

Globus Online File Transfer Service Security Overview

Draft last modified on 07 October 2011

Home	Contributors	Documents	Development	Operations	Support	Contact Us
----------------------	------------------------------	---------------------------	-----------------------------	----------------------------	-------------------------	----------------------------

- [Service Overview](#)
- [Security Contacts](#)
- [User Assets](#)
- [Infrastructure Assets](#)
 - [Postgres Database](#)
 - [EC2 AMI Images](#)
 - [Live EC2 Instances](#)
 - [Grid Endpoints \(External\)](#)
 - [Globus Connect Endpoints](#)
 - [Source Code Repository](#)
 - [Software](#)
- [Architectural Overview](#)
 - [Actors and Privilege Levels](#)
 - [Restricted CLI User](#)
 - [GO Application User](#)
 - [System Administrator](#)
 - [REST User](#)
 - [REST Admin User](#)
 - [Architectural Components](#)
 - [REST Interface](#)
 - [CLI Interface](#)
 - [Restricted Shell](#)
 - [Activation](#)
 - [Back-end Processing Components](#)
 - [Starter Component](#)
 - [Worker Components](#)
 - [Administrative Components](#)
 - [Relay Component](#)
 - [Database Component](#)
 - [Grid Security and Trusted Certificate Authorities](#)
 - [Firewall and Security Groups](#)
- [System Availability](#)
- [Monitoring](#)
- [Logging, Auditing, and Intrusion detection](#)
- [Future Improvements](#)

Service Overview

This service transfers files between GridFTP endpoints on a user's behalf. The file transfer servers run on Amazon's EC2, and users connect using gsissh or ssh and execute commands in a restricted shell; this is called the CLI interface. A REST Transfer API is also available, allowing transfers to be initiated and progress to be monitored by sending HTTPS requests. The www.globusonline.org website includes graphical interfaces for monitoring and transfer submission which use this REST API. (Globus Connect)

Users authenticate to the service using ssh public key authentication, gsissh x509 authentication, or x509 SSL client authentication. The service obtains credentials for accessing GridFTP endpoints via gsissh credential delegation or by running myproxy logon at the user's request. The service stores the temporary delegated credential(s). The myproxy username and passphrase are not stored.

Security Contacts

Main Contact: support@globusonline.org
Emergency Contact: TBD

User Assets

The following user data is collected by the service. Data is stored in the database unless otherwise specified.

1. ssh public key - used for authentication for cli.globusonline.org. Stored in the database and on the instance filesystem.
2. x509 DN - used to setup gridmap for authentication to cli.globusonline.org with gsissh. Stored in the database and on the instance

filesystem.

3. MyProxy hostname, username, and password - entered by the user so the service can fetch a credential and perform actions on the user's behalf. The MyProxy username and password are not stored, and are only sent over the network protected by ssh or https. Myproxy hostnames are saved if a user associates a default MyProxy with one of their gridftp endpoints.
4. x509 credentials - obtained when the users enters their myproxy login information, or via gsissh x509 delegation. Stored in the database, and used to establish gridftp connections to external servers. Credentials have a limited lifetime.
5. transfer activity - the complete history of a user's transfer requests is stored in the database.
6. endpoint file paths - the transfer activity includes file paths, which may include endpoint usernames if the home directory is specified explicitly in the path.
7. endpoint data - users may create a shortcut for frequently used transfer endpoints, which includes the URI to the remote server and optionally a default myproxy server. The user may optionally share this shortcut with other users.
8. public endpoint usage - when users make use of public endpoints, it leaves a trail in the transfer history.
9. email address - used to send users updates about transfer progress and send administrative information.
10. external user assets - the user credentials delegated to the service give it access to any asset protected by that credential, for the lifetime of the credential. This can include files stored on gridftp endpoints, and potentially other non file resources. In addition, GO will have access to files on connected Globus Connect endpoints (usually a user's personal laptop or desktop computer). The service only uses these credentials to manipulate files according to user requests.

Infrastructure Assets

Postgres Database

A postgres database contains user and request metadata, and is stored on an EBS volume. Application access to the database machine is restricted to other specifically configured production instances running on the same Amazon EC2 account, using EC2 Security Groups for firewalling. Administrative access is allowed via secure shell to a small set of administrators.

Database snapshots are stored in Amazon S3 storage. Access is restricted to small set of administrators.

EC2 AMI Images

AMI images containing a working production system are stored on Amazon S3. Access to the images is limited to GO development and operations personnel.

Live EC2 Instances

Several server instances on EC2 are used to provide the service:

1. cli.globusonline.org - front facing server exposed to users, who log in to a restricted shell to run transfer commands.
2. transfer.api.globusonline.org - front facing server exposed to users for interacting with the service over HTTPS with the Transfer API.
3. relay.globusonline.org - relay server allows Globus Connect clients to reliably and securely exchange GridFTP control channel data with the GO service.
4. database and backend servers - depending on load requirements, these may run on cli.globusonline.org, or as separate instances.
5. tutorial endpoints - two gridftp servers are exposed for access by all GO users; all users of the service are given limited space on these endpoints to experiment and learn how to use the service.
6. email server - used for sending email to users with updates about transfer progress.

Administrative access to production EC2 instances is limited to a small set of core developers and system administrators.

Grid Endpoints (External)

The service interacts with external grid resources at many different locations. This includes GridFTP servers and MyProxy servers.

Globus Connect Endpoints

Globus Connect Endpoints run on individuals desktop and laptop computers, allowing transfer of files between these computers and external GridFTP servers. These endpoints connect to the relay service using a secure GSISSH tunnel, allowing GO to issue control requests in a similar manner as it does to normal Grid Endpoints.

Source Code Repository

Source code and deployment data, including host keys, is stored on the Computation Institute (CI) Subversion server, in a project-specific repository. It is accessed via HTTPS and requires a username and password. Only contributors are allowed access; however the username and password are shared with other CI services. Backups are maintained by the CI system administrators.

Commits are closely monitored by the development team.

Software

The servers run Ubuntu 10.04 (Lucid), and make use of the following software packages:

1. Python 2.6
2. PostgreSQL 8.4
3. Apache 2.2, mod_wsgi
4. Globus Toolkit 5.0 (gsissh, myproxy-logon)
openssh

Non-application software updates occur on a periodic basis along with the normal release process. The Globus Online development and operations staff monitor the security update mail lists. Any critical updates are expedited.

Architectural Overview

Actors and Privilege Levels

Restricted CLI User

A user account is created for each registered globusonline user. CLI users are authenticated either via SSH public keys or GSI authentication. All such users belong to a common group. Members of this group are mapped to a restricted shell upon login. (more information below on the restricted shell).

Weakness: Processes running at this level have database access, currently to the entire database. This is mitigated by the restricted shell.

GO Application User

A user account exists for the globusonline system itself. GO services run at this privilege level. Most GO services act in the context of a single user. Some GO services look across users.

This account has full password-less sudo access. The code is also owned by this user. Note that this account does not have login privileges - you have to su to it.

Weakness: This account has full sudo access. This is probably more of a development / admin convenience and could be removed with some testing. It looks like the user provisioning is using sudo to root capability to run install-users.

System Administrator

This level of privilege has full access to the system and all assets managed by the system. Access is limited to a small set of system administrators and core developers. Access is via SSH public keys.

REST User

REST users are authenticated with either SSL x509 client authentication or signed cookie authentication. The x509 client authentication uses the same credentials that provide CLI access via gsissh and is intended for scripted implementations. Signed cookie authentication is used for a single sign-on experience via the www.globusonline.org web interface. All rest processing uses a non-privileged user account that does not have write access to the source code. It accesses the database via a single database account. At this privilege level, a user identifier is always included in any database queries preventing unintended access to other user data.

REST Admin User

A small number of users are able to use the globusonline monitoring web interface to monitor transfers of other users. These users are denoted through a permissions attribute stored in the database which is manually manipulated by a globusonline administrator.

Architectural Components

The file transfer service can be broken down into five main components, a REST interface, a CLI interface, back-end worker tasks, a relay server, and a database server. The system is designed such that these components can coexist on a single server or each be run on a dedicated server. Access to TCP ports used for internal communication are firewalled with Amazon EC2 security groups. In general, only requests originating from servers running within the same or other configured security groups are allowed access to private ports such as the database access port. (I don't like this last sentence. Too vague, need to pin it down to specifics in some way)

REST Interface

The Transfer API runs under mod_wsgi in Apache, using a non-privileged user account that does not have write access to the source code. Actors are either the REST User or REST Admin User depending upon permissions configured in the database. The processing code interacts with the database at the privilege level described above.

CLI Interface

The command line interface can be accessed via standard ssh or gsissh. When an account is created, the user has the option of providing their ssh public key as well as their x509 public key and / or subject. The 'Restricted CLI' actor is used here.

Restricted Shell

All CLI commands and user interaction are performed within a restricted shell which is implemented in Python. All transfer users are mapped to a single UNIX primary group to simplify SSH configuration. The shell and command set run under the user,Ãs UNIX account UID.

The CLI command set interacts with a Postgresql database using a single, shared database account. The CLI command programs ensure that only data belonging to the user can be viewed or changed. There are two general categories of commands: status or query commands and action or request commands. The former retrieves and formats information stored in the database. The actions or request commands enter information into the database. This information can then be acted upon by the back-end processing components of the system.

Activation

In order to carry out file transfer activity on behalf of a user, credentials are needed which allow access to user assets such as files on endpoint resources. Proxy credentials are obtained through a process called activation. These credentials are stored in the Postgresql database and are accessed by back-end processes running on the user's behalf. (More about storage policy of these credentials). Activation can occur either via gsissh x509 delegation or through a MyProxy logon process. In the case of gsissh, no private user information is made available to globus online other than the proxy itself. In the case of obtaining a proxy credential from a MyProxy server, the user supplies the server, user name, and pass phrase. The service then obtains the proxy certificate via myproxy-logon. The user name and pass phrase are not stored by the service. (Add OAuth MyProxy activation here, including eventual CLI integration) When a user requests a transfer involving their Globus Connect endpoint, GO generates a short-lived credential on the user's behalf. Mutual authentication between the user and GO has already occurred at this point so GO is able to generate a credential without further interaction with the user.

Back-end Processing Components

There is a collection of back-end processing components (sometimes called workers) which carry out the requests entered by the user via the CLI or REST interfaces. The 'GO Application' actor is used here.

Starter Component

The starter component monitors the database for requests to carry out on behalf of all users and enforces configurable user and system-wide limits of activity. When new work is identified that fits within the configured limits, the starter will run a worker component to carry out the work on behalf of the user.

Worker Components

Various worker components exist to carry out portions of the file transfer request. These components run on behalf of the user and use the activated credentials to access user resources on endpoint servers.

Administrative Components

Another class of back-end components are those that perform general administrative tasks and house cleaning. An example of this is a component that finds and cancels expired requests. Some of these components also send email to users to inform them of important events.

Relay Component

The relay component acts as a secure conduit for Globus Connect(GC) endpoints. When requested by the user, a Globus Connect endpoint establishes a gsissh tunnel between relay.globusonline.org port 2223 and localhost 2811. GridFTP control channel traffic can then flow over this tunnel. This allows GC to operate behind a firewall or NAT that allows only outbound connections. The GridFTP server deployed as part of Globus Connect listens on port 2811 for requests coming from GO. Only GO can connect to the tunnel on the server side.

Database Component

At the heart of the system lies a Postgresql database. This database stores user and request meta-data. User meta-data includes user name, email address, ssh public keys, x509 subjects and a preferred set of endpoints. In addition, when endpoints are activated, user proxy credentials are stored. Request data includes file or directory name(s), date and time of request, total size of file(s), bytes transferred, and options such as recursive and sync transfer.

Weakness: The 'GO Application' actor, 'Restricted CLI' actor and REST actors have full access to the postgresql database.

Grid Security and Trusted Certificate Authorities

The globus online file transfer service is configured to trust a standard set of certificate authorities including those provided by IGTF, TeraGrid, European Grid, etc. In addition, individual CAs are trusted on a case-by-case basis with a manual approval process which includes review of the signing policy associated with the CA.

These authorities are used to verify x509 certificates used for gsissh authentication to the CLI and x509 authentication to the Transfer API. They are also used for mutual authentication when communicating with external MyProxy and GridFTP servers.

In addition, Globus Online operates a CA which is trusted by all Globus Connect endpoints as well as the Globus Online tutorial endpoints. Resources external to GO are not required to trust this CA.

Firewall and Security Groups

TODO: the internal firewall (same account only) is mentioned above; this should probably talk about what ports are open on the public firewall.

System Availability

All system components are hosted in Amazon EC2, a highly available cloud infrastructure. Key system assets including database files and system images are stored in a combination of Amazon S3 and Amazon EC2 EBS (Elastic Block Store). Database snapshots are performed every hour and saved to Amazon S3. Note that occasionally there will be scheduled downtime, for example when component software needs to be upgraded.

The main file transfer components are currently not configured in a highly available fashion. The system is designed and managed in a way that a component can be recovered using a manual process in a short period of time. System images live on Amazon S3 storage - a highly-redundant, highly-available storage system. Customized globus-online images are maintained which allow for quick initialization and configuration of failed components. Snapshots of the central postgresql database are taken on an hourly basis. These snapshots are stored in Amazon S3 as well. In the event of a volume failure, the most recent snapshot can be used to quickly create a new database volume. This volume can then be mounted on an existing or new EC2 instance.

Monitoring

Nagios and Amazon CloudWatch are used for system monitoring. Nagios sends alerts via email and SMS to the GO operations and development staff. Second level support is provided by the Users Services team (MCS ANL / CI Uchicago) and third level support is provided by the core development team.

Logging, Auditing, and Intrusion detection

ssh sign in attempts, both successful and failed, are logged according to the default ssh logging facilities. These logs are not currently audited by any automatic analysis tools, and there are no policies in place for regular manual auditing.

Future Improvements

We have researched horizontally scalable request and worker components as well as HA replication of the database. This will provide more scalability and availability.