GFD-I.032
SEC SAAAR-RG

October 13, 2004
Shawn Mullen, IBM
Matt Crawford, FNAL
Markus Lorch, VT
Dane Skow, FNAL

Site Requirements for Grid Authentication, Authorization and Accounting

Status of This Document:

This document provides information to the Grid community. It does not define any standards or technical recommendations. Distribution is unlimited.

Copyright Notice:

Abstract:

The purpose of this document is to collect requirements from existing grid resource sites with respect to the management of security for their grid services. Where those requirements are non-uniform, or even mutually exclusive, we recommend hooks which grid toolkits or applications should provide for the sites to insert their own implementations of their requirements.

This document is an informational GGF document which grid application and library coders can use as a reference guide. It also contains suggestions for future grid development work. The reader is assumed to have an basic understanding of common security terminology and current grid security infrastructure.

Requirements expressed in this document are not binding on grid developers nor define any compliance standard. The normative language is used to express the degree of importance assigned to a requirement on toolkit or application acceptance. Since there are few absolutes in application selection, the term "must" should be interpreted as indicating a requirement which if not met would cause

serious impediment to acceptance. Requirement expressed with the term "should" indicate desirable, but not essential, requirements.

Index:

Chapter 1 Site Authentication Requirements

1.1 Terminology and definitions

The following terms are used in this document as described here.

1.1.1 "User secrets" refers to values intended to be known only by the user, known by the user and an authentication infrastructure, or known only to an authentication infrastructure and employed on the user's behalf after the user has authenticated with some other secret(s).

1.1.2 To sidestep such questions as whether "a day" means eight hours or 24 hours and just how long a month is, we will deal in seconds but not quibble over implementation variances at the 10% or 20% level.

1.1.3 Credentials are assumed to have lifetimes which bound their period of validity. "Long-lived" credentials have lifetimes of 1,000,000 seconds (1 megasecond or 1 Ms) or more. "Short-lived" credentials have lifetimes of 100,000 seconds (0.1 Ms) or less. Lifetimes between those limits are "intermediate." The terms long-lived and short-lived may also be applied to the secrets employed by a user to acquire credentials, although the only short-lived user secrets known to be commonly employed are one-time (or "single-use") authenticators.

(Conversions: 0.1 Ms is a bit more than a day; 1 Ms is a bit less than 2 weeks.)

1.1.4 If a credential's lifetime can be extended by the user, using no more proof of identity than the credential itself, this is considered "renewal" of the credential, while if the process of extending the lifetime requires measures equivalent to those employed in its initial acquisition, we consider the result a new credential.

1.1.5 We specifically do not consider "post-dated" credentials -- those with lifetimes that begin at some point later than the time of the authentication act. Neither do we consider the relative strengths of cryptographic protocols, algorithms, and key lengths. We assume they are always designed, selected and implemented appropriately.

1.2 Identity

1.2.1 Sites will generally make authorization decisions on an aggregate basis: on Virtual Organization (VO) membership or group membership. However, at times it will be necessary to set access rights at the granularity of a single user. Sites must reserve the right, and preserve the ability, to set authorization at this level. Also, incident handling requires the ability to identify the legitimate owner of credentials presented during transactions under investigation.

Accordingly, every set of authentication credentials should be tied to the identity of an individual, because this provides stronger security by way of audit ability, revocation, and problem determination. However, there may be occasion to forfeit these benefits in order to provide temporary and generic identities.

For example, an Internet cafe could provide temporary (very limited lifetime) credentials authorizing use of grid resources based solely on the fact that access was purchased. Such an identity may be a psuedonym such as "Customer 24."

Other, similar identity indirections are expected:

- action traceable to a specific organization within a specific VO
- action traceable to a specific VO
- action purely anonymous

1.2.2 Secure anonymous communications may still be allowable, and appropriate, for functions that do not require user authentication.

For example, in the case above of cafe access to Grid resources; the user may still require secure conversation because the results of the data derived may have some proprietary value.

1.3 Assurance

1.3.1 An authentication system may provide multiple methods for a user to perform their initial authentication, and these methods may differ in their convenience, resistance to attack, and risks of exposure of secrets. Even when an implementation offers its users only one method, it may not be clear to relying parties which method it is. Since some inverse correlation does exist between convenience and strength of authentication, there may be inducements to allow and employ multiple levels of authentication if sites make some class of services available through weaker but less burdensome authentication methods.

1.3.2
We define three levels of authentication strength:

Strong - long-lived reusable secrets are not transmitted over the network.

Encrypted - long-lived reusable secrets are transmitted on the network in encrypted form. The encryption techniques (including key management) MUST be of sufficient strength that secrets are unlikely to be recovered by a hostile party before their expiration.

Cleartext - reusable identifying information (it would be an exaggeration to call it a secret) is transmitted in the clear. Cleartext authentication is considered equivalent in strength to no authentication at all.

1.3.3 We recognize the following modes of storage of users' long-term secrets, each with its own set of vulnerabilities:

1.3.3.1 What you know
Mental - secrets are held in users' own memory (PIN or password).

1.3.3.2 What you have
Secured - secrets are stored in electronic devices with credible protection against disclosure to unauthorized parties, even in the event of user carelessness.

Stored - secrets are stored in electronic devices in a manner that relies on users' willing diligence in protecting them against disclosure e.g. Biometric, or smartcard.

1.3.4 It is not possible to give a strict ranking of storage modes discussed section 1.3.3 relative to safety without asking and answering a number of questions about the details of the secrets, their storage, and their registration as the users' authentication information. Also, users may perform unsafe actions (knowingly or unknowingly) which place their secrets at much greater risk of disclosure.

1.3.4.1
Authentication strength must be mechanically deducible from credentials. The method used to perform authentication should be deducible from credentials.

1.3.5 There are a number of cases where processes running on a machine need to authenticate to other processes. Automated processes may have to act as authenticated clients and users may wish to have automatic software ("cron jobs") that require automatic authentication. All of these should be somehow restricted such that theft of credentials from an individual machine does not easily permit their reuse elsewhere. In either case, secrets will be of the "stored" class and must be considered to be stored in cleartext form, regardless of any measures which obfuscate them.

1.3.5.1
Authenticated identities of automated client processes should include identification of the machine which is intended to have access to the authentication secret.

1.3.5.2
Authentication methods based on stored secrets should indicate the machine from which they were used. If they do not, then this information must be available in auditable records.

1.4. Lifetimes

1.4.1. All forms of digital credential in common use are subject to possible theft and misuse. The probability of such an event is monotonically nondecreasing with time. The countermeasures against eventual credential theft are expiration and revocation. Neither measure alone is sufficient to prevent all misuse, nor is the combination of the two.

1.4.1.1
User authentication credentials must not be valid for more than 1 Ms if there is no method for checking for revocation. User authentication credentials should be renewed or checked for revocation every 0.1 Ms.

1.4.1.2
Authorities issuing revocable credentials must publish the

procedures for initiating revocation. In the case of X.509 certificates, each revocable certificate should include a pointer to such procedures. These procedures must include the location and publication frequency of revocation information and an upper bound on the time required to act on a revocation request.

1.4.1.3
It should be possible for authority parties other than the credential issuer or the credential owner to initiate revocation, under some circumstances. (For example the authority that vetted the identity of the user.) The processing time bound above may not apply to third-party requests for revocation.

1.4.2 The lifetime of authentication secrets is a separate parameter from the lifetime of credentials.

1.4.2.1
User secrets stored mentally should have a lifetime of 50 Ms or less. Some environments or applications may demand shorter lifetimes, down to perhaps 10 Ms. These times may vary depending on the strength of the password enforced by the password requirements of the system.

1.4.2.2
Secured user secrets may reasonably have lifetimes of 100 Ms or more depending on the securing technology.

1.4.2.3
Stored user secrets should not be valid for more than 1 Ms, and if valid longer than that, their associated credentials must declare that fact.

1.4.2.4
The above lifetimes are relevant to both the strength of the password and the strength of the crypto-analysis or password cracking tools. These lifetimes should be adjusted to reflect the current state of the art in these two related technologies.

Chapter 2 Site Authorization Requirements

2.1 Terminology

Terminology used in this document strives to be consistent with that used in the Authorization Frameworks working group.

2.1.1 "User" is a synonym for end entity and for subject used in the more general framework document. We preserve the use of "user" since it is more widely used within the site operations community.

2.1.2 "Groups" refer to groups of end entities which are accorded equivalent rights for purposes of obtaining a particular set of privileges.

2.1.3 "Role" refers to the set of attributes an end entity is presenting with a particular request for obtaining or asserting a privilege.

2.1.4 "Provenance" refers to information about the history of a request. For example, the identity of the original requester.

2.2 Authorization Process

2.2.1 The authorization process must be consistent within a VO. The process may have several steps (e.g. user authorization, VO authorization, site authorization, resource authorization) with various implementations. Users and VO managers must be able to rely on consistent interpretation of their policies.

2.2.2 The Virtual Organization must be able to decide user membership policy and user authorization policy.

2.2.3 The authorization method must be application independent.

2.2.4 Mutual authorization may be required.

An application or end entity may need assurances that the resource is authorized to run a specific job. The distributed program or grid job in and of itself may be of value. The results may be of value and need protection from dubious resources.

A grid job may need to specify that it is only run on systems with security level B operating systems, or systems not directly connected to the Internet, or some other operations requirement. This is more relevant in the OGSA model where service factories may incorporate more resources to handle service request loads.

2.2.5 Maintain Provenance

The authorization mechanism must preserve the Subject Identity of the user who originated the request.

2.2.6 Provide for method of grouping users

It should be possible to assign a user to a group. The authorization of resource access can be managed by managing permissions of the group.

2.2.7 Authorization Level Dependent on Authentication Strength

The authorization for access to a resource at a particular level may depend on the strength of the authentication. The level of authentication must be included with the credential information presented to all resource managers.

2.2.8 Call-outs

Call-outs prior to access to resources may be provided as a form of authorization control by the virtual organization, the site(s) and each resource provider.

2.2.9 Revocation

There must be the ability to quickly revoke a particular remote authorized service that may be operated under dubious procedures. The timescale for this revocation should be of order 0.1 Ms.

For example, if a remote processing resource steals computation results, it should be removed from the directory of processing resources. This is difficult in the context of the current Grid technology because of the open resource registration process and aggressive discovery algorithms. Similar such directory services on the Internet have a history of exploitation.

2.3 Authorization Attributes

2.3.1 Attribute Authorities

In expected grid operations, authorization attributes are generated by authorization servers run by VOs, by sites or other authoritative entities. These authorization attributes may contain specific authorization privileges, authorization to act in a particular role, or may contain statements of membership in a particular group within the VO.

2.3.2 Numbers of Attributes

2.3.2.1

Users or end entities may have any number of roles within a given

Virtual Organization. Whereas VOs may choose to structure themselves and express authorization policy in an arbitrary form, resource providers need appropriate mechanisms to enforce that policy in the local authorization infrastructure. Current uid/gid mapping mechanisms may become unwieldy when used to express possible combinations of several roles.

2.3.2.2

Users or end entities may be members of any number of Virtual Organizations.

2.3.3 Currency of Membership

Assertions of membership in roles and groups within a VO must be able to be validated by relying parties. Validation of such assertions should not succeed more than 1Ms after an authority removes the subject's membership.

2.3.4 Resource Administrators Authorize by Groups and Roles

VO attributes describing the roles and groups must follow a published standard, agreed upon at least within the domain of the VO. This consistency gives the Authorizer or Resource Administrator a manageable and trusted view of the membership pool. The administrator must be able to trust the concurrency of the roles and groups. This removes the need for Authorizer to have an understanding of each member. The Authorizer needs to only understand the groups and roles within this assigned membership pool.

2.3.5 User Selection of roles

A user must be able to select and de-select VOs and roles for a specific access. (analogous to the substitute user or 'su' command on UNIX systems, allowing an entity to change the current role briefly for a critical section before returning to a role and access privilege less vulnerable or potentially dangerous.)

In addition, a user should be able to individually define the set of privileges to be used with a specific service request. This allows for least privilege access tailored to the requested service and increases system security.

2.4 Policies

2.4.1 Authorization decision criteria

The owner of a resource or data must be able to allow or deny the authorization of an end entity to carry out an action using any of the following criteria:

1) none
2) having some acceptable authentication without specifying identity
3) membership in a VO
4) role(s) within a VO
5) a combination of memberships of VOs and roles
6) individual identity certificates
7) the presence/absence of specific authorization attribute(s)

2.4.2 Precedence rules for applying authorization decision criteria must be clearly stated.

2.4.3 Source of authorization also a decision criterion

It may be desirable for a resource manager to be able to disable access based on the source of the authorizations presented in case of compromise of a particular remote authorization service.

2.4.4 Combinations

The authorization method must allow any combination of the above authorization requirements, including any combination of VOs and roles (see requirement 2.3.2)

2.4.5 Authorization may be based on Operation criteria

It should be possible to base authorization on any of the following, in addition to the authorization requirements of section 2.4.1.

1) Resource namespace (e.g. file server, directory, filename, etc.)
2) Operation (including metadata and file operations)
3) Resource usage limits (E.g. quota)
4) Environmental data (e.g. time, current or anticipated resource utilization)

2.4.6 Granularity of Authorization

Depending on the application scenario, the granularity requirement for authorization decisions vary from fine grain (e.g. based on individual subject, requested action, privilege restrictions, and assets involved) to coarser-grained authorization on the basis of groups or even sites. Support for role based access control mechanisms is specifically requested for future collaborative environments but may also be desirable for other grid systems.

2.4.6.1 Collections

There should be no restrictions on the degree/level of granularity of authorization. In particular, no hard-coded limits to how the granularity is set should exist. This should include, for example, allowing authorization to a hierarchy of directories, individual directories, or individual files. It may become burdensome on the resource to support a high level of granularity, therefore it is left to the resource to set a practical level of granularity collecting objects into manageable sets.

2.4.6.2 Catalog by user

It must be possible to determine the list of resources to which an end entity has access and what actions that entity is allowed to carry out in the VO(s) and role(s) set for the current session. The burden of creating this list is on the end entity. It is left to the end entity to know or lookup or discover the resource and query for access permissions. This relieves the resource from having to know how to report to the end entities. This also averts a security vulnerability similar to the historical NIS (Network Information Services) hack in which the complete access lists being pushed to slave servers were intercepted and exploited. It is recommended that resources reveal access permissions only to the authenticated entities that hold these permissions and to administrative entities. (see 2.4.6.3)

2.4.6.3 Catalog by role

It must be possible to determine if a role or group has access to a resource. This access information is necessary to accurately stage and schedule jobs. This access information is sensitive because it could be used to exploit the Grid's security. For example, knowing that Bob has access to the targeted resource, the hackers attention is turned to Bob or his home computer.

Therefore, the following access levels are needed: A resource's access information must be accessible in its complete form to the administrator of that resource and security personnel for security audit and forensic purposes. Authenticated users may have information about all accesses he/she is allowed on that resource using the asserted identity and authorizations. Others must have access to authorization data only in the form

1) permit and permit qualifier (e.g. PERMIT/always or
PERMIT/8:00am-5:00pm)

and/or,

2) denied and denied qualifier (e.g. DENY/always or
DENY/QoS load).

2.4.7 Authorization control points

Control points must exist to allow for enforcement of authorization decisions and the inclusion of local policy decision functions. Management of these control points should not place a large maintenance demand on the resource administrator.

2.4.8 Authorization Policy Change Control

2.4.8.1 Policy coherency tools needed

Authorization policies may change over time. Mechanisms to manage policy specification across the administrative domain of the resource, site, VO, application manager, and user should be provided.

2.4.8.2 Timely updates of policy needed

A time delay between publication of a policy change and implementation or enforcement is to be expected. There should be prompt implementation of policy change. The resource manager will implement the policy change and log compliance. The resource manager will define a prompt and reasonable time delay appropriate for the resource. Policy changes may require verification and validation before deployment.

2.4.8.3 Suspension of privileges should not delete policy

Sites and virtual organizations should have the ability to suspend resource authorization for a particular grid identity without actually deleting the authorization and therefore possibly losing tracking information.

2.5 Transparency

2.5.1 Directory of user's roles

VOs should provide a method providing membership and role/group information for a given user. An example of this might be extended attributes within the users proxy certificate.

2.5.2. Transparency of Authorization information and policy

Certain groups or roles may require additional authorization before membership information is released (so as to not leak information about which accounts are privileged).

2.5.3. Protection of Authorization Info

Alterations of the information should only be possible through secure, authenticated access paths using procedures such that the sites are willing to trust the role / membership information returned. This requirement may involve a detailed description of how virtual organizations maintain and protect this data. (Similar, perhaps to a Certificate Policy / Certification Practices Statement for Certificate Authorities.)

Current proxy certificate specifications ensure that proxy and delegation operations never require private keys to be sent across the network. It is important to state clearly to developers that all future protocols must continue this practice. If it is necessary to send a passphrase or password across the network, they need to be encrypted at a strength equivalent to the strength of the key.

2.5.4 Dynamic Revocation of authorization

There is a dynamic nature to authorized access in that it may depend on the resource load, quality of service, or time of day. If authorization access changes during access, an error code should be propagated back to the application or the application should query for the authorization deny qualifier.

2.5.5 Standard Error Codes

The consistency and transparency to the application is aided by the use of standardized error codes of authorization denials. The error information should not provide more information than necessary, lest it create a security risk. An error return code may be accompanied with a log entry number to assist the resource administrator in synchronizing the denial instance. For example, a user may call a helpdesk to report access problems, giving the error code and log entry number. The resource administrator can reference this log entry number to provide detailed information.

2.5.6 Role Confirmation

2.5.6.1 Trust Model

It must be possible for the resource to confirm that a user has the VO membership(s) they claim. This is done through the trust model with the authority vetting the identity of the user. This is described in the "CA-based Trust Model for Grid Authentication and

Identity Delegation" from the GGF Grid Certificate Policy Working Group.

2.5.6.2 Timeliness

It must be possible for the resource to confirm the user's claimed role(s) or group membership at the time access to a resource is requested. For example, in the Globus environment, resources assign these groups via the grid-mapfile.

2.5.6.3 Privacy

It must not be possible for unauthorized users to produce a list of members of a VO, or the list of VOs to which a user belongs. Authorized VO administrators may have access to the full list of members.

2.6 Operations

2.6.1 Logging

Logs documenting the resource access decisions, policies, policy changes, and resource implementation of policies should be kept. The virtual organization, site(s) and resource managers should log such events and retain these logs for 10Ms (approximately 4 months). The logs should be protected to ensure privacy and integrity.

2.6.1.1
Logs should be frequently archived on a machine different than the one on which they were generated.

2.6.1.2
When archived, the logs should be digitally signed by the archive server.

2.6.2 Revocation

2.6.2.1
It must be possible for the authorized administrators to revoke all of a user's authorizations based on VO membership by removing the user from the VO.

2.6.2.2
It must be possible for the authorized administrators to revoke a user's authorization by removing the user's ability to claim a given role, a number of roles, or other attributes issued by an authority.

2.6.3 Revocation Timeliness

Authorization revocation should be done in a time frame consistent with the authentication revocation of 0.1Ms.

2.6.4 Fault Tolerance

Grids should gracefully survive partitioning so that local services can continue their operation in case a resource is disconnected or to avoid a DoS attack. This may require redundant or distributed Authorization Services.

2.6.5 Providing credentials to service

The authentication and authorization credentials that a user presents should be made available to the execution environment by something like a gatekeeper or job manager. In other words, the gatekeeper may have passed a request based on the presented credentials, but if this results in delegation of the request (e.g. running a job) the authentication/authorization credentials should be made available to the final execution environment via some standard mechanism.

2.7 Authorization for Replicated Data

2.7.1 Dependency on unreplicated authorization service.

If files are replicated, authorization for access to this replicated data should not depend on the availability of a single source of authorization. Simply put, the source site and the source site authorization server can go down without effecting access to the replicated data at other sites. Otherwise the service is not replicated.

2.7.2 Consistent authorization on all replicas.

The authorization requirements on data access should be consistently applied for all replicas of the same data.

Chapter 3 Site Accounting and Audit Requirements

3.1 Accounting and Audit Requirements Introduction

Accounting has historically had close ties to Authentication and Authorization because of the certainty with which they need to identify the entity to be associated with the accounting data. This is particularly important in the areas of security audits, intrusion detection, and computer and network forensics.

Accounting also has importance beyond accurate billing. IT management use accounting for controlling and managing operational costs. Accounting links to other IT disciplines such as capacity planning, service level management, and performance management.

3.2 Terminology

3.2.1 Grid Resource Accounting

Grid resource auditing is the more traditional sense of accounting that accounts for resources usage and billing.

3.2.2 Grid Auditing

Grid Auditing is the focus on accounting as a security component, and the need for a seamless relationship between accounting, and the authentication and authorization components of the Grid. Simply put, with a small addition to existing accounting data, an audit mechanism could greatly enhance Grid security.

3.2.3 Monitoring

The term "monitoring" refers, in the accounting and audit context, to the recording of transaction data. It is synonymous with "logging" in this document and does not imply timely human oversight.

3.3 Requirements Gathering

3.3.1 Requirements Gathering for Grid Accounting

Requirements for Grid accounting focus on the relationship of monitoring and metering authentication and authorization for auditing security. This information binds an end entity to the resource for the time and duration of access. The consumer of this information is Grid admin, helpdesk , intrusion detection or computer forensics.

3.3.2 Requirements Gathering for Grid Resource Accounting

It is important to understand how the audit data will be used. This will help define the accounting data gathered and the data flow. It is the goal of this document to describe the requirements of Grid accounting and audit components which satisfy a broad range of instances and usage. This chapter will also identify other current Grid working groups and accounting standards that are addressing these needs.

3.3.3 Non-Goals

This chapter will consider the consumers of the accounting data and their requirements, but will not analyze the consumers or make recommendations on how consumers should process the accounting data. It is not the goal of this chapter to reproduce or reinvent past accounting standards or duplicate current Grid accounting work.

3.4 Grid Auditable Data

The Grid auditing examines accounting requirements from a security perspective: audit logs, intrusion detection, and forensics. These requirements are not disjoint for mainstream accounting concerned with billing and metering, but in this section the requirements are described from the security perspective.

3.4.1

Grid Accounting must log the following data per resource access.

- -Resource
- -End Entity Identity and Provenance
- -Authentication and Authorization
- -Action Time and Duration

3.4.2 Resource Identification (RID)

The resource must be identified. The resource identity can be layered or accumulative or onion fashioned. This identification may be any or all of the following and more:

1) IP address
2) Web Service
3) vnode, or inode and generation of some other file handle
4) file set or disk volume group

The RID should be descriptive of the state of the resource. For example, if the resource is a file, the exact content of the file at the time of access would be an optimal piece of information for a forensic analogy. This type of metadata is difficult and expensive to maintain, and usually requires replay logs for the most accurate view of the data at and during the time of access. Nonetheless, the more accurate the accounting description of the resource, the more options are open for damage assessment and recovery.

3.4.3 End Entity Identification EEID

The EEID accurately describes the end entity to the resource. Commonly this will be a GSI proxy certificate, which can be traced back to a credential from some trusted source of identity.

There are a number of requirements related to the handling of the EEID.

3.4.3.1 EEID logging

3.4.3.1.1
Information tying the EEID to the processes executed on its behalf should be kept as part of the Grid auditable monitor data.

3.4.3.1.2
This data should not be recorded locally but should be reported to a remote central system.

3.4.3.2 The provenance of the process or job must extend to the true origin.

3.4.3.3 If a process inherits credentials beyond the subset of its current credentials, an alarm should be triggered.

An illustrative example specific to the Globus toolkit may help clarify the reasoning for these requirements.

Intrusion detection at a file system level when triggered identifies the PID (process id) of the offender. Via the system process table, the associated UID (user id) and PPID (parent process ID) can easily be identified. When a Grid job is submitted and runs on a Grid resource, the parent process is the UID mapped to the certificate in /etc/grid-security/grid-mapfile during the authorization process. Many certificates may be mapped to the same UID. This masks an audit trail needed to link all of the connections from the offending process to the EEID.

The two crucial pieces of information are the PID of the process running on the Grid resource and the EEID responsible for initiating this process. Both the PID and the EEID are known but not necessarily recorded consistently or together. The globus-gatekeeper will log the EEID at authentication time in the syslogd data.

For example,

```
   Feb 14 09:31:32 ipsec GRAM gatekeeper[29452]:
   Authenticated globus user:
   /C=US/O=IBM/OU=GridLPP/OU=austin.ibm.com/CN=shawnm
```

In this example, the EEID can easily be tracked via the CA and RA back to a singular user. The disjoint occurs with the recording of the PID of the actual process that is run on behalf of the EEID on the Grid resource. The PID is returned to the initiator in the form of a JobID.

For example,

```
   % globus-job-submit ipsec /bin/ls ls /tmp
   https://ipsec.austin.ibm.com:62960/27126/1045236692/
```

The middle number is the PID of the 'ls' command run on the Grid resource ipsec.austin.ibm.com. The JobID, which contains the PID, and the EEID should be sent as part of the Grid auditable monitor data. This data should not be recorded locally because it allows a hacker a means to cover his tracks. All Grid data should be reported to a remote central system. The provenance of the process or job must extend to the true origin. The GSI model allows for the propagation of jobs and the inheritance of security credentials. Simply put, as a job propagates from Grid resource to Grid resource, EEID must remain consistent or any transition of identity must be logged.

3.4.5 Authentication and Authorization

Knowing the provenance of a job should allow the audit trail to quickly discern the authentication and authorization used to gain access to the Grid.

Again, in the example of the Globus Toolkit.

The EEID or proxy certificate is logged by gatekeeper on the Grid resource. This is a logging of the authorization processes. The actual authentication took place on the provenance node with grid-proxy-init when the passphrase was entered and the proxy certificate created. The authentication process should be logged. Currently it is not possible to distinguish between a valid authentication via grid-proxy-init and the stealing of the proxy certificate out of /tmp.

This is analogous to the "su" command (substitute user) which is logged by syslog and in sulog. When the grid-proxy-init command is issued the user is taking on the identity of a particular Grid user. This information should be part of the Grid auditable data.

3.4.6 Action, Time and Duration

This section will have some intermingling of the accounting requirements as they relate to security and to resource management. This is done to illustrate that the same accounting data is used for two very different purposes.

3.4.6.1 The attempted action of the process running on the Grid resource should be part of the Grid accounting data.

The action of the process may be attempted but unsuccessful or denied. As an example, consider failed su attempts or failed logins. Action attempts are critical for behavior-based components of Intrusion Detection Systems (IDS).

Alternately, failed actions may be a consequence of a resource shortage or outage. This is useful to track for diagnostics or dynamic resource management. For example, in the Open Grid Services Architecture (OGSA) model this information could be used to create an additional service factory.

3.4.6.2 The time and duration of the process running on the Grid resource should be part of the Grid accounting data.

The time and duration are critical to computer forensics, as they allow for the creation of a time line of activity. Action, time and duration are important to intrusion detection, On Demand or dynamic services, and autonomic or self healing services.

3.4.7 Grid Accounting and Audit Data Conclusion

In a Grid environment it is important to monitor a causally connected sequence of events. It is important to be able to traverse this sequence of events from authentication to action taken on the remote resource. The proper accounting data can enable intrusion detection, the detection of malicious behavior and provide security audit trail.

3.5 Requirements Gathering for Grid Resource Accounting

Grid accounting is closely affiliated with security, but the more traditional computer accounting belongs more in the GGF Scheduling and Resource Management (SRM) Area. Specifically, the Resource Usage Service Working Group (RUS-WG) is relevant. It is not viewed an abdication of responsibility to leave this section to other GGF working groups. It is viewed as an efficient means of coordination between different GGF groups.

3.6 Existing standards and practices

3.6.1 Accounting Institutes

We have not been able to find any standards from computing or IT accounting relating to traditional financial accounting or from other standard bodies such as Oasis or Liberty Alliance. In the IETF, work has been done in this area (but not necessarily relating to Grid) in the following set of IETF RFCs.

RFC3127
Authentication, Authorization, and Accounting: Protocol Evaluation

RFC2989
Criteria for Evaluating AAA Protocols for Network Access

RFC2977
Mobile IP Authentication, Authorization, and Accounting Requirements

RFC2975
Introduction to Accounting Management

RFC2906
AAA Authorization Requirements

RFC2905
AAA Authorization Application

RFC2904
AAA Authorization Framework

RFC2903
Generic AAA Architecture

RFC2866
RADIUS Accounting

IETF Draft on DIAMETER BASE Protocol
<draft-ietf-aaa-diameter-17.txt>
<http://www.ietf.org/html.charters/aaa-charter.html>

Of the RFCs, the reviewing author found the the RADIUS Accounting standard to be the most interesting, since the nature of securely logging onto a network via RADIUS is similar to the nature of securely logging onto a Grid. There is considerable work in this standard that may be leveraged in implementing a Grid Accounting standard.

4 Related GGF Documents

"Security Implications of Typical Grid Computing Scenarios", GFD-I.12

"CA-based Trust Issues for Grid Authentication and Identity Delegation", GFD-I.17

5 Security Considerations

This document lays out a number of usability requirements to Grid developers from the perspective of a site administrator. In some cases, implementing these requirements may lead to conflicts either internally or with requirements from other parties (eg. users). The implementor will have to resolve these conflicts as well as assess his/her implementation for robustness in the face of attack. The implementor will be responsible for identifying the chosen trade-offs.

Desires for control, particularly centralized control, often lead to designs with valuable targets of attack and/or single points of failure. The implementor should take particular care to identify and protect such elements. Deployers should identify control points, their proper use, the information they contain, and assign responsibilities.

Furthermore, centralized control can lead to loss of privacy and freedom of association. Implementors and deployers should carefully weigh their needs for identification and control with their needs for privacy and spontaneity.

Author Contact Information

Shawn Mullen
IBM
11400 Burnet Road ZIP 9551
Austin, TX 78758-3493
shawnm@austin.ibm.com

Matt Crawford
Fermi National Accelerator Laboratory
Kirk and Pine Streets, MS 369
Batavia, IL 60510
crawdad@fnal.gov

Markus Lorch
Virginia Tech University
Department of Computer Science, m/c 106

Blacksburg, VA 24061
mlorch@vt.edu

Dane Skow
Fermi National Accelerator Laboratory
Kirk and Pine Streets, MS 369
Batavia, IL 60510
dane@fnal.gov

Acknowledgments

The authors gratefully acknowledge the helpful input documents from the Particle Physics Data Grid (PPDG) and European Data Grid (EDG) efforts and from the ACE and AuthZ working groups of GGF. This document is the product of much discussion in the SAAAR-RG in GGF and benefits from a host of individual contributions.

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."