Status of SciTokens AuthZ Plugin

Derek Weitzel
SciTokens

- SciTokens are special JSON Web Tokens (JWT, RFC 7519)
- JWT defines the structure and the cryptography
- SciTokens adds attributes and defines a “scope” language
SciTokens

• SciTokens has already been integrated with many services:
  
  • **XRootD**
  
  • **NGINX** (through a configuration)
  
  • Even a **Heroku Application** (Flask)
List of tasks

1. Modifications to CVMFS

2. Integration Tests - Mostly copy from secure CVMFS

3. SciTokens Plugin - Python plugin
Changes to CVMFS

• New type of Auth type, Bearer Tokens

• A bearer token is an opaque string sent to the web server

```plaintext
> GET /protected HTTP/1.1
> Host: demo.scitokens.org
> User-Agent: curl/7.52.1
> Accept: */*
> Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzY3AiOiJy…
```

• Doesn’t need to be a SciToken, though I will be writing the plugin specific to SciTokens
Differences from X.509

• Auth is handled outside of the transport layer!

• No need to setup special SSL settings to include the client proxy / cert

• Much smaller changes to CVMFS
Integration Tests

- The integration test needs to setup a HTTP web server and authenticate SciToken
- Will copy a lot from the secure-CVMFS, since it does most of this
- Timeline: 1-2 weeks
SciTokens Plugin

- SciTokens library is in Python. Already have an “always allow” for testing:
  https://github.com/scitokens/cvmfs-scitokens-helper

- We will chain auth plugins:
  
  1. Try SciTokens
  
  2. Then, try X.509
Plugin Chaining

• Difficult part: Maintaining a process, the x509 plugin
  • Do not want to restart the process often due to reading in globus libraries...
  • If SciTokens authz, simply proxy the connection to the x509 plugin
Summary

1. Modifications to CVMFS

2. Integration Tests - Mostly copy from secure CVMFS

3. SciTokens Plugin - Python plugin