Authentication and Authorization

Charles (Cal) Loomis & Mohammed Airaj
LAL, Univ. Paris-Sud, CNRS/IN2P3
24-25 October 2013

Authentication

For many reasons, users must be authenticated, but…

- Authentication systems between sites vary greatly depending on a site's technology choices, the target users for the cloud, etc.
- Sites may need to use several different user databases.

- E.g. small private cloud may use simple password file
- E.g. large public cloud may use LDAP or certificates

Requirements

- Flexibility: support for different systems with simultaneous use
- Integration with existing infrastructures
- Consistency across all cloud services

JAAS-based Authentication

JAAS : Java Authn and Authz Service

- Flexible system for Java servlet containers
- Separates authentication mechanisms from application

JAAS in StratusLab

- All user interactions take place via java-based services
- Unique set of configuration files for all services
- Multiple services implement authn consistently
- Configuration allows for flexible authn at runtime
- Multiple different methods can be used simultaneously

- Requires different endpoints for certificate and non-certificate methods

Authorization

Service-based Authorization

- User capabilities are determined by specific service
- Generally user has full access to her resources, no access to others
- Some flexibility with storage, but current access control deprecated

Username/Password Properties File

Simple list of users

- `/etc/stratuslab/auth/login-pswd.properties`
- Lists: username, password, groups
- Required group: ‘cloud-access’

```
# Entries look like the following:
#
# username=password,cloud-access
#
# 'cloud-access' is a required role

oneadmin=ONE48394,cloud-access
pdisk=KgpOTgeBC7Jr,cloud-access
test=test10348,cloud-access
```

Username/Password Certificates File

Simple list of certificate DNs

- `/etc/stratuslab/auth/login-cert.properties`
- Lists: DNs and groups
- Required group: ‘cloud-access’
- DNs must be in RFC2253 (not grid!) format

```
# First token on the line is the DN of the user.  This
# MUST be enclosed in double quotes.
#
# All following tokens are taken as groups.  These may
# be separated by whitespace and/or commas.
#
# "DN=John Smith, O=Widget Inc." cloud-access
#
"CN=Charles Loomis, OU=LAL, O=CNRS, C=FR, O=GRID-FR" cloud-access
```

Centralized Database

Global Authn Configuration

- `/etc/stratuslab/auth/login.conf`
- Activates authn mechanisms
- By default, previous mechanisms and LDAP

LDAP

- Username/password information for users
- Optional certificate (DN) information for users
- Works with both username/password and certificate authentication
- Warning: cert. and username/password are considered different users!

Global/LDAP Configuration

```
stratuslab-cert {
  eu.stratuslab.authn.CertLoginModule sufficient
  file="/etc/stratuslab/authn/login-cert.properties";

  eu.stratuslab.authn.LdapCertLoginModule sufficient
  debug="false"
  useLdaps="false"
  contextFactory="com.sun.jndi.ldap.LdapCtxFactory"
  hostname="onehost-5.lal.in2p3.fr"
  port="389"
  bindDn="cn=admin,o=cloud"
  bindPassword="xxxxxx"
  authenticationMethod="simple"
  userBaseDn="ou=users,o=cloud”
…
};
stratuslab-pswd {
…
};
```

Shibboleth

Challenging

- For technical reasons, shibboleth is challenging for java web apps.
- Usual impl. via Apache proxy is not very satisfying or flexible

But…

- Shibboleth federations are becoming more popular
- Have recurrent requests to support this
- eduGAIN will make support more important
  - http://www.geant.net/service/eduGAIN/Pages/home.aspx
- Sites and people willing to work on integration very welcome

Services Using JAAS

Computing: one-proxy

- Proxy service in front of OpenNebula
- Exposes XML-RPC interface of OpenNebula
- Authn information passed to and trusted by OpenNebula

Storage: pdisk

- Same authn methods integrated into separate service

Future Authn Framework

Friend

- Authn framework for “ring” applications written in Clojure
- Flexible support for large number of authn “workflows”
- Includes HTTP basic, certificates, form-based ID, …

Friend in StratusLab

- Friend will be the authn framework used in CIMI service
- CIMI will be the only interface to StratusLab cloud services
- Authn happens in application rather than in servlet container
- Configuration similar to JAAS but in Couchbase JSON documents
- OpenID and GitHub (OAuth2) supported easily

Exercises

1. What authentication methods will you use?

Questions and Discussion

website	http://stratuslab.eu
twitter	@StratusLab
support	support@stratuslab.eu
StratusLab source	http://github.com/StratusLab
SlipStream source	http://github.com/slipstream

http://stratuslab.eu/