Trusted CI Webinar Series

Title: Federated Authorization for Distributed Scientific Computing

Panelists: Jim Basney, Brian Bockelman & Derek Weitzel

Host: Jeannette Dopheide

Slides: https://tinyurl.com/yxjzlcak

The meeting will begin shortly.

Participants are muted. Click the chat button to ask a question.

This meeting will be recorded.

The Trusted CI Webinar Series is supported by National Science Foundation grant #1920430.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF.
SciTokens: Federated Authorization for Distributed Scientific Computing

Illinois: Alex Withers, Jeff Gaynor, Jim Basney, You Alex Gao
Nebraska: Derek Weitzel
Syracuse: Duncan Brown
Wisconsin: Brian Bockelman, Jason Patton, Todd Tannenbaum, 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.
The SciTokens project:

- Introduces a capabilities-based authorization infrastructure for distributed scientific computing,
- Provides a reference platform, combining CILogon, HTCondor, CVMFS, and XRootD, and
- Implements specific use cases to help our science stakeholders better achieve their scientific aims.
• The rest of the world uses capabilities for distributed services implemented through OAuth2

• The authorization service creates a token that describes a certain capability or authorization.

• Any bearer of that token may present it to a resource service and utilize the authorization.

• When you click “allow access” on the right, the client at “OAuth2 Test” will receive a token. This token will permit it to access the listed subset of Google services for your account.

• OAuth2 is used by Microsoft, Facebook, Google, Dropbox, Box, Twitter, Amazon, GitHub, Salesforce (and more) to allow distributed access to their identity services.
Using 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
- JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens (IETF OAuth WG I-D)
  - authorization claims (scope, aud), metadata for validation
<table>
<thead>
<tr>
<th>Software</th>
<th>GitHub Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python library</td>
<td><a href="https://github.com/scitokens/scitokens">https://github.com/scitokens/scitokens</a></td>
</tr>
<tr>
<td>C++ library</td>
<td><a href="https://github.com/scitokens/scitokens-cpp">https://github.com/scitokens/scitokens-cpp</a></td>
</tr>
<tr>
<td>Java client and server</td>
<td><a href="https://github.com/scitokens/scitokens-java">https://github.com/scitokens/scitokens-java</a></td>
</tr>
<tr>
<td>HTCondor CredMon</td>
<td><a href="https://github.com/htcondor/scitokens-credmon">https://github.com/htcondor/scitokens-credmon</a></td>
</tr>
<tr>
<td>SciTokens SSH</td>
<td><a href="https://github.com/XSEDE/oauth-ssh/tree/master/server#scitokens">https://github.com/XSEDE/oauth-ssh/tree/master/server#scitokens</a></td>
</tr>
<tr>
<td>CVMFS</td>
<td><a href="https://github.com/cvmfs-contrib/cvmfs-x509-helper">https://github.com/cvmfs-contrib/cvmfs-x509-helper</a></td>
</tr>
<tr>
<td>dCache</td>
<td><a href="https://github.com/dCache/dcache">https://github.com/dCache/dcache</a></td>
</tr>
<tr>
<td>NGINX</td>
<td><a href="https://github.com/scitokens/nginx-scitokens">https://github.com/scitokens/nginx-scitokens</a></td>
</tr>
<tr>
<td>XRootD</td>
<td><a href="https://github.com/xrootd/xrootd/tree/master/src/XrdSciTokens">https://github.com/xrootd/xrootd/tree/master/src/XrdSciTokens</a></td>
</tr>
</tbody>
</table>
CILogon and SciTokens

CILogon
• Federated Identity Management
• OpenID Connect
• ID Tokens

SciTokens
• Federated Authorization
• OAuth 2.0
• Access Tokens
CILogon and SciTokens

- SciTokens issuer included in CIlogon's IAM as a Service (IAMaaS) platform
- Issue tokens based on:
  - Federated identity (InCommon/eduGAIN)
  - Groups/Roles in research collaboration (COmanage, LDAP)
  - Authorization policies (per OAuth client)
  - Run-time request and approval
- Included in CIlogon Full Service subscriptions
  - https://www.cilogon.org/subscribe
Collaboration and Interoperability

- TAGPMA Workshop on Token-Based Authentication and Authorization (Nov 30 - Dec 1 2020)
  - https://indico.rnp.br/event/33/
  - Participation by WLCG, Globus, LIGO, XSEDE, Fermilab
  - Cyberinfrastructure transitioning from X.509 user (proxy) certificates to OAuth/JWT

- Next steps:
  - Follow-on workshops
  - JWT Profile harmonization
  - Hackathons & Interop Testing
## Threat Model

<table>
<thead>
<tr>
<th>Threats</th>
<th>Mitigations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credential Exposure</td>
<td>Short lifetimes for access tokens</td>
</tr>
<tr>
<td></td>
<td>Encrypted transit</td>
</tr>
<tr>
<td></td>
<td>Well-protected refresh tokens</td>
</tr>
<tr>
<td></td>
<td>Token revocation</td>
</tr>
<tr>
<td>Granting too much access</td>
<td>Least-privilege delegation</td>
</tr>
<tr>
<td></td>
<td>Token exchange to drop privileges</td>
</tr>
<tr>
<td>Malicious client</td>
<td>Client registration and vetting</td>
</tr>
<tr>
<td></td>
<td>Client revocation</td>
</tr>
<tr>
<td></td>
<td>Per-client policies</td>
</tr>
<tr>
<td>Issuer compromise</td>
<td>Key revocation via Authorization Server Metadata</td>
</tr>
</tbody>
</table>

References:
- RFC 6819 (OAuth 2.0 Threat Model and Security Considerations)
- RFC 8725 (JSON Web Token Best Current Practices)
OSCRP Analysis

Science Asset: Embargoed/Internal Data
- Lost Effort Reproducing Data
- Inaccessible or Lost Data
- Credential Exposure

Concerns
- Loss of non-reproducible data
- Corrupted Data
- User Grants Too Much Access Scope
- Malicious Client Obtains Authorization

Consequences
- Incorrect Science Results
- Exposed Data
- Reputation Loss
- Legal Action

Avenues of Attack

https://www.trustedci.org/oscrp
Access to Data in OSG & LIGO
Motivation for Switching

• GSI and GridFTP were always “niche”, but even more so now

• Reference implementations were abandoned by developers

• Internet community has moved to tokens, OAuth and others
What is a “SciToken”

- A SciToken is a JSON Web Token (JWT, RFC7519) with an defined schema.
Token Flow: Technologies

- **HTCondor** - Create, renew, and transfer SciToken from the submit host to the execute host.

- **CVMFS** - Authorize user on the execute machine and cache data locally.

- **XRootD** - Manage regional caches and origin(s), authorize access by token.
Token Flow

- Token is created on the Submit host, no OAuth required
- Implies: “If you can submit jobs on the submit host, you have access to LIGO data”
Token Verification

- CVMFS on WN
- Cache server
- Origin Server (on first download)
- Demo on the LIGO + SciTokens testbed
- Uses a local issuer (described earlier)
- Example script:

```sh
#!/bin/sh -x

# Show the structure of the credential directory
ls .condor_creds/
cat .condor_creds/*

# Decode the token (HTCondor outputs a json with access_key as a member)
./convert-token.py > token
export TOKEN=`pwd`/token

# List and checksum a file from the protected area
ls /cvmfs/ligo.osgstorage.org
sha1sum /cvmfs/ligo.osgstorage.org/frames/
```
$ ls .condor_creds/
scitokens.use

$ cat .condor_creds/*
{"access_token": "eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImU0MmUifQ.eyJqdGkiOiI5ZjI2MGU0Ni04YTQ0LTQ0MmMtYTgzOS05YjM5O...Q5MDJ9.G2L0wBUfyndXUNhXjud7we_GhH37hVTXyNCAFVcc4NcVT7qBK1QY3udqrao21UuRxF7fcs45duNecO7mzXVlKQ", "expires_in": 1200}

$ ls /cvmfs/ligo.osgstorage.org
frames  powerflux  pycbc  test_access

$ sha1sum /cvmfs/ligo.osgstorage.org/frames/...800896-4096.gwf
89ca1ad3ac3a03050627167a009d232071b62f35  /cvmfs/ligo.osgstorage.org/frames/...800896-4096.gwf
CVMFS uses authenticated caching infrastructure.

$ ls /cvmfs/ligo.osgstorage.org
frames  powerflux  pycbc  test_access

$ sha1sum /cvmfs/ligo.osgstorage.org/frames/...800896-4096.gwf
89ca1ad3ac3a03050627167a009d232071b62f35 /cvmfs/ligo.osgstorage.org/frames/...800896-4096.gwf
Recent Developments

- Requires HTCondor OAuth issuer on submit host, OSG is the test case
- XRooD 5.0+ is released with TLS support and infrastructure is updated
- SciTokens support is integrated into XRooD and is built by default in the next releases
SciTokens, WLCG, & HTCondor
The WLCG Authorization Working Group has been working to transition the international WLCG infrastructure to tokens:

- US project need to interoperate with the WLCG; the SciTokens project has been engaging with this group.
- Biggest deliverable so far is the WLCG Common JWT Profile, which incorporates much of the SciTokens ideas.
  - Sufficiently similar that the SciTokens C++ library can interoperate with either WLCG or SciTokens profiles.
- Serves as a forum for implementers, users, and standard-setting to meet.
WLCG Common JWT Profiles

- Available at https://doi.org/10.5281/zenodo.3460257; developed on GitHub.
- Defines policies for Group Based Authorization and Capability Based Authorization.
- Use cases:
  1) Identity Token with Groups
  2) Access Token with Groups
  3) Access Token with Authorization Scopes
- SciTokens supports and helped define #3
WLCG Interop

- SciTokens libraries support both SciTokens and WLCG profiles
- The CILogon token service supports issuing tokens using either profile
  - We are pursuing further harmonization
- Developing guidance on when to use group-based authorization versus capability-based authorization
- Participated in periodic Hackathons to ensure interoperability between implementations

WLCG Hackathon #1 - from back in January 2020 when we could still travel!
A wide suite of client tools used in WLCG must be adopted to use token authentication. These include:

- File upload / download CLI tools. Requires **discovery** of token from environment.
- Bulk data transfer between sites. Requires **discovery** and **delegation** of token to third-party service.
- Job submission. Requires **discovery**.
- Pilot submission. Requires **discovery** and **delegation**.

Note data transfer and job management parallels!
The HTCondor Software Suite sits at the core of many cyberinfrastructure projects:

1. The HTCondor-CE is used by sites to accept remote job submission. For this use case, HTCondor must accept authentication & authorization using SciTokens.
2. The HTCondor access point (SchedD) manages job execution. HTCondor must acquire, manage, and refresh the correct tokens for jobs.
HTCondor uses a custom binary protocol (CEDAR) for communication; since the beginning, its security handshake has allowed for multiple authentication protocols.

- In the 8.9.x series, support for SciTokens-based authentication was added.
- A TLS session is established to determine the identity of the remote server.
- If valid, the client sends the server a token over the TLS session.
  - The server authenticates the client based on token contents.
HTCondor - Token Management

- HTCondor has its own repository of tokens for users and the services their jobs require:
  - The `condor_submit` command-line tool contacts the `condor_credd` daemon on behalf of the user. The user does not need to take any specific action.
  - The `condor_credd` works with another daemon called the `credmon` to create, sign, and place the token in this repository.
  - The `credd` provides the user-facing API while `credmon` provides token management and, potentially, OAuth2 flows.
The job information in the queue is updated to reflect that it has a SciToken associated with it.

The token is monitored in the HTCondor repository even while the job is idle so the job will not attempt to run using an expired token.

When the job is scheduled for execution, the SciToken is securely transferred to the execute machine for use by the job.
• The “job sandbox” is the working directory for the job and holds all the jobs input and output files during execution.

• A directory called “.condor_creds” is created in the job sandbox, and inside this directory is the file “scitokens.use” containing the JWT.

• The environment of the job contains “_CONDOR_CREDS” which points to the full path of the credential directory.

• The job can now easily locate and use the SciToken.
• Support for SciTokens was added during the 8.9.X development series and will be fully supported in 9.0.0.

• Recently in version 8.9.10 we added support for “LOCAL” jobs, which are jobs that are submitted and run locally on the submit/scheduler machine.

• This allows jobs in a DAG to locally acquire data files as part of a larger workflow, for example.

• It also allows a user to use a simple HTCondor job to acquire a SciToken on the submit machine, if desired.
Visit
https://scitokens.org/
for more info.

Contact:
bbockelman@morgridge.org
dweitzel@unl.edu
jbasney@illinois.edu
Questions?

Click on the chat icon to type a question
Community Updates

- **Today** @ 3pm EST: Robert Hanisch (NIST) Preliminary Research Data Framework (RDaF)
  
  https://register.gotowebinar.com/register/3969890497866755597

- NSF Overview of Secure and Trustworthy Cyberspace (SaTC) Program
  
  **Friday Jan 29** @ 1pm EST
  
  https://tinyurl.com/yyuyvmql

- Trusted CI Webinar: **Mon Feb 22** at 11am EST
  
  Topic: The CARE Lab: Application, Research, and Education w/ Aunshul Rege

- February 2021: Trusted CI accepting applications for Engagements in the second ½ of 2021

  trustedci.org/application
About the Trusted CI Webinar series

To view presentations, join the announcements mailing list, or submit requests to present, visit: trustedci.org/webinars or email webinars@trustedci.org

Next Webinar: Monday February 22nd at 11am Eastern
Topic: The CARE Lab: Application, Research, and Education with Aunshul Rege