

## Comments on Security from commissioned reports on Globus and OGSA

### *On Globus*

**From a consolidated report (Oct 2001) – .J. Allan, D.R.S. Boyd, T. Folkes, C. Greenough, D. Hanlon, R.P. Middleton, R.A. Sansum , CLRC e-Science Centre  
Elson Mourão, Rob Baxter, David Henty, Mark Parsons , Edinburgh Parallel Computing Centre  
John Brooke, W.T. Hewitt, Mike Daw, Jon MacLaren, Jon Gibson, Graham Riley, Stephen Pickles , Manchester Research Centre for Computational Science**

### **On Globus 1.0**

#### **Security:**

Globus incorporates a very well implemented security infrastructure based on SSL, the current *de facto* Internet standard (i.e. that adopted by the Internet Engineering Task Force, IETF). This infrastructure is used by Globus for authentication of both resources and users. There is an ongoing discussion about Kerberos as well in addition to PKI security services.

#### **Resource discovery and selection:**

Resource authentication is by a PKI system using X.509v3 certificates. Resource discovery is by selective queries to the LDAP registry. The information provided must be dynamic and as up to date as possible, e.g. regarding cpu load or available memory or disk space. Additional resource discovery mechanisms are required to address software availability and authorisation issues.

#### **User management:**

User authentication and authorisation is by a PKI system using X.509v3 certificates and mapping of Grid-wide distinguished names (DNs) to local UNIX ids.

The current Globus CA, used for test purposes, is not designed to be satisfactory for a production Grid. It is important that user and resource certificates are maintained correctly as there are otherwise possible security loopholes. These are however no more serious than those currently faced on the Web, except that there are many more certificates in circulation. The Globus team has thought about this rather carefully and the proposed solution works well. There are however difficulties with additional packages. For instance the CA software is complex to manage. The complexity of carrying out the installations and proliferation of client-server interactions in the MyProxy certificate repository is likely to hinder widespread adoption of this technology. For example a Web portal must be set up to talk to an individual myproxy-server. This means that a Globus client must store a proxy certificate on exactly the server that is later to be used. If you want to use several different Web portals this makes the procedure rather unsatisfactory both from the management and scalability point of view. It is however probably the best current working scenario if good security is required.

### **On Globus 2.0**

Plans for development in the coming year include on-line credential and certificate authorities and the development of Community Authorization Servers. The on-line credential and certificate authorities will allow the user to go to an on-line service and get a dynamic credential, rather than using a private key living on the user's disk. The goal of this scheme is to authorize new users or resources without requiring that every existing resource profile be updated to reflect the existence of a new user or requiring every user profile to be updated for each new resource. The Community Authorization Service (CAS) is based on the observation that control of resources is often group-based. The CAS allows users to define communities and to authorize resource use based on community membership. The CAS issues capabilities that allow the bearer access to a resource. The capability issued by the CAS contains information about the end user to allow local resources to perform audit and security functions. Thus, while the CAS has a global view of groups and their capabilities, local resources still maintain access control and enforce local policies. This local policy enforcement at resources level is an issue that needs more research and represents an opportunity for innovation by the DataGrid.

## ***On OGSA***

### **Extract from an Evaluation (April 2002) by Peter Z. Kunszt, IT Division - Database Group, CERN**

This issue is touched but not elaborated on sufficiently. The hosting environment gets the burden of authentication – which is reasonable – but there is no discussion on how local and VO-wide security policies are enforced also on authentication. Is there the need for a Grid Service that deals with these issues or should each of the services have an interface addressing this, making it part of the GridService base interface? New developments in this area are necessary.

By relying on Web Services, the strong industrial drive to come to a solution in this area will help speed up the process to design a suitable security infrastructure. Recent press releases by Microsoft and IBM have indicated the industry's commitment in this area.

There needs to be a lot of effort put into this domain also from the Grid community to check how existing Grid security infrastructures might interoperate with Web Service security mechanisms.

Security will have to be dealt with very soon within OGSA since it will depend on the success of the underlying security framework. Open questions include: How are VO-wide policies applied? How are local security policies enforced? What is the role of hosting environments? How is an audit performed? Can a user belong to more than one VO and use both resources even if the security mechanisms differ?

### **Extract from an Analysis (April 2002) of OGSA by Dennis Gannon, Kenneth Chiu, Madhusudhan Govindaraju, Aleksander Slominski, Department of Computer Science, Indiana University**

#### **The Grid Service Handle (GSH)**

A service instance does not necessarily correspond to an operating system entity, but is rather defined by its behavior. That is, any manifestation (process, etc.) of a service is the same instance as long as it *acts* like its the same instance, as seen through its interfaces. From [1], Sec. 6.1 (emphasis original):

Grid services can maintain internal state for the lifetime of the service. The existence of state distinguishes one *instance* of a service from another that provides the same interface.

Also, on page 19 of [1], it states, "If a Grid service fails and is restarted in such a way as to preserve its state, then it is essentially the same instance, and the same GSH can be used."

So the scenario for upgrading an instance would be something like a registry or factory that keeps shareable state on disk. To replace an old registry implementation with a new registry implementation, first start the new registry process, and have it take over the GSH by directing all new mappings to its GSR. Because the old and the new share state on disk, the old registry process can keep fulfilling requests made on its GSR. Eventually, all the old GSRs expire, and the old registry process can be shutdown. From the clients viewpoint, there was ever only one instance.

A GSH for any Grid Service is required to be a URL that contains the GSHHomeHandleMapID. This GSHHomeHandleMapID is globally unique for mapping the GSH to one or more valid Grid Service References (GSRs). The GSHHomeHandlerMapID includes a hostname at which the HandleMap service resides. The HandleMap provides a mapping from a GSH to a GSR. This may present several problems

- Since the location of the HandleMapper is included in the GSH, it is not possible to move the HandleMap service to another host. For the lifetime of the GSR, the Mapper service has to reside on a fixed machine. However, it is reasonable to expect that over a period of time there will be a need to move the HandleMap service to a different machine or administrative domain. In such cases the binding from a GSH to a GSR may be lost.

- In the absence of a well-defined security policy, it is not possible for distrustful organizations to communicate and obtain the mapping from a GSH to a GSR.

There is one solution that should be considered. The Secure Grid Naming Protocol (SGNP) has been proposed to the Grid Forum to alleviate the location-dependency and security problems that arise with GSH and HandleMap service of OGSA. The SGNP model can be considered as an alternative way of defining the specification for the HandleMapper PortType in OGSA. SGNP provides a mechanism for "Naming" of Grid resources. It defines a scheme that assigns "logical" and thus location-independent names to Grid resources. The scheme obviates the need for authentication of two Grid resources via a trusted third party. The logical name is a combination of the identity and security information of a Grid resource. The security information can be (1) nothing (2) RSA public Key (3) X.509 certificates (4) Open PGP certificate. An SGNP name is a Location Independent Object Identifier (LOID), which is globally unique and immutable for the lifetime of the Grid resource. The actual location of a Grid resource can be determined by associating the LOID with one or more communication protocols and network endpoints. This two level naming scheme of SGNP (LOID and its mapping) allows a Grid resource to be migrated both spatially and temporally. The binding of a LOID can be represented as a WSDL document. SGNP defines a Grid Naming Service (GNS), with a well-known binding that provides clients access to SGNP naming services. This naming service provides access to authoritative LOID-to-Binding mappings and a Resolver Hierarchy Service that resolves increasingly specific portions of LOIDS. We feel this is an idea of considerable merit and it is worth evaluating.

## **Notification**

We feel that notification is an extremely important service that OGSA must support. However, we have several technical quibbles with what is proposed here.

.NotificationSourceTopicNames should return list of QNames and not nmtokens. The sink in NotificationSource::SubscribeToNotificationTopic assumes that GSH can be always mapped to a GSR, which may not always be the case. We feel that returning actual GSR would be better. By examining GSR, the client can find out how long the service instance is valid and also if a GSH mapping is supported.

The specification also requires a notification sink to expose a network accessible endpoint when calling SubscribeToNotificationTopic because the notification sink must have a GSR that identifies network accessible location to deliver notifications. That will not work for services or clients that are behind a firewall - not only are their endpoints inaccessible, the GSH to GSR mapping service may also be inaccessible. This is related to bigger problem of participation in Grid Web Services by small devices that may not have permanent IP address and for OGSA clients that reside behind firewalls or NAT and need to access notification services. We suggest augmenting the notification source interface to allow a client to subscribe for information pulling. In this way, clients behind firewall can request and pull notification data.

## **Grid Firewall recommendations (February 2001), Stephen Booth, EPCC for the ETF**

### **Introduction**

Internet firewalls are now considered to be an essential part of computer security. This document addresses the role of firewalls in a grid environment.

The purpose of a firewall is to restrict traffic between a protected network and the Internet. In principle they should not be necessary, every computer system should be completely secure against attack from the Internet. In practice this is difficult to achieve, the number of Internet services supported by modern operating systems are very large and it is very easy to accidentally install or activate unwanted services. In addition there are many services that are acceptable for use within the local network but should be denied to external users. By restricting Internet traffic to a set of permitted services administrators are able to concentrate their attention on the security of these permitted services.

One traditional approach to firewall design is to deny any direct connection to the Internet from the internal network. Permitted services are passed through the firewall by proxy servers running on the firewall host.

This approach is too restrictive for network centric Grid activities. The alternative is to filter the Internet traffic using a packet filtering firewall.

### Packet filtering

**tcp** and **udp** packet headers contain Internet addresses and port numbers for the source and destination machines. These can be used to identify the service being used and hence restrict traffic to particular services. In most cases the service provider will use a well known port number and the client will allocate a random port number (>1024). Firewall administrators will therefore want to apply different rules depending on whether the internal or external host initiated the connection, so as not to block return packets for connections made by Grid users from the local network while still restricting incoming connections.

**Stateful** firewalls explicitly track the state of tcp connections and can filter incoming and outgoing connections separately. Other firewalls can achieve something similar by allowing all packets from established connections (packets with the acknowledge bit set) and filtering the initial **tcp** handshake packet setting up the connection. Globus Here is the network traffic generated by Globus and GSI applications:

Application	Globus Version	Network Ports	Network Addresses	Comments
Gatekeeper	1.1.3 and later	2119/tcp	From client machines to hosts running Globus gatekeepers. Direction of traffic depends on location of client and server.	Defined by IANA.
MDS Grid Resource Information Service (GRIS)	1.1.3 and later	2135/tcp 2135/udp	To hosts running a GRIS service (typically all machines that run the Globus services).	Defined by IANA.
MDS Grid Information Index Service (GIIS)	1.1.3 and 1.1.4	site-selected	To hosts running a GIIS service (typically a small number of machines that index information from multiple GRIS services for searching purposes)	Each site defines its own GIIS hosts and port numbers.
MDS Grid Information Index Service (GIIS)	2.0 and above	2135/tcp 2135/udp	To hosts running a GIIS service (typically a small number of machines that index information from multiple GRIS services for searching purposes).	In Globus 2.0 GIIS and GRIS are combined in a single service
GridFTP	All	2811/tcp (control) See below for Data channel	To hosts running GSI-enabled ftpd.	Defined by IANA.
GSI-Enabled SSH	All	22/tcp	To hosts running GSI-enabled sshd.	Same as normal ssh
MyProxy	All	7512/tcp	From myproxy-init or myproxy-get-delegation to the	Default. Can be modified by

			MyProxy server.	site.
--	--	--	-----------------	-------

Notes on above table: All tcp connections have a return connection on a client port (>1024). Therefore outgoing connections should be enabled for this range. If you register an MDS server with a remote MDS you must open the server to incoming connections from that MDS server.

### Limiting GlobusIO, GridFTP data channel and Nexus port range usage

In addition to permanent services on fixed ports Globus may also allocate random ports for transient services, such as the data channel for GridFTP or GASS services started by user jobs.

It is possible to restrict the port numbers that globus\_io and Nexus will associate with a listener. By setting the environment variable GLOBUS\_TCP\_PORT\_RANGE <min,max> (as a comma separated pair) the libraries will only create listeners with ports in that specified range.

Firewalls can then open an equivalent restricted range of high numbered port numbers that are unlikely to be used by other services.

This variable needs to be set for all jobs started by Globus. This can be achieved by renaming the *globus-job-manager* binary to *globus-job-manager.real* and replacing it with the following script.

```
#!/bin/sh

GLOBUS_TCP_PORT_RANGE=50000,52000

# Exporting this still allows job submission AND restricts the
# GRAM port range - so it works!
export GLOBUS_TCP_PORT_RANGE

# run the real program from the same directory as the wrapper
exec ${0}.real $*
```

It is usually simplest to allow all connections that are initiated from within the firewall. If you wish to restrict the range of ports used by outgoing connections local Globus users will also need to set this environment variable to correspond to the permitted range.