Globus Security Deep Dive

GlobusWorld 2014

Steve Tuecke
Globus Federated Identity Authentication and Linking
User A chooses alternate IdP, IdP1, to log in.

Redirected to IdP1.

Linking a Federated Identity.
Linking a Federated Identity

User A Authenticates

OAuth interactions

User A's identity: userA@IdP1.org

Check for linked Globus account

Federated Identity Provider IdP1

Prompt for login
Linking a Federated Identity

User A Authenticates

Prompt for login with Globus account

globususerA@globus.org
userA@ldP1.org
Linking a Federated Identity

User A chooses alternate IdP, IdP1, to log in

User A logged in as
globususerA@globus.org
with linked identity

.globususerA@globus.org
userA@IdP1.org
Login with Federated Identity

User A is redirected & authenticates

User A logged in as

globususerA@globus.org

User A's identity:

userA@IdP1.org

User A logged in as

globususerA@globus.org

User A logs in using alternate IdP: IdP1
Using Campus IdP (via CILogon)

User A logs in using Campus IdP

User A is redirected to Campus IdP

User A is redirected to CILogon

CILogon

User A logs in using Campus IdP

→ globususerA@globus.org

→ userA@uchicago.edu
Using Campus IdP (via CILogon)

User A logs in using Campus IdP

Prompt for login

User A Authenticates

SAML interactions

User A's identity: userA@uchicago.edu (optional attributes)

CILogon

.globususerA@globus.org

userA@uchicago.edu
Using Campus IdP (via CILogon)

User A logs in using Campus IdP

User A logged in as globususerA@globus.org

User A's credentials with identity: userA@uchicago.edu
  (optional attributes)

- globususerA@globus.org
- userA@uchicago.edu

CILogon
Future Identity Directions

• Move to user@domain user names
  – Current Globus usernames become user@globus.org
  – Users not required to have @globus.org name

• Auto-provision accounts from other identity domains

• XSEDE identities will fold into this
Globus Endpoint Authentication
Globus Sharing Security
Globus Sharing

1. User A selects file(s) to share, selects user or group, and sets permissions.

2. Globus manages ACLS on shared files; no need to move files to cloud storage!

Configuring a Managed Endpoint for Sharing

Globus Connect Server trusts MyProxy OAuth

Data Source

Configurable policies:
- Enable share
- Share restricted path
- Read only or Read/Write
- Local users that can share

Admin configures Managed Endpoint /etc/globus-connect-server.conf
Activate Endpoint

User A authenticates using resource credentials

User A selects endpoint to activate

User A’s credentials returned to Globus

User A’s credentials

MyProxy OAuth

Data Source

Manage Shared Endpoint

Manage Permissions For IanFoster

Host: uwresearch:/~/Share/

<table>
<thead>
<tr>
<th>ID (User or Group)</th>
<th>read</th>
<th>write</th>
<th>delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ian Foster</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Globus Team</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steve Tackett</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Liska</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(view transfer path)
User A creates a shared endpoint, userA#share for path /projects/
User A creates a shared endpoint, userA#share for path /projects/
Create Share

User A creates a shared endpoint, userA\#share for path /projects/.

Does the endpoint allow shared endpoint for path /projects/?

Yes

User A’s credentials
User A creates a shared endpoint, userA#share for path /projects/
Set permissions

User A sets permissions for User B to read path userA#share:/dir1

- User A’s credentials
- userA#share, UUID, User A’s credentials, /projects/
- ACL: userA#share:/dir1, read, User B

For User A, file share.UUID with path /projects/
Check ACLs for User B

User B lists
userA#share:/dir1

For User A, file
share.UUID with
path /projects/

User A's credentials
userA#share, UUID, User A's credentials, /projects/
ACL: userA#share:/dir1, read, User B
User B lists userA#share:/dir1

For User A, file share.UUID with path /projects/

User A’s credentials
userA#share, UUID, User A’s credentials, /projects/
ACL: userA#share:/dir1, read, User B
User B lists userA#share:/dir1

Request sharing from UUID with User A's credentials

Get User A's local account

Data Source

For User A, file share.UUID with path /projects/

- User A's credentials
- userA#share, UUID, User A's credentials, /projects/
- ACL: userA#share:/dir1, read, User B
User B lists userA#share:/dir1

Request sharing from UUID with User A's credentials

- User A's credentials
- userA#share, UUID, User A's credentials, /projects/
- ACL: userA#share:/dir1, read, User B

Check for User A's share.UUID file

For User A, file share.UUID with path /projects/
User B lists userA#share:/dir1

Request sharing from UUID with User A’s credentials

For User A, file share.UUID with path /projects/

User A’s credentials
userA#share, UUID, User A’s credentials, /projects/
ACL: userA#share:/dir1, read, User B
User B lists userA#share:/dir1

Request sharing from UUID with User A’s credentials

Data Source

Setuid to User A local account, change root to path

For User A, file share.UUID with path /projects/

User A’s credentials
userA#share, UUID, User A’s credentials, /projects/
ACL: userA#share:/dir1, read, User B
User B lists 'userA#share:/dir1'

User A’s credentials
'userA#share, UUID, User A’s credentials, /projects/
ACL: 'userA#share:/dir1, read, User B'

Combine user path restrictions, & sharing restricted path

For User A, file share.UUID with path /projects/
User B lists userA#share:/dir1

User A’s credentials
userA#share, UUID, User A’s credentials, /projects/
ACL: userA#share:/dir1, read, User B

For User A, file share.UUID with path /projects/
Transfer from share

User B transfer from userA#share:/dir1

Enforce share permissions

For User A, file share.UUID with path /projects/

User A’s credentials
userA#share, UUID, User A’s credentials, /projects/
ACL: userA#share:/dir1, read, User B
For User A, file share.UUID with path /projects/

Data Source

Globus moves the files

Data Destination

User B transfer from userA#share:/dir1

User A’s credentials
userA#share, UUID, User A’s credentials, /projects/
ACL: userA#share:/dir1, read, User B
Future Possible Sharing Directions

• Share to users by email address

• Level of Assurance policies
  – E.g., In order to write to a shared endpoint, user must have authenticated with one of these IdPs, in the last N minutes, within this browser session.

• Time-based shared endpoints

• Time-based ACLs

• Periodic re-validation of shared endpoints
Operational Security
Operational Security

• **Separate AWS security groups for:**
  – Nexus vs Transfer, Production vs Test and QA
  – Check ports hourly

• **Central logging with Nagios monitoring**

• **OSSEC intrusion detection**

• **Globus root CA for sharing access on offline hardware security module**
Who has access to what?

- Access to production backends restricted to only those ops staff who need it to operate the service.
- Globus Connect Server endpoint restrictions prevent Globus ops access.
What data does Globus see?

• User profile: email, name, etc.
• Linked identities: no secrets stored
  – With OAuth, we never see passwords
• Temporary user credentials
• File paths, but NOT file contents
• File level transfer logs retained for 1 month
• Summary level transfer history retained indefinitely
• Publication metadata