SciTokens at LIGO

Ron Tapia (PSU ICDS/IGC) and James Clark

OSG Token Transition Workshop: Oct 14-15, 2021

What is a SciToken?

- Formally: A JSON Web Token (JWT, RFC 7519) with a defined schema
- “Token” comes from “Bearer Token” a concept in OAuth 2.0
- Example: curl to access an authenticated “web API”
- Tokens started as meaningless strings. JWT encodes JSON in tokens
- JWTs come can be cryptographically signed (or not)
- SciTokens are cryptographically signed JWTs
- If you **trust** the public key used to sign a SciToken, you can **trust** the payload
- Explore SciTokens: https://demo.scitokens.org/

A SciToken (encoded)


```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImtleS1yczI1NiJ9.eyJ2ZXIiOiJzY2l0b2tlbjoyLjAiLCJhdWQiOi
JodHRwczovL2RlbW8uc2NpdG9rZW5zLm9yZyIsImlzcyI6Imh0dHBzOi8vZGVtby5zY2l0b2tlbnMub3JnIiwic2NvcGUiOiJyZ
WFkOi9mcmFtZXMiLCJleHAiOjE2MzI1ODY1MjIsImlhdCI6MTYzMjU4NTkyMiwibmJmIjoxNjMyNTg1OTIyLCJqdGkiOiI4ZDgz
OTUyMi0xZTYwLTQwYzItOTI2OC1lYzA4OGM0ZTFjMzEifQ.Lab15pt2h9Y0ZUOWk0vS9F_3Vn0fVArbn9XFTPh9Q_87qpE5jBlq
b5bCgD3wQCpyxSSG9QOPlEf0MZOuh2aOJn9MMFICToZkT7-JRpNEBYU04Km_kqDKPYf-bgn0QoaSofLlkciJdJ6DQnT6RF22tF7
Ry8sA6ZDtW8R7WVa4ETKFOUkFE2aqXZvpZ033ZE586wR7OKNWFgFqbUQYLyh-j4K4DuJ0-LGD8NP9Y2ScJU1iWJT2pVtQPSzkLu
JTw8T4dZFTPj2evH8WKEimOOnGP-W06_abbJWAWlNzwsJxnPKjwh_Xbj6cbHiKwpMUqRfo3AfKfv6fbs_aNH0h_FY6ZA
```


- Three parts: Header, Payload, and Signature
- Fun fact: Base64 encoding cannot output a period

A SciToken (decoded)

Header:

```
{
  "typ": "JWT",
  "alg": "RS256",
  "kid": "key-rs256"
}
```

Payload:

```
{
  "ver": "scitoken:2.0",
  "aud": "https://demo.scitokens.org",
  "iss": "https://demo.scitokens.org",
  "scope": "read:/frames",
  "exp": 1632586522,
  "iat": 1632585922,
  "nbf": 1632585922,
  "jti": "8d839522-1e60-40c2-9268-ec088c4e1c31"
}
```

How are SciTokens used?

- A service trusts and issuer (or not)
- Trusting an issuer means trusting public keys served by the issuer
- When presented with a SciToken, a server verifies it:
  - Ensure token is valid (properly formatted)
  - From `iss`, find issuer public keys (RFC 8414: Authorization Server Metadata)
  - Use `kid` to find the right public key from the issuer
  - Compare signature based on the public key and `alg` to token signature
- Verify that `aud` applies to the service (service decides)
- After verification, server interprets the `scope` as capabilities granted to the presenter

How are SciTokens Acquired?

- Easy: some process (an **issuer**) with access to authorization information and a private key gives a SciToken with an appropriate scope to an authenticated user.
- But hard: you have to implement an architecture that fits in with existing authn/authz infrastructure and existing workflows.
- Fortunately, others have blazed a trail.
  - HTCondor local issuer
  - HashiCorp Vault + CILogon

LIGO

LSC

X.509?

What is an X.509 Certificate?

- Just like certificates used for websites.
- A Certificate Authority (CA) signs the certificate that contains an identity (analogous to website name).
- If you trust the CA that signed the certificate, then you trust that the whoever has the corresponding secret key is the entity identified.
- Just like a browser trusts the identity of a website because it trusts a set of CAs and one of them signed the website's certificate.

Concretely...

```
-----BEGIN CERTIFICATE-----
MIIEQjCCAyqgAwIBAgIDI6+XMA0GCSqGSIb3DQEBCwUAMGsxEzARBgoJkiaJk/Is
ZAEZFgNvcmcxFzAVBgoJkiaJk/IsZAEZFgdjaWxvZ29uMQswCQYDVQQGEwJVUzEQ
  ...
wvqLaskGt0v+80QlOH+AIjvi4pdloEwSebTnbNqA1Dgq0GQR1bLB6wavGhYIwtYa
amw1OKipi7xO9i5yNtb/dTJJ77QVSv9zp9F7g8f4xYYr1cITZDs=
-----END CERTIFICATE-----
-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQCuiy4Iny3Y4dqB
ln5Vhj4ryusETidGLdJ7EbO+URhhmAP0ATiyj91W29u/78VZhN2BqBgmmDeX7YYq
  ...
FY9qfv4TY/bKkgepS9/CcUsUz5cO+WweHcDQpfCaqK9jXX9trcpN/rb9TAVPYAMI
gsRDPzmGDkqIB2lNkx84vzNK6w==
-----END PRIVATE KEY-----
```

The Certificate Contains

```
Data:
Version: 3 (0x2)
Serial Number: 2338711 (0x23af97)
Signature Algorithm: sha256WithRSAEncryption
Issuer: DC=org, DC=cilogon, C=US, O=CILogon, CN=CILogon Silver CA 1
Validity
     Not Before: Oct  1 14:37:42 2021 GMT
     Not After : Oct 13 03:42:42 2021 GMT
Subject: DC=org, DC=cilogon, C=US, O=LIGO, CN=Ron Tapia ron.tapia@ligo.org
Subject Public Key Info:
     Public Key Algorithm: rsaEncryption
          Public-Key: (2048 bit)
...
Signature Algorithm: sha256WithRSAEncryption
37:05:8e:60:35:c6:c3:e7:21:f6:da:bc:36:13:d9:f8:5a:9a:
...
```

Where We Started

The LIGO SciTokens Team

- System administrators and programmers
- Mostly, not IAM people
- Mostly IAM adjacent
- Mostly concerned with running services that require authn/authz

X.509 for Authentication

- X.509 certificates for authentication
- Possession of cert implies identity
- Each service responsible for authorization configuration
- Authorization via group membership (LDAP)
- Possessor of certificate is entitled to *all* capabilities granted to identity
- Services using X.509: XRootD, CVMFS, DQSegDB, GWDataFind, GraceDB, GridFTP

Services using X509

- **CVMFS:**
  - LIGO hosts embargoed instrument data in CVMFS for distributed HTC (e.g. OSG) workflows; embargo is currently enforced via X509 credentials, with proxy certs passed along with HTCondor jobs
- **GWDataFind**:
  - instrument data-discovery utility; users to query for the location of files containing gravitational-wave detector data for consumption by data analysis pipelines.
- **DQSegDB**:
  - data quality segment database service & client package used to store, access instrument status metadata
- **GraceDB**:
  - Gravitational-Wave Candidate Event Database, provides a centralized location for aggregating and retrieving information about candidate gravitational-wave events
- **Rucio**:
  - Bulk archival data management and replication; operator authentication currently through SSH & transfers between GridFTP end points authenticated with delegated X509 proxy

The Destination

SciTokens Motivation

- OSG plans to retire Grid Community Toolkit (Jan 2022)
  - Implications for CVMFS, GSI OpenSSH, Grid FTP
- Improved security:
  - Capabilities based authorization vs identity based authorization
- LIGO observing run O4 early start date: June 2022

SciToken Goals

- Replace X.509 certificates with SciTokens
  - Retire ligo-proxy-init
- Replace grid-mapfile authorization with capabilities-based authorization
  - Grid map files are used by sites to associate X.509 distinguished names with a local users
- Migrate to federated identity
  - Remove reliance on LIGO.ORG kerberos
  - Kerberos supported but not required

SciTokens Use Cases

- HTCondor Jobs
  - Access data: XRootD, CVMFS, StashCache
  - Access GraceDB
- CLI Tools on cluster submit nodes
  - DCC, GraceDB, DQSegDB, GWDataFind
- Robots:
  - Cron jobs accessing DQSegDB
  - CI jobs (GitLab)
- The researcher's laptop
  - Only 64-bit Linux *must* be supported
  - Other operating systems are supported as best-effort

First Steps

Consult Experts

- Jim Basney (NCSA) and the SciTokens/SciAuth projects
  - https://scitokens.org/
  - https://sciauth.org/
- Dave Dykstra (FermiLab) help with HashiCorp Vault/CILogon
- Brian Bockelman (Morgridge Institute for Research) XRootD
- Bi-weekly working meetings
- OSG Slack

HTCondor Local Issuer

- SciTokens generated by HTCondor credmon
- `iss` in the token set to https://scitokens.org/ligo
- Static website based on https://github.com/scitokens/ligo
- Private key configured into HTCondor
- Public keys manually added to https://scitokens.org/ligo/oauth2/certs
- OSG XRootD configured to trust https://scitokens.org/ligo
- OSG XRootD configured to map scopes to file system paths

Local Issuer Lessons

- **Issuer** is an overloaded term
  - The value of `iss` in the SciToken payload (serves public key)
  - The generator of SciTokens (uses private key to sign token)
- Easy Condor-only solution
- Not easily adaptable to non-HTCondor use cases

Vault + CILogon

- SciTokens served by HashICorp Vault server
  - vault.ligo.org
  - CLI client: `htgettokens`
- Vault configured to use CILogon:
  - cilogon:/client_id/caltech/ligo/test
  - cilogin:/client_id/caltech/ligo/prod
- `iss` in SciToken set to: https://cilogon.org/ligo
- JWKS discovery: https://cilogon.org/ligo/.well-known/openid-configuration
- JWKS (certs): https://cilogon.org/oauth2/certs
- XRootD configured to trust https://cilogon.org/ligo

Vault + CILogon Features

- htgettoken paper by Dave Dykstra
  - https://github.com/fermitools/htgettoken/files/6063416/CHEP21_Paper_Htgettoken.pdf
- HTCondor integration
- Kerberos support & convenient CLI → easy integration & adoption with existing workflows
- Support for long-lived processes/robots
  - Method 1: User stores refresh token in a vault path accessible by a Kerberos credential
  - Method 2: Vault admin gives user an indefinitely renewable vault token
- Supports *researcher laptop* use case
- Direct line & support from developers :)

The Token Zoo

- Access Token
  - A SciToken. This is what is presented to services.
- Refresh Token
  - A long-lived token used to obtain an access token. Vault stores these.
- Vault Token
  - A string (no semantics) used to access a vault server. Lives about a week. To get a Vault token you must authenticate with vault using either OIDC or Kerberos.
- IDToken
  - A JWT used internally by HTCondor. **Not** a SciToken.

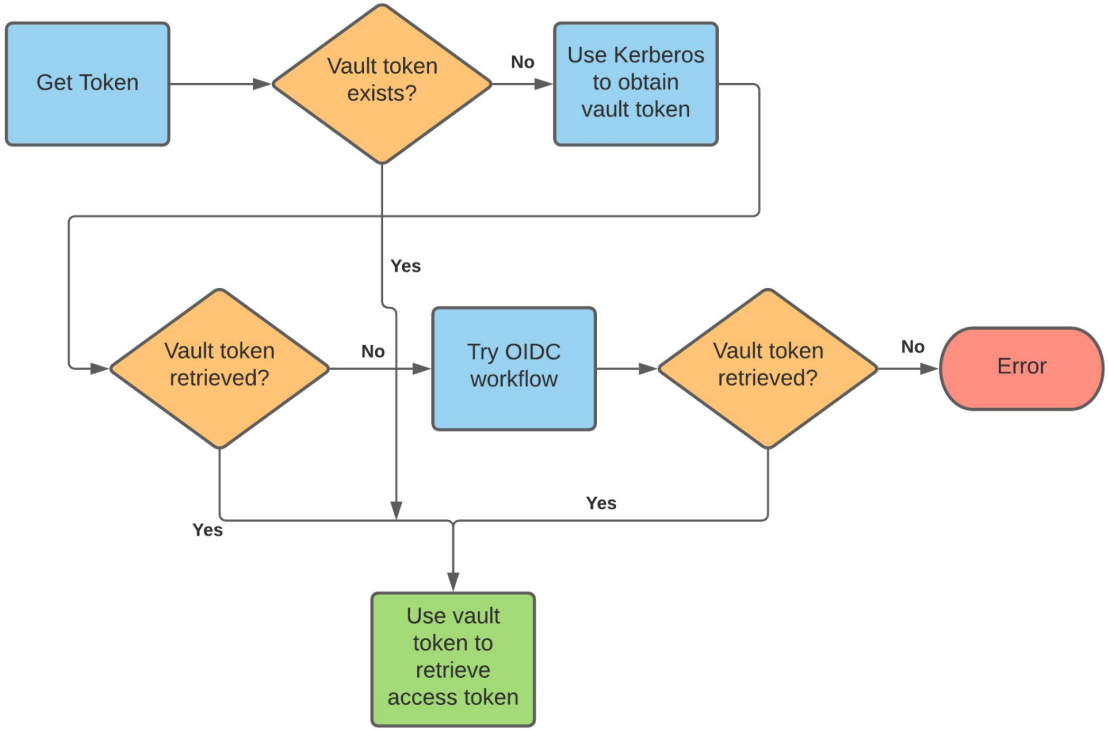
LIGO

HashiCorp Vault

LSC

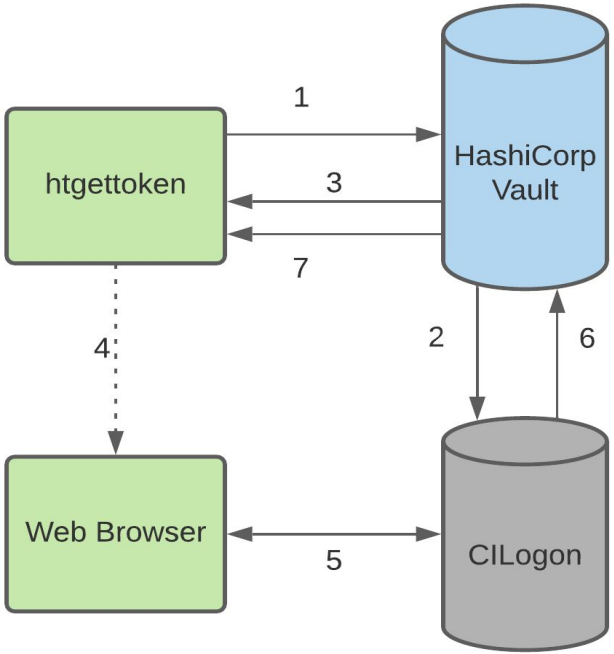
- Secure path-secret storage
- Secrets can be dynamic
- Supports authentication plugins (`vault-plugin-auth-jwt`)
- Supports secret backend plugins (`vault-plugin-secrets-oauthapp`)
- Supports kerberos authentication

htgettoken Flowchart



OIDC Workflow

Only done if kerberos fails.



1. htgettoken contacts vault server.
2. Vault contacts CILogon to start transaction.
3. Vault responds with a URL and then htgettoken asks the user to use a browser to complete the workflow.
4. The user uses a browser to complete the workflow.
5. The user is redirected to a CILogon URL, where the user selects an identity provider and authenticates.
6. After successful authentication, CILogon contacts vault and sends vault refresh and access tokens.
7. Vault responds to htgettoken with a success or failure message. Upon success it sends a vault token and an access token.

OIDC Workflow Video

Permission denied at 1:08. Use gear to set quality to 1080p.

HTCondor with Vault

- *Using Vault as the OAuth client* in HTCondor Admin Manual
- Install `condor-credmon-vault` from the HTCondor yum repository
- Vault config based on: https://github.com/fermitools/htvault-config
- Tokens fetched using https://github.com/fermitools/htgettoken
- `htgettoken` is available in OSG 3.5 yum repository

Current vault deployment:

- vault.ligo.org: single VM (2 CPUs, 2G RAM, 20G HDD), hosted by LIGO lab @ CIT
- Expect to exploit native HA support for production, machine specs TBD

LIGO

LSC

Now

Current State

- Development/testing server vault.ligo.org
- Researcher Laptop use case supported for CVMFS (note: Kagra)
- HTCondor use case supported for CVMFS
- Robot use case supported
- OIDC workflow an alternative to kerberos
- Path forward for other services that use X.509

Next Steps

Finalize Namespaces

- Requires coordination
  - CILogon configuration
  - LDAP/Grouper groups
  - Service configuration/behavior
- Audience values
- Scope values

Outstanding Issues

- Can use SciToken to access XRootD metadata, but not data.
  - This is a known issue and being worked on
- Tracing/auditing
  - Services need to be able to associate access via a token with a responsible party
  - Identity is encoded in the sub claim.
  - Use LIGO username as sub claim `albert.einstein`
  - Honor system to not use sub for authorization
  - GDPR implications for VIRGO users?

Timeline for a rolling transition (i.e., X.509 → mix of X.509 & scitokens → scitokens):

- ~End of 2021 / start of 2022: Production-level Vault service and CILogon configuration, integration/support for authenticated LIGO data in CVMFS
- ~June 2022: Transition all services by start of next observing run (O4)

LIGO

LSC

Acknowledgements

Acknowledgements

- This work was supported by NSF Grants PHY-1764464, PHY-1700765 and PHY-2110594
- This material is based upon work supported by OAC-1738962

LIGO

LSC

Questions?