Demo

Paste the ID Token and Public Key into https://jwt.io/ to verify it.
Least Privilege Authorization

- Good security practice: grant only those privileges that are required
  - for only as long as they are required

- Identity-based authorization
  - Limit the privileges granted to an identity

- Attribute-based authorization
  - Use attributes to determine appropriate privileges at this time

- Role-based authorization
  - Assign privileges to roles, and activate roles only when needed

- Capability-based authorization
  - Issue tokens granting only those privileges that are required, for the required lifetime
OAuth and Least Privilege

- OAuth Access Token "scope" identifies specific actions that are authorized on resources in the token "aud" (audience)
- OAuth obtains consent from the resource owner prior to token issuance
- OAuth clients should request only those "scope" values that are required
Developing a capabilities-based authorization infrastructure for distributed scientific computing
Using the OAuth and JWT standards for distributed authorization
Implementing the Principle of Least Privilege
Visit https://www.scitokens.org/ for specifications, publications
Visit https://github.com/scitokens for open source implementations
SciTokens
JWT Demo

Visit https://demo.scitokens.org/ and click the "Set Payload" button.

Try the curl command.

* Note: This demo implements an early draft specification that used "scp" instead of "scope".
Implementing Standards

- RFC 6749: OAuth 2.0 Authorization Framework
  - token request, consent, refresh
- RFC 7519: JSON Web Token (JWT)
  - self-describing tokens, distributed validation
- RFC 8414: OAuth 2.0 Authorization Server Metadata
  - token signing keys, policies, endpoint URLs
- RFC 8693: OAuth 2.0 Token Exchange
  - token delegation, drop privileges (reduce "scope")
- draft-ietf-oauth-access-token-jwt: JWT Profile for OAuth 2.0 Access Tokens
  - authorization claims using JWT "scope" and "aud"
Implementing WLCG Common JWT Profiles

- Defines profiles for Group Based Authorization (wlcg.groups) and Capability Based Authorization (scope)
- Use cases:
  a. Identity Token with Groups
  b. Access Token with Groups
  c. Access Token with Authorization Scopes
- SciTokens supports and helped define use case (c)

https://doi.org/10.5281/zenodo.3460257
https://github.com/WLCG-AuthZ-WG
SciTokens & HTCondor

User

Job Submission
- condor_submit
- condor_schedd
- condor_credd
- condor_shadow

Job Execution
- condor_startd
- condor_starter
- User's job

Data Access
- Data Server (CVMFS / XRootD)

Policy DB

Token Server

Identity Provider

R = refresh tokens
A = access tokens
OAuth support in HTCondor is not just for SciTokens…

See next talk for details…
Thanks!

Questions?

Contact: jbasney@ncsa.illinois.edu

SciTokens Project Team:
Alex Withers, Brian Bockelman, Derek Weitzel, Duncan Brown, Jason Patton, Jeff Gaynor, Jim Basney, Todd Tannenbaum, You Alex Gao, and Zach Miller

This material is based upon work supported by the National Science Foundation under Grant No. 1738962. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.