R-GMA Security
Current Status, Authorization Design and Implementation Strategy

Linda Cornwall

WP3 meeting Jan 2004

Authentication in R-GMA

- The edg-java-security Trustmanager has been integrated into R-GMA
  - Both the Java Servlets and the Java API
- Authentication takes place on connection to the tomcat servlet
- A `Trust Properties' file defines where to find the appropriate certificates (and CRLs)
  - Both for the R-GMA Service and for the User or other service connecting to R-GMA
- Works for CA signed certificates (e.g. Host Certificates) and Proxy certificates.
  - Both for Authenticating the R-GMA service and for Users
  - Including re-loading the certificate from file if the connection fails in case the proxy has expired

Authentication – other APIs

- C++ API has been written (mainly by Jason)
- Uses the same TrustProperties file as the Java API
- Defaults to use the GSI Proxy generated by grid-proxy-init
- Copes when R-GMA is authenticated with a CA signed certificate and key or with a proxy
- All other APIs based on this

Authentication - Current restrictions

- No Host name verifier in Java
  - Rogue service an authenticate with a stolen certificate
- No Delegation
  - Client Authenticates with servlet, servlets authenticate with each other
- Cannot Authenticate Service with Proxy
  - Nothing to do with R-GMA software, edg-java-security or EDG
  - Standards compliant browsers (e.g. Internet Explorer) only allow services to be authenticated with CA signed certificates.
  - Currently IETF defining standards for Authentication using Proxies
- Secure connections for some services switched off in EDG testbeds

Is R-GMA Secure?

- No
- Need to look at Security Holes and close them
  - E.g.MySQL User Name and Password
- Need to look at the design, the implementation and all connections
  - close any holes

What is Authorization?

- Whereby a principal is allowed to or prevented from carrying out an action.
- In edg there are requirements to carry out an authorization decision based on
  - Specific DN
  - VO membership(s)
  - Role within VO
  - Group within VO
  - By allowing anyone with an acceptable certificate to carry out an action
  - Allowing anyone to carry out an action

Current Authorization in R-GMA

WP3

- Only Authorization in R-GMA is for the Registration of Producers and Consumers
- Not based on EDG Authorization model or methods
- It is possible to set up R-GMA such that only Producers and Consumers from a defined set of URLs can register.
- Alternatively all Producers and Consumers can Register, except for those from a list of URLs

VOMS

- VOMS = `Virtual Organisation Membership Service'
- VOMS allows a user (or any principal) to generate a short-time proxy where the public certificate has VOMS credentials added
- The VOMS certificate contains proof that a user is a member of a VO, is a member of certain groups in a VO, and has certain roles and capabilities within the VO

Authorization – EDG Principle

WP3

- Principle within EDG is that the Authorization decision is made close to the resource or data, based on a combination of local Authorization information and attributes from the user (e.g. VOMS)
- This enables e.g. resource owner or administrator, or a file owner or administrator to keep control over it's access.
- Details of EDG Security Design is in D7.6

Course grained vs fine grained authorization

- Course grained – authorization on front of service (I.e. y/n can the person connect).
  - "Is the user allowed to use this service – if so – what role"
- Fine grained – authorization takes place within a service.
  - E.g. can this user read this file?
- R-GMA could have services which decide whether or not a connection is allowed, as well as services which decide whether to satisfy the request within the service.
- For R-GMA authorization decisions being made within services – a combination will be rather cumbersome.

Distributed and Onward Connection Authorization

1. Only pass on information to authorized principals
   - Servlets may be authorized
   - Trust Authorized servlets to comply with the rules
   - Principals external to R-GMA may be authorized
2. Producers only pass information into R-GMA if requested by an authorized principal
   - Need proof that it has been requested by an authorized principal
   - This requires delegation
3. Encrypt information

Authorization on Views

WP3

- R-GMA is not as simple as `can this principal access this file', authorization needs to be based on views of the tables – as talked about by Steve Fisher at Coseners last year
- Need to develop a way of specifying how to carry out authz based on a view of a table
  - GACL on a view? (Has the problem, I think, where we can't say e.g. O.K. if DN matches)
  - Our own?
  - Something from OGSA Authz?

Where to specify Authz rules?

- Schema
  - Define Authz rules in the schema.
  - Good for merging information from different sources
  - Means it's not necessary to copy Authz rules with the info
  - Good for allowing Access Control on any view you like
  - `Mediator' can make a decision on what queries may be successful
  - Does not allow producers control over data access ☹
- Registry
  - Producers define the rules in the registry
  - Makes it necessary to copy authz rules around
  - Makes it difficult to authorize on views other than a `per row' if data is merged from more than one producer
  - `Mediator' can still make a decision on what queries may be successful

Where to specifiy rules - cont

- Per item
  - Producers define `per item' rules
  - Makes it necessary to copy authz rules around
  - Probably not possible to authorize on views other than a `per row' if data is merged from more than one producer
  - `Mediator' cannot make any decision on what queries may be successful

Authz Strategy Summary for R-GMA

- Authz decisions all made within Service
  - Use edg-java-security authorization manager
- Publish Policy in Registry
  - Allows ability to only ask producers questions they are likely to answer. (Mediator Functionality)
  - Mediator can make a first decision e.g. re-formulate a request
  - Which means that the mediator can have non-confidential authorization information on general policy
- All R-GMA Servlets must abide by the policy
- Use Delegated VOMS proxy's
- Final Authorization Decision made by the producer of the information
- Need to extract DN, VO, Groups and Roles

1st Step for Auth

- Work out how to setup the schema such that authz rules may be defined by the producer in the Registry
  - Include allowing a flag for `only if Authz request'
  - Include Authz for access to the info on producers.
- Implement enforcement within R-GMA ensuring that the Authorization rules are obeyed whenever data is passed on
  - Suggest that if a general request is made, producers supply what the principal is authorized to receive – request should not fail if there is some info that the user is authorized to receive.
  - Should there be a flag the user says `info available but the user isn't authorized' (I think not). If there is, it should be possible to turn it off.
- Setup to trust all R-GMA servlets and test

2nd Step for Authz

WP3

- Proper Authorization of the R-GMA servlets
  - `Mutual Authorization'
- This means we need at least a `host name verifier' and provide a list of trusted hosts for passing info to.
- Alternatively, could improve on the `which site is authorized to register' and ensure it is secure
- Eventually, VOMS service cert should be used – but this depends on it being O.K. to authenticate a service with a Proxy – or further developments of how we use 1 cert for auth and another for authz.

Confidentiality

- There are certain requirements on confidentiality. To satisfy these an authorization decision at the source of info AND a delegated VOMS proxy is needed.
- If a third party can say ‘tell me if Linda is banned’ without the use of a delegated certificate – then the fact Linda is banned can be found out without Linda’s permission.
- Similarly for any info – a hacked or rogue R-GMA can get any info they want. Can only make things difficult.

3rd step for Authz

- Allow the possibility of producers only putting info into R-GMA if it has been requested by an authorized principal
  - Delegated VOMS proxy
- This paves the way for the possibility of only putting information into the system if that specific information has been requested using a proxy where the principal has signed a request for that specific info.
- Without a delegated VOMS certificate – authorization is not very secure – any hacked consumer can do what they like.

4th Step for Authz

- Mediator only makes requests that are likely to succeed.
- Mediator re-formulates more general requests to only request what will succeed

Later on...

- Encryption?
- Only allow info into the system if it has been specifically requested by an authorized principal?
- Only allowing information to be passed straight from a producer to the authorized principal?

Some Other WP3 specific requirements

- It must be possible to restrict knowledge of the existence of producers of information to specific authorized users
  - Solution – authorization necessary to obtain information from the registry – only those authorized are granted info on the producer
  - Need to consider this when defining the Authz schema
- A producer must be able to restrict the publishing of information to specific authorized users.
  - Need Authorization on Registry information

Other WP3 requirements - contd

- A user can only see certain information on their own job
- A producer must be able to restrict read access to information to specific authorized users.

These are are covered by the basic authorization planned for R-GMA

Other matters

- Need to look at OGSA security – see how we can fit with this
- Including OGSA Authorization WG