APS Components Administrator Document

1. MyProxy Online CA

MyProxy is an online credential repository. You can store X.509 *proxy credentials* in the MyProxy repository, protected by a passphrase, for later retrieval over the network. This eliminates the need for manually copying *private key* and certificate files between machines

Administrator Guide

Starting with v3.0, the MyProxy server includes the ability to act as a Certificate Authority (CA), signing certificates with a configured CA key on request for authenticated users that don't already have certificates stored in the MyProxy repository. Users can run `myproxy-logon` to authenticate and obtain a certificate from the MyProxy CA when and where needed, without needing to store long-lived keys and certificates in the MyProxy repository or elsewhere.

The MyProxy CA has been developed to meet the requirements of the Short Lived Credential Services X.509 Public Key Certification Authorities Profile of The Americas Grid Policy Management Authority, a member of the International Grid Trust Federation. The NCSA MyProxy CA has been accredited under the Profile.

The MyProxy CA functionality requires the configuration of PAM and/or SASL to support username/password and/or Kerberos authentication for obtaining certificates. The myproxy-server administrator must also configure the myproxy-server with the CA key, other CA attributes, and a method for mapping MyProxy usernames to Distinguished Names (DNs) in the signed certificates. Three mapping methods are supported: `certificate_mapfile`, which follows the Globus Toolkit grid-mapfile format, `certificate_mapapp`, which provides a general-purpose call-out interface, and LDAP.

The MyProxy server can act as a CA, a repository, or both. The CA functionality is disabled unless the CA configuration options in the myproxy-server.config are enabled. Likewise, the myproxy-server.config `accepted_credentials` setting controls whether clients can store credentials on the server. A myproxy-server.config template is provided in `$GLOBUS_LOCATION/share/myproxy`.

By default, the MyProxy CA issues certificates valid for 12 hours. This can be customized using the `max_cert_lifetime` option in myproxy-server.config.

Firewall Requirements

MyProxy is part of the Globus Toolkit as of the GT4 release. Prior to that release it was distributed separately.

MyProxy is a credential storage service for X.509 credentials. MyProxy connections are authenticated and secured with GSI and are normally from ephemeral ports on the client to 7512/tcp on the server.

MyProxy servers are not normally run on each computational resource alongside GRAM or GridFTP services, but instead instantiated once for an organization or project.

Security Considerations

You should choose a well-protected host to run the myproxy-server on. Consult with security-aware personnel at your site. You want a host that is secured to the level of a Kerberos KDC, that has limited user access, runs limited services, and is well monitored and maintained in terms of security patches.

For a typical myproxy-server installation, the host on which the myproxy-server is running must have /etc/grid-security created and a *host certificate* installed. In this case, the myproxy-server will run as root so it can access the host certificate and key.

2. gx-map

gx-map is a set of Perl programs designed to automate the maintenance of files in the /etc/grid-security directory, including the grid-mapfile, and trust root configuration. (CA certificates, signing_policy files, and CRLs)

Administrator Guide

This is gx-map, a system for automated updates of Globus grid-mapfiles and CA files. The current release is 0.5.3.3. The primary documentation is in the man pages (new in release 0.5.1) and on the gx-map web site, <http://users.sdsc.edu/~kst/gx-map/>.

The "gx-request" command is used to request the addition or deletion of an entry in the Globus grid-mapfile. In the simplest case, users can run "gx-request -quick-add"; this will generate a request to add a mapping for the DN extracted from the user's default proxy or user certificate, asking for verification before submitting the request. "gx-request -quick-remove" generates a removal request. For more complex actions, users can run "gx-request -interactive" and follow the instructions, or use other command-line options. Run "gx-request -help" or see the gx-request(1) man page for details. (Prior to release 0.4.5, the "gx-request" command was called "gx-map"; it can still be invoked by the old name, but the "gx-map" command will be removed in a future release.) A user may map an arbitrary DN (distinguished name) to his own Unix account. An administrator (as specified in the config file) may map any DN to any Unix account.

Firewall Requirements

None, it is a software on the local machine.

Security Considerations

The **gx-map** system is designed to maintain the grid-mapfile and certificate files used by Globus. These files are critical to the security of any system providing Globus services, and any errors in these files (deliberate or otherwise) can open security holes, either allowing a user to access another user's account, or allowing an outside attacker to gain access to the system.

A single Unix account must be chosen to install, own, and run the **gx-map** system. If possible, this should be an account dedicated to the purpose (such as ``gxmap''). Access to this account must be carefully secured.

By installing, deploying, and using this software, you are trusting its author with the security of your system. (You can limit this trust by using ``paranoid mode''; see below.) I promise that I have done my best to make this software as secure as possible. I do not

promise that I have succeeded. See the *LICENSE* file for the legalese. (This isn't anything unusual; the same is true of most security-related software.)

Historically, in the past several years that various versions of gx-map have been used at SDSC and other sites, there has been only one exploitable security hole that I'm aware of. This hole only occurred in rare circumstances, was exploited exactly once (in my own testing with the permission of the target user), and was corrected in the following release.

3. GridFTP

GridFTP is a high speed data transfer service designed on top of and full backward compatible to the FTP protocol. It is generally used to transfer files off a standard file system from one location to another but it can be used as a frontend to specialized file systems as well. The protocol is open and published and a variety of tools and APIs exist in several languages which interact with it.

Administrator Guide

Once the proper certificates are in place administering a GridFTP server is fairly simple.

Information on security configurations can be found here: http://www.globus.org/toolkit/docs/latest-stable/admin/install/#gtadmin-basic-host

GridFTP can be run without any PKI security in either anonymous mode or by using a password file. Information on both can be found here: http://www.globus.org/toolkit/docs/latest-stable/data/gridftp/admin/#gridftp-admin-config-security

We also provide a reference client called globus-url-copy. It is a simple command line client that allows you to transfer from one URL to another. Information on how to use it can be found here: http://www.globus.org/toolkit/docs/latest-stable/data/gridftp/admin/#id2597598

Configuration of GridFTP servers can be done with inetd, xinetd, GFork, or running it as its own daemon. For simple cases we recommend running it via xinetd. Instructions for this are available here: http://www.globus.org/toolkit/docs/latest-stable/data/gridftp/admin/#gridftp-admin-inetd

Information on how and why to run gridftp under GFork is available here: http://www.globus.org/toolkit/docs/latest-stable/data/gridftp/admin/#gridftp-admin-gfork-master

Firewalls

GridFTP is a two channel protocol. There is a control channel for messaging and a data channel for data streaming. The data channel is brought up separately from the control channel with each transfer session. Therefor it is not statically configured on a port and further it cannot be associated with a single port. The advanced and faster data channel protocols in GridFTP require the receiver of the file to be the passive TCP listener, ie: the send connects, the receiver accepts. This means that that an open range of ports must be available on which the receiver can listen. To allow for this the administrator needs to open up a contiguous range of ports that the gridftp server can use for establishing data channels and then set the environment variable GLOBUS_TCP_PORT_RANGE=min,max such that

mix,max encapsulate that range. More information on this is available here: http://www.globus.org/toolkit/docs/latest-stable/data/gridftp/admin/#gridftp-security-firewalls

4. RFT

RFT is a transfer service that interacts with GridFTP. While GridFTP puts forth a best effort to securely and reliably transfer a data
RFT adds some additional guarantees. GridFTP is an on demand service which means that if you are allowed access to it, you can immediately transfer data. The down side of this however, is that you may not be immediately allowed access to the service. Further, if gridftp runs into a problem while transferring data it check points the transfer and ends it leaving it up to the client to restart it from the given checkpoint. While this is a useful component in a data transfer architecture and provides for a fast service, it places the burden of monitoring GridFTP services for open connections and successful authentications, as well as retrying transfers under failure, on the client. RFT addresses this situation.

RFT allows a user to request a transfer be performed on their behalf. The user can then disconnect from RFT and wait for a notification telling them that the transfer has completed. RFT takes on the burden of waiting to obtain access to GridFTP servers as well as retrying the transfers from check points in the event of failure. RFT also maintains state in a database so even if the RFT service itself fails, it will continue performing its requested transfer as soon as it is restarted.

Administrator Guide

As of the GT4.2 release RFT has become much easier to administer. It now uses an internal database easing the burden of database administration. More information of RFT configuration can be found here: http://www.globus.org/toolkit/docs/latest-stable/data/rft/admin/#rftAdmin