An OAuth Service for Issuing Certificates to Science Gateways for TeraGrid Users

Jim Basney and Jeff Gaynor
{jbasney,gaynor}@illinois.edu

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

This material is based upon work supported by the National Science Foundation under grant number 0932251.
Goals

- Support use of *individual TeraGrid accounts* via gateways
  - Independent of support for gateway *community accounts*
  - For more accurate accounting, greater resource access
- Avoid disclosing TeraGrid user passwords to gateways
  - Avoid risk to long-lived credentials (i.e., user passwords)
  - Use TeraGrid passwords only on systems operated by TeraGrid
- Use standard security protocols: TLS, OAuth
  - More trustworthy
  - Ease of integration for gateway developers

http://security.ncsa.illinois.edu/teragrid-oauth/
Current Approach  →  New Approach

http://security.ncsa.illinois.edu/teragrid-oauth/
Benefits

• **Security WG** concerns about password disclosure to external science gateway sites are addressed

• **Science Gateways** can support individual TeraGrid account access via standard protocols

• **Resource Providers** can support user access via gateways using existing certificate-based interfaces

• **Users** can access their individual TeraGrid accounts via gateways using their TeraGrid Portal login

http://security.ncsa.illinois.edu/teragrid-oauth/
OAuth Example

1. Request Access to Photos
2. Authenticate & Grant Access to Photos
3. Token
4. Token
5. Token
6. Photos

Web User (Resource Owner)

Photo Printing Service (Client)

Photo Sharing Service (Server)
Current Approach  →  New Approach

http://security.ncsa.illinois.edu/teragrid-oauth/
User's browser \rightarrow OAuth client

initiate(certreq, consumer_key, callback, signature)

temp_token

OAuth server \rightarrow science gateway

authorize(temp_token)

authenticate and approve

MyProxy username and password given here

callback(temp_token, verifier)

token(consumer_key, temp_token, verifier, signature)

access_token

req(consumer_key, access_token, signature)

certificate

MyProxy server \rightarrow TGUP

get(username, password, certreq)

certificate
Protocol

• RFC 5849 – OAuth 1.0a
  • OAuth client: science gateway
  • OAuth server: TeraGrid User Portal
  • OAuth resource owner: TeraGrid user
• All connections use HTTPS for integrity + confidentiality
• OAuth client messages signed using RSA-SHA1
• PKCS#10 certificate request → PEM encoded certificate
  • Private key never sent over the network
• Future work: OAuth 2.0 (under IETF development)
Current Status

• Code complete
  • Java API: requestCertificate() and getCertificate() functions
• Acceptance testing with Globus Online in progress
• Next Step: Production User Portal deployment

• Code, Documentation, Specifications, etc. at:

  http://security.ncsa.illinois.edu/teragrid-oauth/
Design Decisions

• OAuth server independent from Liferay
• Store all server-side state in a replicated database
  • Leverage existing User Portal load balancing, fail-over, and replication mechanisms
• No changes to TG MyProxy servers
• Initially support only password-based authentication
  • Federated authentication (InCommon/Shibboleth) a possible future enhancement
• No initial support for certificate renewal
  • Certificates valid for up to 11 days
• Explicit user approval for every certificate issuance
• Initial support for web browser use cases only
Security Considerations

• Our paper addresses each security consideration identified in RFC 5849 (15 items)

• Summary:
  • HTTPS provides message integrity+confidentiality and server authentication, avoids HTTP proxy caching
  • RSA-SHA1 signature method:
    • If gateway private key is compromised, revocation is a server-side database operation
    • Only public key need be stored on server-side
    • Address SHA-1 weakness in move to OAuth 2.0
  • Requiring user authentication+approval for every certificate issuance addresses “clickjacking” and similar threats

http://security.ncsa.illinois.edu/teragrid-oauth/
Related Work

• OAuth use by Gateways
  • Open Protein Simulator (OOPS)
  • Open Life Science Gateway (OLSG)
  • Open Grid Computing Environments (OGCE)
  • Also future work for PolarGrid, QuakeSim, TG Viz Gateway

• OAuth for certificate access
  • Confusa (confusa.org) used by TERENA Certificate Service with European SAML federations
  • CILogon (cilogon.org) with US InCommon SAML federation

http://security.ncsa.illinois.edu/teragrid-oauth/
Possible Future Work

• OAuth 2.0 update
• General-purpose MyProxy OAuth package w/o TeraGrid dependencies
• Integrate existing TeraGrid federated authentication (InCommon/Shibboleth) with OAuth Sign In page
• Certificate renewal using OAuth refresh tokens
• Support for non-browser use cases (e.g., REST services)

http://security.ncsa.illinois.edu/teragrid-oauth/
Conclusion

• A new standards-based service to issue certificates to science gateways for TeraGrid users
  • Available now for testing
• Eliminates need for TeraGrid users to disclose TeraGrid passwords to science gateways when accessing individual accounts
• Independent of support for gateway community accounts

Questions? Comments?

Thanks!

http://security.ncsa.illinois.edu/teragrid-oauth/