Panel: LIGO's use of SciTokens

Panelists

Jim Basney (NCSA, SciTokens), moderator

Duncan Brown (Syracuse, PyCBC, SciTokens)

Zach Miller (UW-Madison, HTCondor, SciTokens)

Derek Weitzel (Nebraska, OSG, SciTokens)

Duncan Meacher (UW-Milwaukee, LIGO)

WoTBAn&Az 2020 - November 30, 2020

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.

Panel Outline

- Access to LIGO Data with SciTokens (Derek)
- SciTokens & HTCondor (Zach)
- SciTokens & LIGO/Virgo DQSegDB (Duncan Brown)
- SciTokens & GWDataFind/GraceDB (Duncan Meacher)
- Discussion (moderated by Jim)

Access to LIGO Data (Derek)

SciTokens Project

- The SciTokens project aims to:
  - Introduce a ***capabilities-based*** **authorization infrastructure** for distributed scientific computing
  - Provide a **reference platform**, combining CILogon, HTCondor, CVMFS, and XRootD
  - **Implement specific use cases** to help our science stakeholders (LIGO and LSST) better achieve their scientific aims

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

Decoded EDIT THE PAYLOAD

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "ES256",
  "kid": "key-es256"
}
```

PAYLOAD: DATA

```
{
  "iss": "https://demo.scitokens.org",
  "exp": 1585665764,
  "iat": 1585665164,
  "nbf": 1585665164,
  "jti": "10b2d6db-8e8d-4635-a66c-0b255580bf73",
  "scope": "read:/",
  "aud": "StashCache-LIGO",
  "sub": "dweitzel"
}
```

Encoded

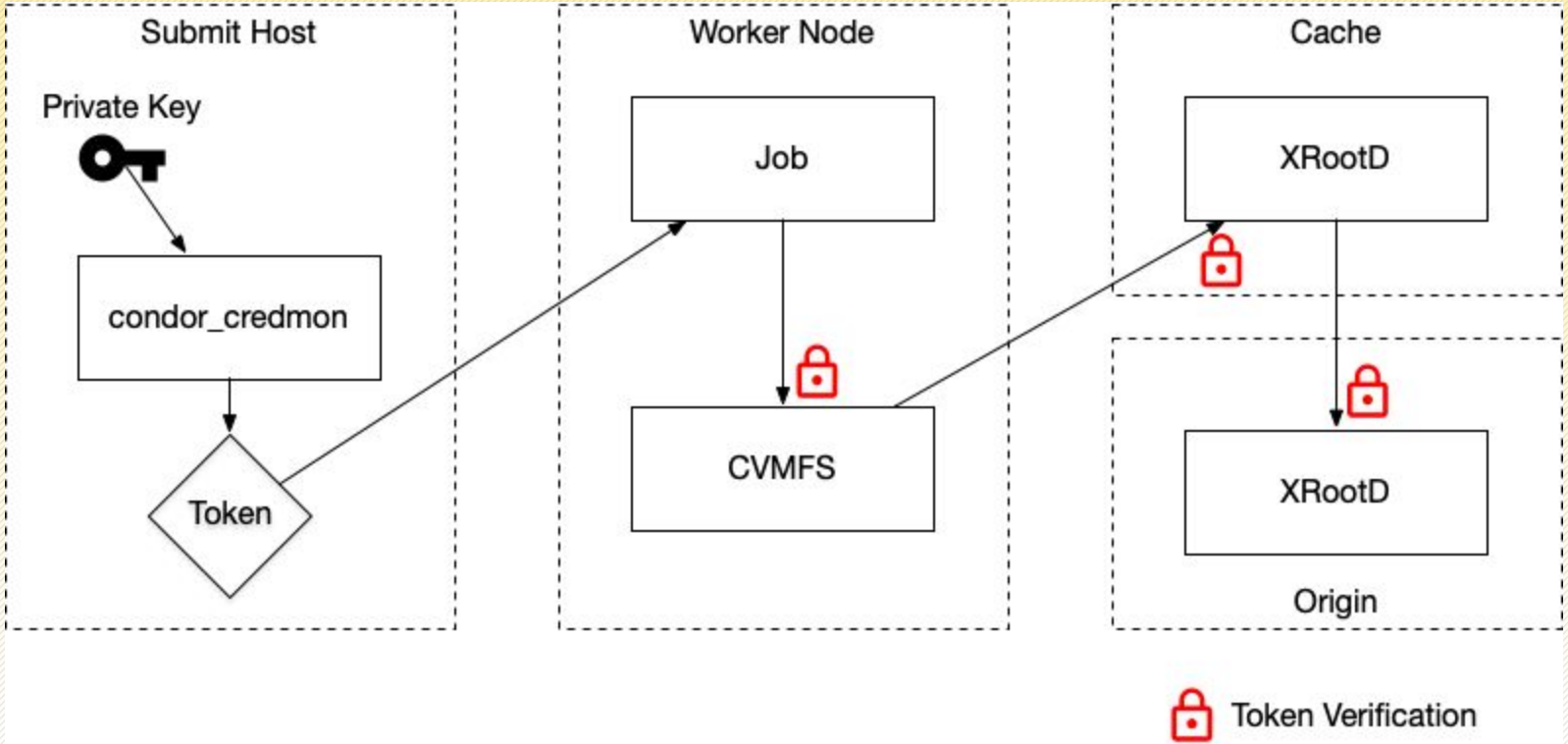
```
eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NiIsImtpZCI
6ImtleS1lczI1NiJ9.eyJpc3MiOiJodHRwczovL2Rlb
W8uc2NpdG9rZW5zLm9yZyIsImV4cCI6MTU4NTY2NTc2
NCwiaWF0IjoxNTg1NjY1MTY0LCJuYmYiOjE1ODU2NjU
xNjQsImp0aSI6IjEwYjJkNmRiLThlOGQtNDYzNS1hNj
ZjLTBiMjU1NTgwYmY3MyIsInNjb3BlIjoicmVhZDovI
iwiYXVkIjoiU3Rhc2hDYWNoZS1MSUdPIiwic3ViIjoi
ZHdlaXR6ZWwifQ.XR8Fx9F7gadMdsiVIJIJEDeF41RF
kjBG1BYkhI8-
d2pNWSkDlBQc9tQPJrid1_0QvDFyqwTdVhC5_Vy7Gy-
RfA
```

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.

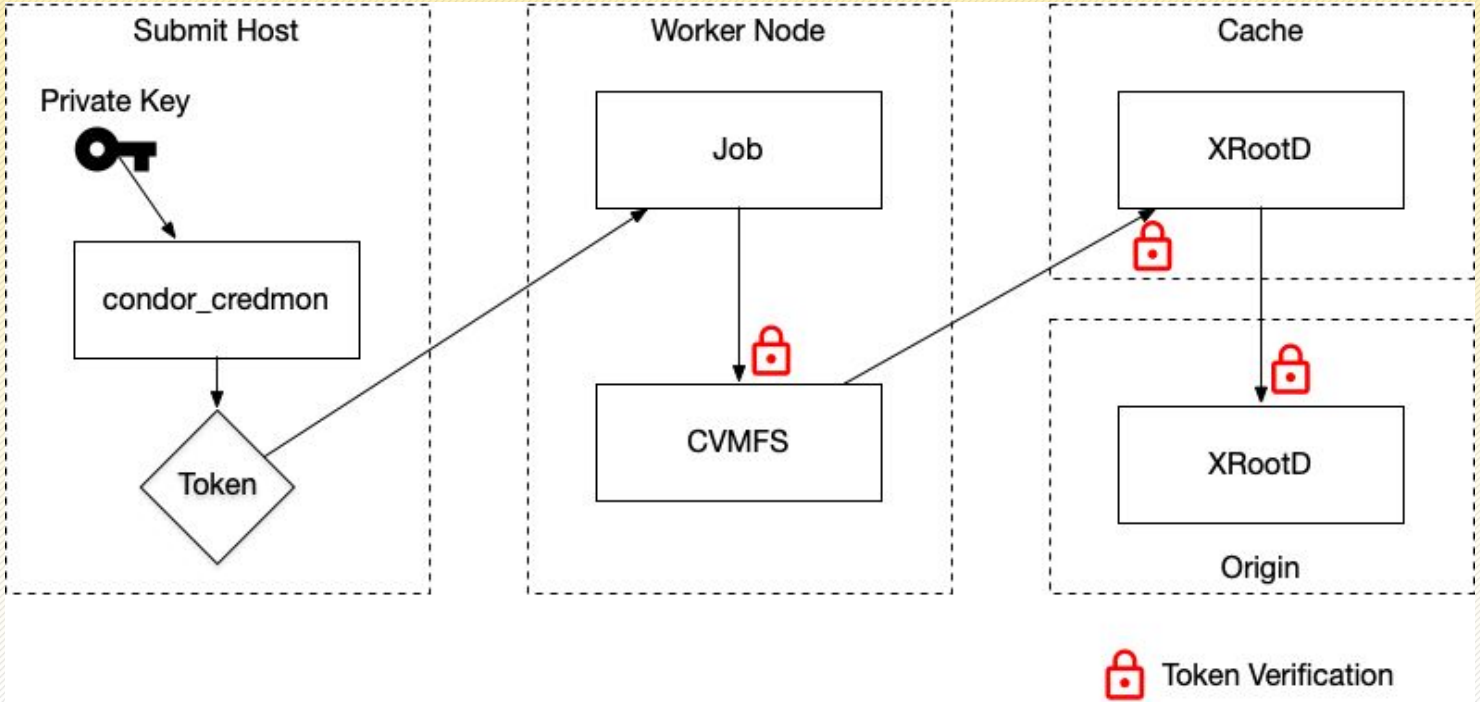
Token Flow

- 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)



Recent Developments

- Requires HTCondor OAuth issuer on submit host, OSG is the test case
- XRootD 5.0+ is released with TLS support and infrastructure is updated
- SciTokens support is being integrated into XRootD and will be built by default in the next releases

SciTokens and HTCondor (Zach)

SciTokens and HTCondor

- A SciToken is acquired during job submission.
- 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.

SciTokens and HTCondor

- The job information in the job 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.

SciTokens and HTCondor

- 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.

SciTokens and HTCondor

- 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.

SciTokens & SegDB (Duncan Brown)

LIGO/Virgo DQSegDB

SCITOKENS

DQSegDB is a time-interval database for storing gravitational-wave observatory metadata.

When is the data good?

When is the data bad?

“Segments” (GPS time intervals) are generated and retrieved by automated processes and by humans

Ryan P. Fisher, Gary Hemming, Marie-Anne Bizouard, DAB, Peter F. Couvares, Florent Robinet, Didier Verkindt (arXiv:2008.11316)

LIGO/Virgo DQSegDB

API is RESTful web interface, X509 certificate authentication

Authorized users are allowed to query segments (LIGO/Virgo collaboration members)

Second, smaller, group is allowed to insert and update segments

<table>
<tr><td colspan="2">https://hostname/dq/ifo/flag/version</td></tr>
<tr><td colspan="2">Custom WSGI Python script</td></tr>
<tr><td>mod_wsgi</td><td>ODBC</td></tr>
<tr><td>X509 certificate</td><td>MariaDB</td></tr>
<tr><td>Apache</td><td>Storage</td></tr>
</table>

LIGO/Virgo DQSegDB

Set up SciTokens server to allow users (or HTCondor managed processes) to obtain tokens


```
{
  "aud": "segments.ligo.org",
  "nbf": 1606583049,
  "scope": "read:/DQSegDB",
  "iss": "https://test.cilogon.org",
  "exp": 1606583954,
  "iat": 1606583054,
  "jti": "https://test.cilogon.org/oauth2/accessToken/31dbe174a55c2a600046e45d4d99d0f5/1606583054943"
}
```

SciTokens Server and DQSegDB server need to agree on audience and issuer:

```
########################
# SciTokens constants #
######################
scitokens_issuer = 'https://test.cilogon.org'
scitokens_audience = 'segments.ligo.org'
scitokens_cache_dir = '/var/cache/httpd'
```

LIGO/Virgo DQSegDB

Use SciTokens Python Library to replace check on X509 subject in WSGI DQSegDB server code

Pass HTTP auth headers to WSGI script in Apache config:

```
WSGIPassAuthorization On
```

Deserialize and validate token, then check scope.

LIGO/Virgo DQSegDB


```
auth_type, auth_payload = environ['HTTP_AUTHORIZATION'].split(' ')
token = scitokens.SciToken.deserialize(auth_payload, audience=self.constant.scitokens_audience)
```

Wrap with try/except to check for invaid audience and expired token

```
class SciTokensAuthorization():
 def __init__(self):
   self.admin = Admin.AdminHandle()
   self.constant = Constants.ConstantsHandle()
   os.environ['XDG_CACHE_HOME'] = self.constant.scitokens_cache_dir
   self.token_enforcer = scitokens.Enforcer(self.constant.scitokens_issuer, audience=self.constant.scitokens_audience)

 def check(self, token):
  if self.token_enforcer.test(token, "read", "/DQSegDB"): r = [200]
  else: raise UnauthorizedError
```

GWDataFind & GraceDB (Duncan Meacher)

GWDataFind

GWDataFind is a package consisting of a server (gwdatafind-server) and client (gw_data_find) that is used to find GW data file locations, discovered based on metadata such as the interferometer, GPS start and end times, and the data frame types via a simple command-line tool.

```
[duncan.meacher@ldas-grid ~]$ gw_data_find --server datafind.ligo.uwm.edu:443 -o H -t 'H1_HOFT_C00' -l  -s 1180000000 -e 1180013000
H H1_HOFT_C00 1179996160 4096 file://localhost/cvmfs/oasis.opensciencegrid.org/ligo/frames/O2/hoft/H1/H-H1_HOFT_C00-11799/H-H1_HOFT_C00-1179996160-4096.gwf
H H1_HOFT_C00 1180000256 4096 file://localhost/cvmfs/oasis.opensciencegrid.org/ligo/frames/O2/hoft/H1/H-H1_HOFT_C00-11800/H-H1_HOFT_C00-1180000256-4096.gwf
H H1_HOFT_C00 1180004352 4096 file://localhost/cvmfs/oasis.opensciencegrid.org/ligo/frames/O2/hoft/H1/H-H1_HOFT_C00-11800/H-H1_HOFT_C00-1180004352-4096.gwf
H H1_HOFT_C00 1180008448 4096 file://localhost/cvmfs/oasis.opensciencegrid.org/ligo/frames/O2/hoft/H1/H-H1_HOFT_C00-11800/H-H1_HOFT_C00-1180008448-4096.gwf
H H1_HOFT_C00 1180012544 4096 file://localhost/cvmfs/oasis.opensciencegrid.org/ligo/frames/O2/hoft/H1/H-H1_HOFT_C00-11800/H-H1_HOFT_C00-1180012544-4096.gwf
```

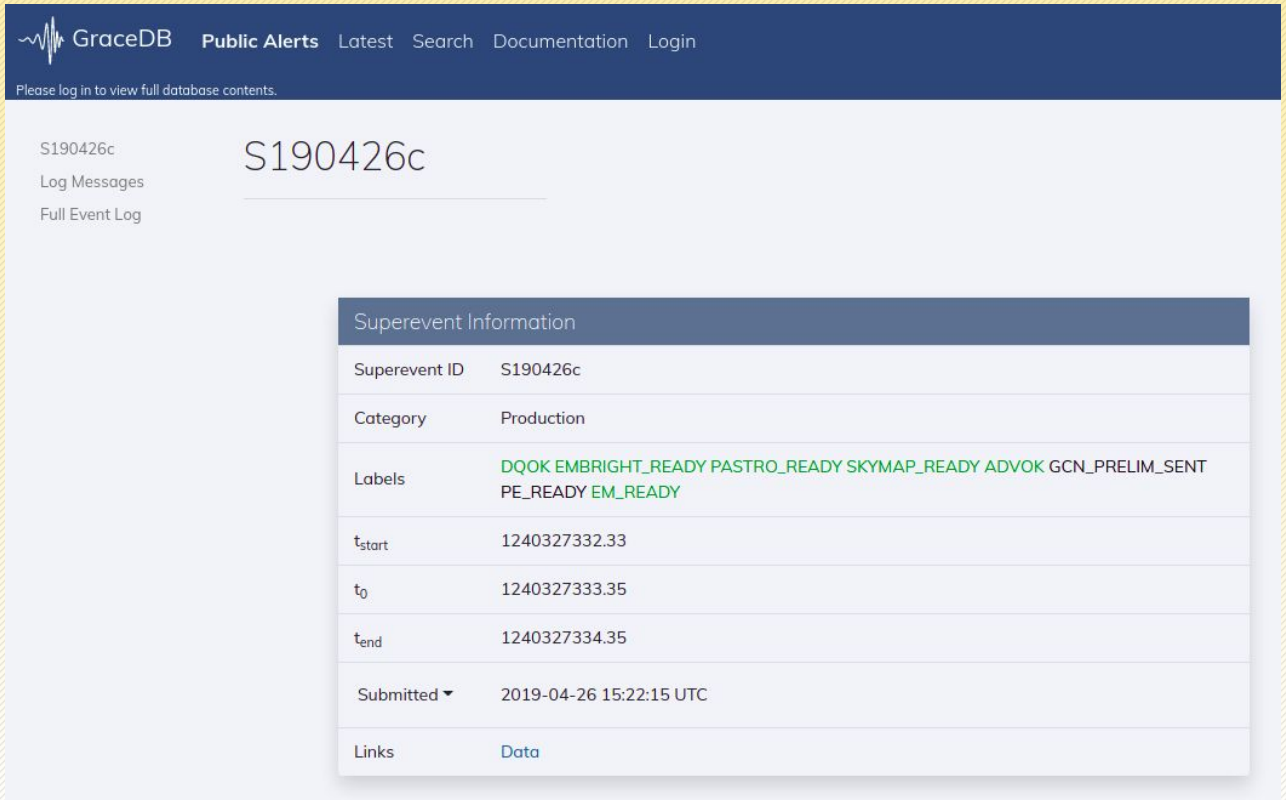
- Currently uses X.509 authentication
- Accessible only to LIGO+Virgo Collaboration members

GraceDB

SCITOKENS

The Gravitational-Wave Candidate Event Database has served as the repository for candidate events and associated data products since 2010.

- Currently uses X.509 authentication
- Accessible to everyone for public data, and LVK + observing partners for private data



GWDataFind and GraceDB SciToken Access

SciToken access to GWDataFind currently in development. Once this is complete, the changes will be applied to GraceDB. Using a three-stage plan:

1. Generate own SciToken and encode with SSH key, then decode with local public key and authenticate on the server. Done
2. Use cluster/Condor generated SciTokens, and encode/decode with local cluster SSH key/public key.
3. Set up server that contains all cluster public keys for decoding.

Discussion (moderated by Jim)

What interfaces (command-line, web, etc.) do LIGO scientists need for obtaining/using tokens?

Can you walk us through the use of tokens on a cluster head node (from login to job submission to data access)?

What are pros/cons of OAuth-based token issuance versus local token issuance for LIGO?

Is there a role for OAuth device flow?

StashCache federation support requires TLS encryption needed to be enabled to use bearer tokens.

Are there other cases where encryption will need to be enabled?

What is the plan for access by unattended processes (Robots)?

What is the status of pyCBC compatibility with SciTokens?

Any use cases for ID Tokens or Group-Based Tokens?

How about use cases for HTCondor's new IDTOKENS authentication method?

Are there other token use cases we haven't discussed (e.g., Rucio, LVAlert)?

What are the timelines for LIGO's migration from X.509 to tokens?

How are international collaborations (Virgo, Kagra) impacted by the migration to tokens?

Open Discussion

For more info:

https://scitokens.org/

https://groups.google.com/a/scitokens.org/g/ligo-discuss