Globus Toolkit Firewall Requirements

Version 6

3/19/05

Von Welch, NCSA/U. of Illinois
vwelch@ncsa.uiuc.edu

Abstract

This document provides requirements and guidance to firewall administrators at sites that are deploying the Globus Toolkit clients and/or services. It describes the network traffic generated by using the Globus Toolkit, both in terms of what ports are used by what clients and services and the nature of the traffic in terms of authentication of connections and message protection of the data.

1 Introduction

The Globus Toolkit® [3][7] is a collection of clients and services to enable Grid Computing [1]. The toolkit includes components that enable users to securely initiate and manage compute jobs, move data onto and off of resources, and query resources to determine their availability and suitability for their use. Each of these components has it's own network traffic characteristics that must be considered when deploying the toolkit at a site with a firewall.

This document summarizes the toolkit's traffic requirements and also describes the nature of the traffic. A brief description of the toolkit's security mechanism is also given to explain how the network connections are secured. This document defines some specific ports that must be allowed by server sites in order to allow clients to access specific services. It also describes a site-definable ephemeral port range that must be open at both client sites, to allow callbacks from Grid services to clients, and at server sites in order to allow connection to transient Grid Services.

This document divides sites into two categories: client sites, which have users that are acting as clients to Grid services, and server sites, which are running Grid services. Server sites also often act as client sites either because they also have users on site or jobs submitted by users to the site act as clients to other sites by retrieving data from other sites or spawning sub-jobs.

Section 2 defines terminology used in this document and also provides an overview of the different versions of the Globus Toolkit. Section 3 gives a description of the Globus Toolkit services and the network traffic they generate. Section 4 describes how the Globus Toolkit's use of the ephemeral port range can be controlled. Sections 5 and 6 give firewall requirements for client and server sites respectively. Section 7 discusses our work to improve the Globus Toolkit's interoperability with firewalls and Section 8 contains references.

The latest version of this document can be found at [5].

2 Globus Toolkit Overview and Terminology

2.1 Versions of the Globus Toolkit and Underlying Technology

Services in the Globus toolkit can be characterized by their underlying technology and by the version of the toolkit in which they appear. There have been four versions of the Globus Toolkit: 1, 2, 3, and 4. As an abbreviated notation we refer to a particular version as *GTx*, where *x* is the major version number – e.g. GT2 refers to version 2 of the Globus Toolkit. The abbreviation *GT* without numeric qualifier is used to refer to the Globus Toolkit generically.

Each major version of GT has one or more minor versions. For the purposes of this document these minor versions are ignored since their network traffic characteristics do not differ.

Version 1 of the Globus Toolkit used a very different security mechanism. It is deprecated and is not discussed in this document.

Services in the Globus Toolkit are based on two distinct forms of technology, which are referred to as pre-web services (pre-ws) and web services-based. Pre-WS GT services are modeled after typical Unix services and are started by inetd. GT2 was composed entirely of pre-ws components. These components continue to exist in subsequent versions of GT alongside web services components.

GT3 and GT4 contain both pre-ws and web services based components. In GT3 the web services components are based on the OGSI [18] specification. In GT4 the web services components are based on the WSRF [21] specifications. In terms of network traffic, these web services components are very similar and this document describes them together, with any differences called out. The web-services components used hosting environments to encapsulate multiple services, giving them a common network endpoint.

2.2 Terminology

The following terms are used in this document:

Ephemeral Port: A non-deterministic port assigned by the system in the untrusted port range (>1024).

Controllable Ephemeral Port: An ephemeral port selected by the Globus Toolkit libraries that is constrained to a given port range as described in Section 4.

Grid Service Ports: Static ports for well-known Grid services described in Section 3.

GSI-Protected Connection: A network connection authenticated with the Globus Toolkit's Grid Security Infrastructure as described in Section 3.1. These connections are integrity protected and will also be encrypted if so requested by the controlling application.

GT2: Globus Toolkit version 2.x. This release contains only services based on pre-web services technology.

GT3: Globus Toolkit version 3.x. This release contains both pre-web services technology and web services-based technology based on OGSI.

GT4: Globus Toolkit version 4.x. This release contains both pre-web services technology and web services-based technology based on WSRF.

Hosting Environment: In web services-based versions of the Globus Toolkit, the hosting environment is a piece of software encapsulating a number of different services and acting as a single network endpoint for those services.

IANA: The Internet Assigned Numbers Authority [9].

OGSI: The Open Grid Services Infrastructure [18]. Some services in GT3 are based off of OGSI.

Pre-WS Services: Globus services predating the adoption of web services. I.e. the services present in GT2, but also found in GT3 and GT4.

Well-known Port: A port number registered with IANA [9].

WSRF: The Web Services Resource Framework [21]. Some services in GT4 are based off of WSRF.

3 Description of Globus Toolkit Traffic

This section gives a description of the Globus Toolkit services and the network traffic generated by their use. The Globus Toolkit's Grid Security Infrastructure (GSI) is briefly described, which is used to secure the network traffic to and from Globus Toolkit services. The traffic for the Globus Toolkit services of Job Management (GRAM), Data Movement (GridFTP) and Monitoring and Discovery (MDS) are briefly described. GSI-enabled OpenSSH and MyProxy, which are often used in conjunction with the Globus Toolkit, are also described. A table summarizing network traffic for all versions of the Globus Toolkit concludes the section.

3.1 Globus Authentication and Message Protection (GSI)

The Globus Toolkit uses the Grid Security Infrastructure (GSI) to provide authentication and message protection for its traffic. The features of GSI are described here briefly. Readers interested in more information on GSI should consult the descriptions of GSI in [15], [12], [19] and [20].

GSI uses the SSL protocol [1] along with X.509 credentials [8] issued to all users and resources to implement the Globus Toolkit's security services and to protect Globus Toolkit network traffic. Specifically GSI provides the following to GSI-Protected connections:

- **Secure Mutual Authentication:** Using the X.509 credentials and the SSL protocol, clients and servers authenticate to each other. This authentication involves a challenge-response protocol and no secrets (e.g. passwords, private keys) pass over the network. Normal SSL mechanisms are in place to prevent replay attacks. This authentication allows for authorization checks to be made by both parties.
- **Integrity Protection:** GSI-protected connections are always integrity protected to prevent tampering with data in-flight and hijacking of the connection by malicious parties.
- **Encryption:** GSI-protected connections are encrypted to provide privacy, if so requested by the application initiating the connection.
- **Secure Delegation:** Users may delegate short-lived credentials, called Proxy Certificates [17], to remote processes. This allows these processes to securely act on the user's behalf for the lifetime of the credentials (normally measured in hours). Proxy Certificates are similar in form and functions to Kerberos 5 Tickets [10].

The services described in the following subsections use GSI to protect their network connections as indicated.

3.2 Job Initiation and Management (GRAM)

The GRAM (Grid Resource Acquisition and Management) protocol is different in pre-WS and web services (OGSI, and WSRF) versions of GT and these are discussed separately here.

All the connections associated with GRAM are authenticated and integrity-protected with GSI as described in Section 3.1.

3.2.1 Pre-WS GRAM Protocol (found in GT2, GT3, GT4)

The Pre-WS GRAM protocol has two components:

1. Clients first connect to the Globus Gatekeeper to initiate a Job Manager (JM) and user job.
2. Communications then take place from the client and the JM to manage the job, stage data and/or executables, and return job output to the client. These connections are initiated in both directions.

This results in three traffic components that should be allowed by firewall policy:

1. A connection from a *controllable ephemeral port* on the client to the well-known static port of 2119/tcp to initiate a Job Manager (JM).
2. The JM listens on a *controllable ephemeral port* for management requests from the client (which come from ephemeral ports).
3. The JM may connect back from a *controllable ephemeral port* to a *controllable ephemeral port* on the client to response to management request, return job output, or stage input data or executables as requested by the client.

3.2.2 WSRF and OGSI GRAM Protocol (found in GT3 and GT4 respectively)

While the WSRF and OGSI GRAM protocols are more complex in terms of transactions than its GT2 counterpart, their usage of ports is simpler. As described in [19], all traffic to initiate and control jobs goes through a single hosting environment which listens, by default, on port 8080/tcp from ephemeral ports on a client.

Notifications of events on the job can be made from the GRAM service back to the client. These will come from ephemeral ports on the server host back to controlled ephemeral ports on the client.

As in pre-ws GRAM, file staging will result in connection out from the server to *controllable ephemeral ports* on the client (or other host specified by the client for file staging).

3.3 Data Movement (GridFTP)

The Globus GridFTP server appears in all versions of the toolkit and is a pre-ws component. Though the GridFTP server was re-implemented for the GT4 release (previous version were based on the wuftpd code base), the network traffic characteristics of all versions are identical. The GridFTP protocol, like the standard FTP protocol, has two connections: a control channel, over which commands and responses flow, and a data channel, over which data (e.g. files) are transfers. This results in two traffic components that should be allowed by firewall policy.

The control connection is established from a *controllable ephemeral port* on the client to the well-known static port of 2811/tcp on the server.

Establishment of the data connection depends on whether there is single data channel or multiple, parallel connections (a feature offered by GridFTP for high performance).

In the case of a single data channel, the connection is established from a *controllable ephemeral port* on the client to a *controllable ephemeral port* on the server. In the case of third-party transfers (a client controlling a file transfer between two servers), this connection may be from a server to another server.

In the case of multiple parallel data channels, the direction of the connection establishment is dependant on the direction of data flow – the connection will be in the same direction the data flow.

For example if a file is being sent with parallel data channels from the client to the server (an FTP PUT operation) then multiple connections will be established from *controllable ephemeral ports* on the client to a single *controllable ephemeral port* on the server. If the client were retrieving the file (an FTP GET operation) with parallel data channels then multiple connections will be established from *controllable ephemeral ports* on the server to a single *controllable ephemeral port* on the client.

Both the control connection and the data connection(s) of GridFTP are authenticated with GSI as described in Section 3.1. The control channel is integrity protected. The data channel is optionally integrity protected and encrypted if requested by the client.

3.4 Monitoring and Discovery Service (MDS)

MDS serves as a means for users to discover what machines are available and to query the machines to determine their attributes so they can ascertain if they are appropriate to their needs. When architecturally similar in pre-ws and web services versions of the Globus Toolkit, the traffic characteristics differ between the two versions and are described separately here.

3.4.1 Pre-WS MDS (GT2 and subsequent versions)

The Pre-WS MDS architecture has two main components: Grid Resource Information Servers (GRISs) and Grid Information Index Servers (GIISs). GRISs run on resources and respond directly to any queries. GRISs also typically register themselves to one or more GIISs. This allows users to query a GIIS, see all the available resources in an organization and then query the resource's GRIS directory or through the GIIS.

This results in two traffic components that should be allowed by firewall policy:

1. Queries will take the form of a connection to a GRIS, from either a user directly or forwarded by a GIIS, from an ephemeral port to the well-known static port of 2135/tcp on the server.
2. Registrations will take the form of an outgoing connection from an ephemeral port on the server to the well-known static port of 2135/tcp on the GIIS. GIIS may also need to send traffic out to GRISs on the well-known static port of 2135 from an ephemeral port.

Pre-WS MDS connections are optionally authenticated with GSI as described in Section 3.1. A server can be configured to require authenticated connections.

3.4.2 OGSI and WSRF MDS

The web services-based MDS model is similar to the pre-ws MDS model described in the previous section. Each service and job publishes some set of data about the local host or itself. Aggregators serve the roll of GIIS and combine information from related resources.

The hosting environment serves to route the query to the appropriate service handles all queries for information. This results in traffic from ephemeral ports on clients to port 8080/tcp on the server.

3.5 GSI-Enabled OpenSSH

GSI-enabled OpenSSH is part of the Globus Toolkit as of the GT4 release. Prior to that release it was distributed separately. More information on GSI-enabled OpenSSH can be found at [6].

GSI-enabled OpenSSH has exactly the same network traffic characteristics as normal OpenSSH [14]. Authentication with GSI is another authentication mechanism added to the suite of authentication mechanisms supported by OpenSSH. This traffic pattern of OpenSSH is a connection from an ephemeral port on the client to the well-known static port of 22/tcp on the server.

OpenSSH connections authenticated with GSI are protected with OpenSSH's normal message protection mechanisms [14].

3.6 MyProxy

MyProxy is part of the Globus Toolkit as of the GT4 release. Prior to that release it was distributed separately. More information on GSI-enabled OpenSSH can be found at [13].

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.

3.7 MPICH-G2

While not part of the Globus Toolkit, MPICH-G2 is often used with the Toolkit. The reader may find information on using MPICH-G2 in the Firewalls section of the MPICH-G2 webpage [11].

3.8 Summary of Pre-Web Service GT Traffic

Table 1 summarizes the traffic characteristics for pre-web services Globus Toolkit services.

Table 1: Pre-WS network traffic characteristics

Application	Network Ports	Comments
GRAM Gatekeeper (to start jobs)	To 2119/tcp on server from controllable ephemeral port on client	Connections back to client (controllable ephemeral port to controllable ephemeral port) required if executable or data staged from client or output from job sent back to client. Port 2119/tcp defined by IANA
GRAM Job-Manager	From controllable ephemeral port on client to controllable ephemeral port on server.	Port on server selected when original connection made by the client to the Gatekeeper and returned to the client in a URL. May result in connection back to client from ephemeral port on server to controllable ephemeral port on client.
MDS Grid Resource Information Service (GRIS) or Grid Information Index Service (GIIS)	From ephemeral port on client to port 2135/tcp on server	To hosts running a GRIS service (typically all machines that run the Globus services) Port 2135/tcp defined by IANA
MDS GRIS or GIIS registration to a parent GIIS	To 2135/tcp on parent GIIS from ephemeral port on child GRIS/GIIS To 2135/tcp on child GRIS/GIIS from ephemeral port on parent GIIS	Connections from GRIS to GIIS are for registration. Connections from GIIS to GRIS are for queries. Port 2135/tcp defined by IANA
GridFTP	From controllable ephemeral port on client to port 2811/tcp on server for control channel.	Port 2811/tcpdefined by IANA For information on data channel please see Section 3.3
GSI-Enabled SSH	From ephemeral port on client to port 22/tcp on server.	Same as standard SSH. Port 22/tcp defined by IANA.
MyProxy	From ephemeral port on client to port 7512/tcp on server.	Default. Can be modified by site.

3.9 Summary of OGSI and WSRF (GT3 and GT4) Traffic

This table is intended to summarize the Globus Toolkit services based on web services (either OGSI or WSRF).

Table 2: GT3 network traffic characteristics

Application	Network Ports	Comments
GRAM (job startup and control)	To 8080/tcp on server from ephemeral port on client.	Connections back to client (controllable ephemeral port to controllable ephemeral port) required if executable or data staged from client or output from job sent back to client. Port 8080/tcp is default and configurable.
MDS	From ephemeral port on client to port 8080/tcp on service.	Same port as for GRAM.
GridFTP	From controllable ephemeral port on client to port 2811/tcp on server for control channel.	Port 2811/tcpdefined by IANA For information on data channel please see Section 3.3 Same as pre-WS GT.
GSI-Enabled SSH	From ephemeral port on client to port 22/tcp on server.	Same as standard SSH. Port 22/tcp defined by IANA.
MyProxy	From ephemeral port on client to port 7512/tcp on server.	Default. Can be modified by site.

4 Controlling the Ephemeral Port Range

Here we describe how *controllable ephemeral ports* in the Globus Toolkit can be restricted to a given range and give examples of common techniques for configuring services.

4.1 Pre-Web Services GT

For pre-web services Globus Toolkit components, setting the environment variable GLOBUS_TCP_PORT_RANGE can restrict ephemeral ports. The value of this variable should be formatted as *min*,*max* (a comma separated pair). This will cause the GT pre-WS libraries (specifically GlobusIO) to select port numbers for controllable ports in that specified range.

For example:

```
% GLOBUS_TCP_PORT_RANGE=40000,40010
% export GLOBUS_TCP_PORT_RANGE
% globus-gass-server
https://globicus.lbl.gov:40000
^C
%
```

4.1.1 Configuring the pre-WS GRAM Gatekeeper/Job-Manager to use GLOBUS_TCP_PORT_RANGE

There are two ways to configure the Job-Manager (JM) to use GLOBUS_TCP_PORT_RANGE, either by using a wrapper script or by setting the environment variable in inetd/xinetd.

Using a wrapper script:

This method involves replacing the JM binary with a wrapper script that sets the environment variable and then executes the actual JM binary (note that when the JM is invoked by the Gatekeeper the user has already been authenticated and a setuid has been done, so this doesn't have the risk of a potential exploit of the script).

To wrap the JM start by moving the real JM executable aside:

```
% mv $GLOBUS_LOCATION/libexec/globus-job-manager \
        $GLOBUS_LOCATION/libexec/globus-job-manager.real
```

And replace it with a script such as the following:

```
% cat > $GLOBUS_LOCATION/libexec/globus-job-manager
#!/bin/sh
# Replace 40000,45000 here with your actual port range
GLOBUS_TCP_PORT_RANGE=40000,45000
export GLOBUS_TCP_PORT_RANGE
# Note: Replace G_L on following line with actual path of
# $GLOBUS_LOCATION
# Don't use the environment variable as it may not be set.
```

```
exec G_L/libexec/globus-job-manager.real "$@"
^D
% chmod 755 $GLOBUS_LOCATION/libexec/globus-job-manager
```

Setting the environment variable in inetd or xinetd:

For inetd change the line in your inetd.conf file that starts the Gatekeeper service to use /bin/env to set GLOBUS_TCP_PORT_RANGE. For example:

```
gatekeeper stream tcp nowait root \
        /bin/env env GLOBUS_TCP_PORT_RANGE=40000,45000 \
        GLOBUS_LOCATION/sbin/globus-gatekeeper \
        -conf GLOBUS_LOCATION/etc/globus-gatekeeper.conf
```

The above string would need to be customized to reflect your configuration by replacing GLOBUS_LOCATION with the actual value of $GLOBUS_LOCATION and 40000,45000 with you choice of port range.

For xinetd add an env line to your /etc/xinet.d/globus-gatekeeper file. For example:

```
service globus-gatekeeper
{
      socket_type  = stream
      protocol     = tcp
      wait         = no
      user         = root
      server       = GLOBUS_LOCATION/sbin/globus-gatekeeper
      server_args  = -conf GLOBUS_LOCATION/etc/globus-gatekeeper.conf
      disable      = no
      env          += GLOBUS_TCP_PORT_RANGE=40000,45000
}
```

The above string would need to be customized to reflect your configuration by replacing GLOBUS_LOCATION with the actual value of $GLOBUS_LOCATION and 40000,45000 with you choice of port range.

4.1.2 Configuring GridFTP to use GLOBUS_TCP_PORT_RANGE

Since the GridFTP server is started by inetd/xinetd and not by the Gatekeeper, this means that it is not safe to wrap the executable with a script wrapper like it is with the job manager since it will be running as root. This means you need to configure inetd or xinetd to set the environment variable when it invokes the GridFTP server.

For inetd change the line in your inetd.conf file that starts the service to use /bin/env to set GLOBUS_TCP_PORT_RANGE. For example:

```
gsiftp stream tcp nowait root \
        /bin/env env GLOBUS_TCP_PORT_RANGE=40000,45000 \
        GLOBUS_LOCATION/sbin/in.ftpd -l -a
```

The above string would need to be customized to reflect your configuration by replacing GLOBUS_LOCATION with the actual value of $GLOBUS_LOCATION and 40000,45000 with you choice of port range.

For xinetd add an env line to your /etc/xinet.d/gsiftp file. For example:

```
service gsiftp
{
        socket_type  = stream
        protocol     = tcp
        wait         = no
        user         = root
        server       = GLOBUS_LOCATION/sbin/in.ftpd
        server_args  = -l -a
        disable      = no
        env          += GLOBUS_TCP_PORT_RANGE=40000,45000
}
```

The above string would need to be customized to reflect your configuration by replacing GLOBUS_LOCATION with the actual value of $GLOBUS_LOCATION and 40000,45000 with you choice of port range.

4.1.3 Configuring MDS to use GLOBUS_TCP_PORT_RANGE

MDS does not open ephemeral ports to accept connections so it does not need to be configured to use GLOBUS_TCP_PORT_RANGE.

However if it were needed for some reason, the environment variable could be set in the $GLOBUS_LOCATION/sbin/SXXgris script.

4.2 Web Services versions of GT

Web services versions of the Globus Toolkit are composed of applications and services written in C and in Java. Both of these have a different means of configuration for controlling the ephemeral port ranges as described in the subsequent sections.

4.2.1 C Libraries, Client Applications and Services

Web Services portions of the Globus Toolkit written in C, which includes some client applications, some client libraries and the GridFTP server, use the same libraries as the pre-web services portions. This allows their ephemeral ports to be controlled with the GLOBUS_TCP_PORT_RANGE environment variable as described in Section 4.

4.2.2 Java Libraries, Client Applications and Services

For web services Java-based code the system property "org.globus.tcp.port.range" effects ports used for incoming connections in the same manner as the GLOBUS_TCP_PORT_RANGE environment variable does as described in the previous section.

This value can be set in the following manners:

- It can passed on a command line:

```
%java -Dorg.globus.tcp.port.range=5000,6000
```

- It can be specified by the application directly:

```
System.setProperty("tcp.port.range", "5000,6000")
```

- It can also be placed in the file *~/.globus/cog.properties* where it will be read automatically by the libraries. The line in the file should appear like:

```
tcp.port.range=5000,6000
```

5 Client Site Firewall Requirements

This section describes the requirements placed on firewalls at sites containing Globus Toolkit clients. Note that often jobs submitted to sites running Globus services will act as clients (e.g. retrieving files needed by the job, spawning subjobs), so server sites will also have client site requirements.

5.1 Allowed Outgoing Ports

Clients need to be able to make outgoing connections freely from ephemeral ports on hosts at the client site to all ports at server sites.

5.2 Allowed Incoming Ports

As described in Section 3.2, the Globus Toolkit GRAM service use callbacks to communicate state changes to clients and, optionally, to stage files to/from the client. If connections are not allowed back to Globus Toolkit clients, the following restrictions will be in effect:

- You can't do a job submission request and redirect the output back to the client. This means the globus-job-run command won't work. globus-job-submit will work, but you can't use globus-job-get-output. globusrun with the -o option also will not work.
- Staging to or from the client will also not work, which precludes the -s and -w options.
- The client cannot be notified of state changes in the job, e.g. completion. With the web services GRAM client in GT4 (globusrun-ws), this can be worked around by having the client poll through the use of the *–status* commandline option (note that this comes with some penalty in terms of greater delay in determining state changes).

To allow these callbacks client sites should allow incoming connection in the ephemeral port range. Client sites wishing to restrict incoming connections in the ephemeral port range should select a port range for their site. The size of this range should be approximately 10 ports per expected simultaneous user on a given host, though this may vary depending on the actual usage characteristics. Hosts on which clients run should have the GLOBUS_TCP_PORT_RANGE environment variable set for the users to reflect the site's chosen range as described in Section 4.

5.3 Network Address Translation (NAT)

Clients behind NATs will be restricted as described in Section 5.2 unless the firewall and site hosts are configured to allow incoming connections.

This configuration involves:

1. Select a separate portion of the ephemeral port range for each host at the site on which clients will be running (e.g. 45000-45099 for host A, 45100-45199 for host B, etc.).
2. Configure the NAT to direct incoming connections in the port range for each host back to the appropriate host (e.g., configure 45000-45099 on the NAT to forward to 45000-45099 on host A).
3. Configure the Globus Toolkit clients on each site host to use the selected port range for the host using the techniques described in Section 4.
4. Configure Globus Toolkit clients to advertise the firewall as the hostname to use for callbacks from the server host. This is done using the GLOBUS_HOSTNAME environment variable. The client must also have the GLOBUS_HOSTNAME environment variable set to the hostname of the external side of the NAT firewall. This will cause the client software to advertise the firewall's hostname as the hostname to be used for callbacks causing connections from the server intended for it to go to the firewall (which redirects them to the client).

6 Server Site Firewall Requirements

This section describes firewall policy requirements at sites that host Grid services. Sites that host Grid services often host Grid clients, however the policy requirements described in this section are adequate for clients as well.

6.1 Allowed Incoming Ports

A server site should allow incoming connections to the well-known *Grid Service Ports* as well as ephemeral ports.

For pre-ws services these ports are 22/tcp (for gsi-enabled openssh), 2119/tcp (for GRAM), 2135/tcp for MDS and 2811/tcp for GridFTP.

For web services-based services these ports are 22/tcp (for gsi-enabled openssh), 2811/tcp for GridFTP and 8080/tcp for GRAM and MDS.

A server not allowing incoming connections in the ephemeral port range will have the following restrictions:

- If port 2119/tcp is open, pre-ws GRAM will allow jobs to be submitted, but further management of the jobs will not be possible.
- While it will be possible to make GridFTP control connections if port 2811/tcp is open, it will not possible to actually get or put files.
- Pre-ws MDS will function normally if port 2135/tcp is open

Server sites wishing to restrict incoming connections in the ephemeral port range should select a range of port numbers. The size of this range should be approximately 20 ports per expected simultaneous user on a given host, though this may vary depending on the actual usage characteristics. While it will take some operational experience to determine just how big this range needs to be, it is suggested that any major server site open a port range of at least a few hundred ports. Grid Services should configured as described in Section 4 to reflect the site's chosen range.

6.2 Allowed Outgoing Ports

Server sites should allow outgoing connections freely from ephemeral ports at the server site to ephemeral ports at client sites as well as to *Grid Service Ports* at other sites.

6.3 Network Address Translation (NAT)

Grid services are not supported to work behind NAT firewalls because the security mechanisms employed by Globus require knowledge of the actual IP address of the host that is being connected to.

We do note there have been some successes in running GT services behind NAT firewalls. In particular we point out Steve Thorpe's success with the GridFTP server documented at [16].

7 Work to Improve Firewall Interoperability

This section describes current work to improve the interoperability of the Globus Toolkit with firewalls and discusses some ideas for future work.

7.1 Improvements in GT3

The GT3.0 release addressed the problem of multiple ephemeral ports on a server for job management by funneling all traffic to managed jobs through a "proxy router". The proxy router is installed as part of the GT3 resource management system. All traffic to managed jobs from clients is send to the proxy router, which directs it to the ultimate destination. In turn, return traffic from managed jobs back to the clients also passes through the router.

7.2 Improvements in GT4

The GT4 GRAM client allow for the use of polling to obtain event changes in the job, as describe in section 5.2. This alleviates the need for connection back to the client from the server.

8 References

[1] T. Dierks, and C. Allen, "The TLS Protocol, Version 1.0." RFC 2246, January 1999

[2] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations." International J. Supercomputer Applications, 15(3), 2001. (http://www.globus.org/research/papers.html#anatomy)

[3] I. Foster, and C. Kesselman, "Globus: A Toolkit-Based Grid Architecture". I. Foster and C. Kesselman, eds. The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999, 259-278.

[4] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Globus Project, 2002. http://www.globus.org/research/papers/ogsa.pdf.

[5] "Globus and Firewalls." XXX http://www.globus.org/security/v2.0/firewalls.html

[6] GSI-Enabled OpenSSH web page, http://www.ncsa.uiuc.edu/Divisions/ACES/GSI/openssh/

[7] The Globus Toolkit web page. http://www.globus.org/toolkit/

[8] R. Housley, W. Polk, W. Ford, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile." RFC 3280.

[9] Internet Assigned Numbers Authority. http://www.iana.org/

[10] J. Kohl, C. Neuman, "The Kerberos Network Authentication Service (V5)." RFC 1510.

[11] MPICH-G2 web page, http://www.globus.org/mpi

[12] J. Novotny, S. Tuecke, V. Welch, "An Online Credential Repository for the Grid: MyProxy." Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001. http://www.globus.org/research/papers.html#MyProxy

[13] "MyProxy Online Credential Repository" http://myproxy.ncsa.uiuc.edu

[14] "OpenSSH Features." http://www.openssh.org/features.html

[15] L. Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke, "A Community Authorization Service for Group Collaboration." Proceedings of the International Workshop on Policies for Distributed Systems and Networks, 2001. http://www.globus.org/research/papers.html#CAS-2002

[16] Thorpe, Steve. Email to globus-discuss regarding GridFTP server behind a NAT firewall. January, 2005. http://www-unix.globus.org/mail_archive/discuss/2005/01/msg00216.html

[17] S. Tuecke, et. al., "Internet X.509 Public Key Infrastructure Proxy Certificate Profile.", (Latest version at http://www.ietf.org/html.charters/pkix-charter.html)

[18] S. Tuecke, et. al. "Open Grid Services infrastrcture (OGSI) Version 1.0", GFD-R-P.15, June, 2003. http://www.ggf.org/documents/GWD-R/GFD-R.015.pdf

[19] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Cajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke, "Security for Grid Services". To appear HPDC 2003. http://www.globus.org/Security/GSI3/GT3-Security-HPDC.pdf

[20] V. Welch, et. al. "Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective", December, 2004. http://www-unix.globus.org/toolkit/docs/development/4.0-drafts/security/GT4-GSI-Overview.pdf

[21] "The WS-Resource Framework", March 2004. http://www.globus.org/wsrf/

9 Acknowledgements

The writing of this document was funded by the U.S. Department of Energy as part of the DOE Science Grid SciDAC project and by the National Science Foundation under the NMI GridsCenter project.

Numerous members of the Globus Project contributed to this document, including Bill Allcock, Jarek Gawor, Nick Karonis, Sam Lang, Stuart Martin, Joe Link and Thomas Sandholm.

The paper "A report on Experiences Operating the Globus Toolkit through a Firewall" (Version 1, September 2001) by M. Baker, H. Ong, G. Smith was useful in designing the tests that were used to verify the GT2 information in this document.

Steve Chan of NERSC provided a firewall testbed that was used for verifying the GT2 information in this document.

Thanks to Dirk Breuer for corrections regarding GridFTP with parallel data channels and to Steve Thorpe for his work on getting a GridFTP server to work behind a NAT firewall.