SciAuth: Deploying Interoperable and Usable Authorization Tokens to Enable Scientific Collaborations

Jim Basney <jbasney@ncsa.illinois.edu>
Brian Bockelman <bbockelman@morgridge.org>
Derek Weitzel <dweitzel@unl.edu>

2021 NSF Cybersecurity Summit

This material is based upon work supported by the National Science Foundation under Grant No. 2114989. 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.

Why SciAuth?

- Transformation underway for authentication and authorization in NSF cyberinfrastructure: from X.509 user certificates to JSON Web Tokens (JWTs)
    - Building on prior work from SciTokens
- An opportunity to realize security benefits:
    - Apply the principle of least privilege
    - Improved support for federated identities (InCommon)
    - Improved support for attribute, role, and capability-based authorization
    - Reduce reliance on coarse-grained identity-based authorization (impersonation)
    - Build on well-supported, widely-used JWT libraries
- With coordination across science projects (LIGO, OSG, WLCG, etc.)
    - For interoperability across infrastructures
    - With common approaches to integration with science software and workflows
    - Working together to maintain/improve reliability/security throughout the transition and beyond

Capability-based Authorization for Distributed Science
Or, “How I learned to stop worrying about identity”

- **Authorization** - determining access rights or privileges to a given resource - is something we do in everyday life:
    - My ID card gets me into my office building.
    - My passport allows me to get into a new country (if we ever get to travel again!).
    - A baseball ticket gets me access into a baseball game.
    - The Zoom URL got everyone into this meeting!
- I think of it as a function that takes some inputs and returns some number of privileges.
- Unfortunately, in computing, we too often intermix the concept of authorization with **authentication**!

$$A_Z(\ldots) \text{ -> } P_1, P_2, \ldots, \; P_N$$

Not every scheme requires auth’n as inputs to auth’z.

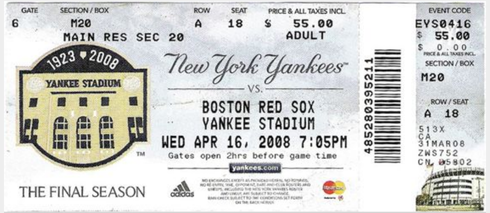
Authentication: establishing an identity for a remote entity.

Potentially, username and password is the most familiar authorization scheme:

- A remote entity provides a username/password (credentials) to authenticate to an identity.
- This identity is mapped to a list of authorized access rights.

Scheme	Credentials	Authentication	Authorization
Gmail login	Password, 2FA	Username	Access to your inbox
Office building access	ID card	Identity in HR database	Elevators
International Travel	Passport	Identity according to US Government	Enter Switzerlan
Baseball Game	Ticket	**NONE!**	Sit in section 4, seat 34B
Webinar	Zoom URL	**NONE!**	Attend this wonderful summit!

Note you can have *traceability* even if you didn’t use authentication for authorization!



Distributed Scientific Computing

Often in distributed scientific computing, the resource provider has a relationship with an organization, *not* the individual:

- In the OSG, the organization requests resources from sites and these join a global pool.
- When computing on AWS for a collaboration, someone centrally handles the billing for the resources launched. Not every user sends their credit card to Amazon.
- In XSEDE community accounts, the resource provider interacts with the community and the community interacts with the user.
- **Counter-example**: for traditional XRAC allocations, the user gets an individual account on the resource.

Scientific Computing and Identity Mapping

Historically, the OSG used client X.509 certificates (and some related extensions for group attributes and impersonation) to authenticate users and identity mapping as an authorization scheme:

- The X.509 certificate established a global identity.
- The site established an identity mapping function to a local identity (e.g., a Unix account).
- The client is authorized for any privileges given to the local identity.

Problems?

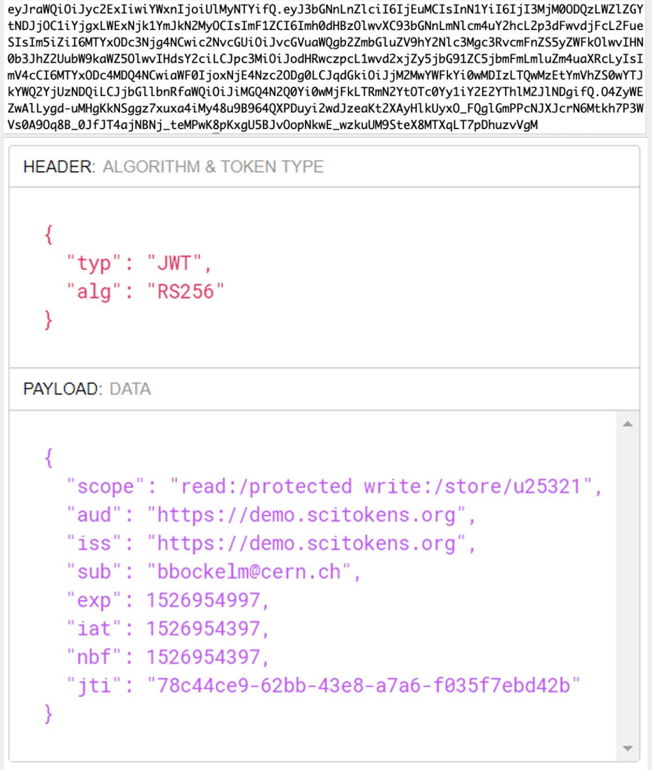
- No fine grained authorization. Your “power of attorney” travels to every remote host on the planet.
- Resource doesn’t have the relationship with the user; the actual authorization decisions belong to the organization *within* their space allocation.

Scientific Computing and Tokens

A **capability** is an (unforgeable) token that authorizes a given action on an object.

- Your house key is a capability: anyone with the key is authorized to enter the door. No need to establish an identity to use a house key!
- The capability says “What you can do, not Who you are!”

Within our community, we have built on top of the JSON Web Token (JWT) standard to allow distributed token verification and a set of agreed-upon capabilities. See doi:10.5281/zenodo.3460258



```
eyJraWQiOiJyc2ExIiwiYWxnIjoiUlMyNTYifQ.eyJ3bGNnLnZlciI6IjEuMCIsInN1YiI6IjI3MjM0ODQzLWZlZGY
tNDJjOC1iYjgxLWExNjk1YmJkN2MyOCIsImF1ZCI6Imh0dHBzOlwvXC93bGNnLmNlcm4uY2hcL2p3dFwvdjFcL2Fue
SIsIm5iZiI6MTYxODc3Njg4NCwic2NvcGUiOiJvcGVuaWQgb2ZmbGluZV9hY2Nlc3Mgc3RvcmFnZS5yZWFkOlwvIHN
0b3JhZ2UubW9kaWZ5OlwvIHdsY2ciLCJpc3MiOiJodHRwczpcL1wvd2xjZy5jbG91ZC5jbmFmLmluZm4uaXRcLyIsI
mV4cCI6MTYxODc4MDQ4NCwiaWF0IjoxNjE4Nzc2ODg0LCJqdGkiOiJjM2MwYWFkYi0wMDIzLTQwMzEtYmVhZS0wYTJ
kYWQ2YjUzNDQiLCJjbGllbnRfaWQiOiJiMGQ4N2Q0Yi0wMjFkLTRmN2YtOTc0Yy1iY2E2YThlM2JlNDgifQ.O4ZyWE
ZwAlLygd-uMHgKkNSggz7xuxa4iMy48u9B964QXPDuyi2wdJzeaKt2XAyHlkUyxO_FQglGmPPcNJXJcrN6Mtkh7P3W
Vs0A9Oq8B_0JfJT4ajNBNj_teMPwK8pKxgU5BJvOopNkwE_wzkuUM9SteX8MTXqLT7pDhuzvVgM
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "RS256"
}
```

PAYLOAD: DATA

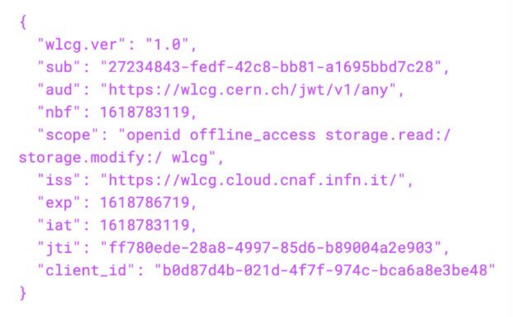
```
{
  "scope": "read:/protected write:/store/u25321",
  "aud": "https://demo.scitokens.org",
  "iss": "https://demo.scitokens.org",
  "sub": "bbockelm@cern.ch",
  "exp": 1526954997,
  "iat": 1526954397,
  "nbf": 1526954397,
  "jti": "78c44ce9-62bb-43e8-a7a6-f035f7ebd42b"
}
```

Not a new idea! See: “Jack B. Dennis and Earl C. Van Horn. **1966**. Programming semantics for multiprogrammed computations. Commun. ACM 9, 3 (March 1966), 143–155. DOI:10.1145/365230.365252”

Scientific Computing and Tokens

Each token has a ‘issuer’ (maps to the scientific organization) and a set of fine-grained capabilities.

- Capabilities are evaluated with respect to the issuer’s authorization.
- Remote resources need to map the issuer to resources. Do *not* need to know the individual users.
- Especially for file access, we can go fine-grained down to individual files.



```
{
  "wlcg.ver": "1.0",
  "sub": "27234843-fedf-42c8-bb81-a1695bbd7c28",
  "aud": "https://wlcg.cern.ch/jwt/v1/any",
  "nbf": 1618783119,
  "scope": "openid offline_access storage.read:/
storage.modify:/ wlcg",
  "iss": "https://wlcg.cloud.cnaf.infn.it/",
  "exp": 1618786719,
  "iat": 1618783119,
  "jti": "ff780ede-28a8-4997-85d6-b89004a2e903",
  "client_id": "b0d87d4b-021d-4f7f-974c-bca6a8e3be48"
}
```

storage.read:/,
storage.modify:/

Read / write within the issuer’s namespace at a storage endpoint

SCI TOKENS

- Demonstrated 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://scitokens.org/ for specifications, publications
- Visit https://github.com/scitokens for open source implementations

Using Standards

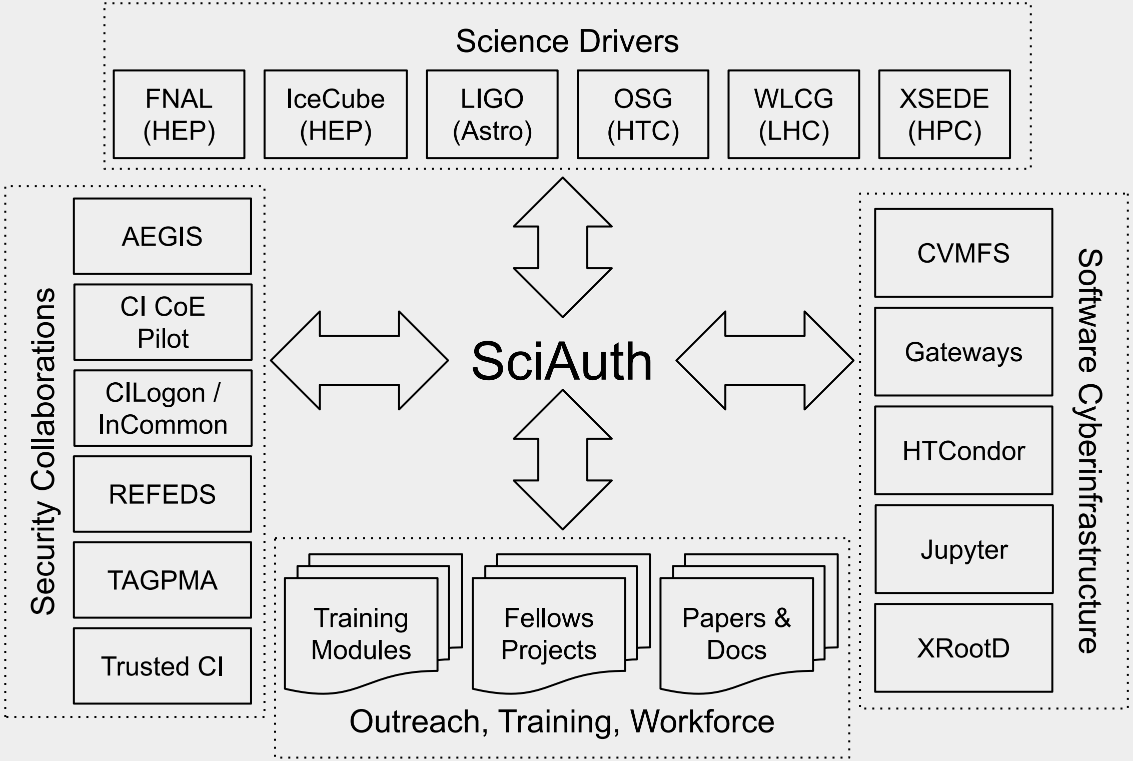
- RFC 6749: OAuth 2.0 Authorization Framework
  - token request, consent, refresh
- RFC 7519: JSON Web Token (JWT)
  - self-describing tokens, distributed validation
- RFC 8414: OAuth 2.0 Authorization Server Metadata
  - token signing keys, policies, endpoint URLs
- RFC 8693: OAuth 2.0 Token Exchange
  - token delegation, drop privileges
- JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens
  - authorization claims (scope, aud), metadata for validation

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

SciAuth Approach



Science Drivers

FNAL (HEP)

IceCube (HEP)

LIGO (Astro)

OSG (HTC)

WLCG (LHC)

XSEDE (HPC)

SciAuth

Security Collaborations

AEGIS

CI CoE Pilot

CILogon / InCommon

REFEDS

TAGPMA

Trusted CI

Software Cyberinfrastructure

CVMFS

Gateways

HTCondor

Jupyter

XRootD

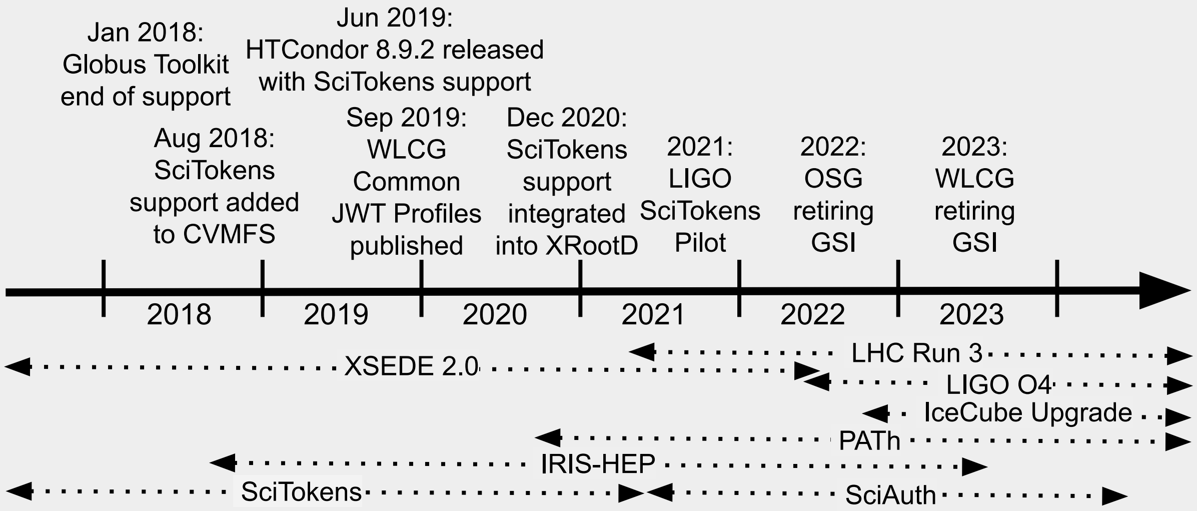
Outreach, Training, Workforce

Training Modules

Fellows Projects

Papers & Docs

What is the timeline?



Jan 2018: Globus Toolkit end of support

Aug 2018: SciTokens support added to CVMFS

Jun 2019: HTCondor 8.9.2 released with SciTokens support

Sep 2019: WLCG Common JWT Profiles published

Dec 2020: SciTokens support integrated into XRootD

2021: LIGO SciTokens Pilot

2022: OSG retiring GSI

2023: WLCG retiring GSI

2018

2019

2020

2021

2022

2023

LHC Run 3

XSEDE 2.0

LIGO O4

IceCube Upgrade

PATh

IRIS-HEP

SciTokens

SciAuth

SciAuth Activities

Areas:

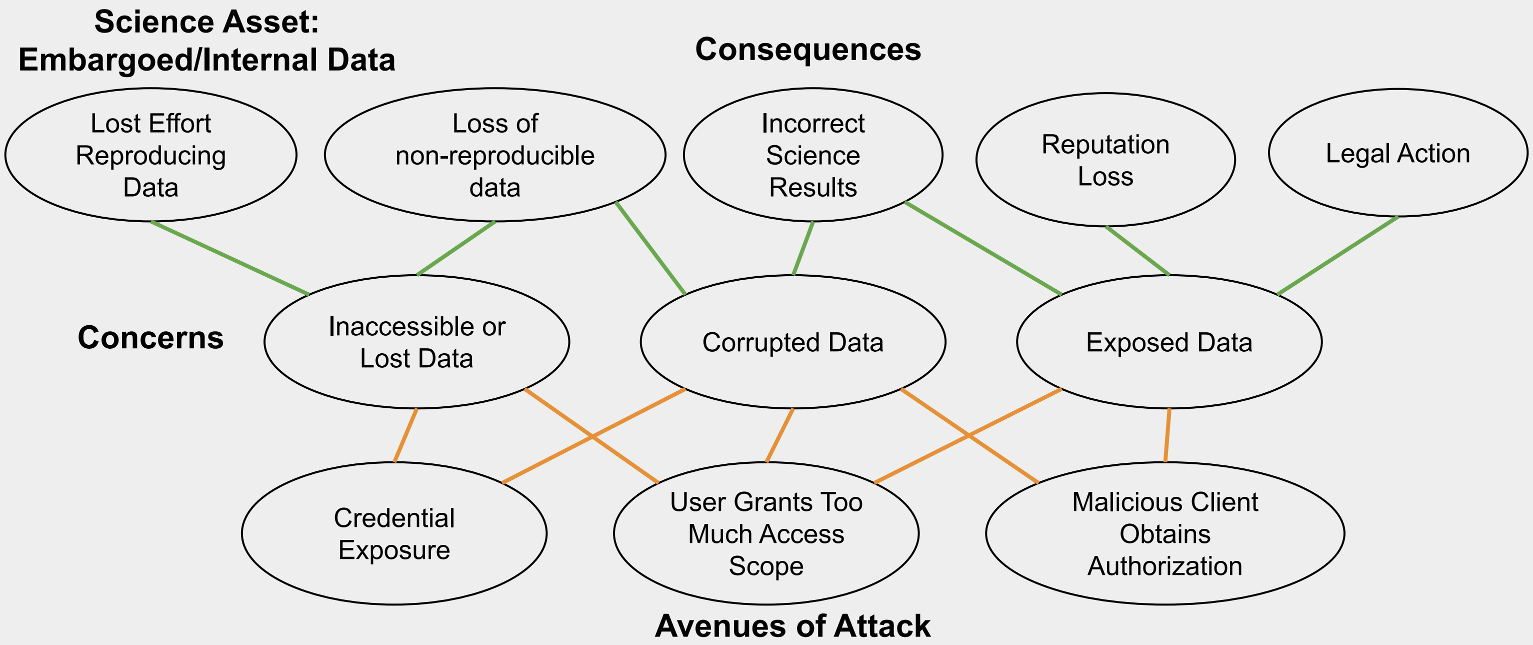
- Coordination, Outreach, and Training
- Security
- Software (demonstrations, integrations)
- Standards (JWT profiles, interop)
- Workforce Development (student fellows)

Tasks in Security Area:

- Threat Model
- Operational guidelines for token issuers
- Assessment of JWT implementations
- Token issuer peer reviews
- Tabletop exercises
  - Refresh token compromise
  - Token issuer compromise
  - Submit node compromise
  - Identity provider compromise (SIRTFI)

SciAuth is about supporting the community's transition to tokens
Not about developing new software or capabilities

Threat Model


Science Asset:
Embargoed/Internal Data
Consequences
Lost Effort Reproducing Data
Loss of non-reproducible data
Incorrect Science Results
Reputation Loss
Legal Action
Concerns
Inaccessible or Lost Data
Corrupted Data
Exposed Data
Credential Exposure
User Grants Too Much Access Scope
Malicious Client Obtains Authorization
Avenues of Attack

https://www.trustedci.org/oscrp

Threat Model

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

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

https://sciauth.org/notebook-demo

Webinars & Workshops

Recent Webinars (https://www.trustedci.org/webinars):

- Jan '21: SciTokens: Federated Authorization for Distributed Scientific Computing
- Jul '21: A capability-based authorization infrastructure for distributed High Throughput Computing in Open Science Grid

Upcoming Events:

- Oct 14-15 OSG Token Transition Workshop https://sciauth.org/2021/10/14/OSG.html
- Oct 18 Summit Workshop on Token-Based Authentication and Authorization https://sciauth.org/workshop/2021/

Workforce Development - SciAuth Student Fellows

- Now accepting applications!
- Seeking students who:
  - are interested in tokens!
  - are currently enrolled at an accredited U.S. higher education institution. Both graduate and undergraduate students are eligible.
  - will reside in the United States during the 12 week fellowship period (schedule to be determined by fellow and mentor).
- Travel is not required. All fellows program activities are conducted online.
- Fellows each receive a $1,000 stipend ($333.33 per month for 3 months) to support their research.
- For more info: https://sciauth.org/fellows

Thanks!

Visit
https://sciauth.org/
for more info.

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