A Shibboleth-Protected Privilege Management Infrastructure for e-Science Education

J. Watt, O. Ajayi, J. Jiang, J. Koetsier, R.O. Sinnott

National e-Science Centre, University of Glasgow, UK
jwatt@dcs.gla.ac.uk

Abstract

Simplifying access to and usage of large scale compute resources via the Grid is of critical importance to encourage the uptake of e-Research. Security is one aspect that needs to be made as simple as possible for end users. The ESP-Grid and DyVOSE projects at the National e-Science Centre (NeSC) at the University of Glasgow are investigating security technologies which will make the end-user experience of using the Grid easier and more secure. In this paper we outline how simplified (from the user experience) authentication and authorization of users are achieved through single usernames and passwords at users' home institutions. This infrastructure, which will be applied in the second year of the Grid Computing module part of the advanced MSc in Computing Science at the University of Glasgow, combines Grid portal technology, the Internet2 Shibboleth Federated Access Control infrastructure, and the PERMIS Role-Based Access Control technology. Through this infrastructure inter-institutional teaching can be supported where secure access to federated resources is made possible between sites. A key aspect of the work we describe here is the ability to support dynamic delegation of authority whereby local/remote administrators are able to dynamically assign meaningful privileges to remote/local users respectively in a trusted manner thus allowing for the dynamic establishment of virtual organizations with fine grained security at their heart.

1. Introduction

One of the key factors crucial to the success of the UK e-Science programme, and Grid computing in general, is to provide an infrastructure that is easy to use. The uptake of large scale resources such as the National Grid Service (NGS) in the UK [1] is proving to be heavily dependent on how easily users can run their jobs, and on how much extra infrastructure is required to be implemented by them to integrate their applications with the Grid. Currently the end-user experience of submitting a job to a resource such as the NGS starts with the acquisition of a UK e-Science X.509 Certificate issued by the UK e-Science Certificate Authority (CA) [2] which is based at Rutherford Appleton Laboratory (RAL) in Oxfordshire, UK. This is a process which involves contacting a central certification authority which issues certificates binding a user's identity to the public key contained in the issued X.509 certificate. Currently the user is expected to initially request a certificate directly from the CA web interface, which will subsequently send the request to a locally delegated Regional (or Registration) Authority (RA) which will normally be a responsible and recognised individual within the user's organisation, but may be at a remote institution if the user's home institution has no registered RA. The user is required to meet the RA in person to confirm the certificate request, and provide a form of standard identification (driving licence, passport, staff card) to confirm their identity. Once their identity is established, the RA approves the certificate request, and the central CA is contacted again to issue the certificate.

This process may take several days, especially if the user has no RA close by. Also, the issuing of nationally recognised digital certificates carries with it certain protocols and best practises which must be adhered to if the integrity of the CA is to be trusted. This involves measures such as strong password encryption of the private key, appropriate backup of issued certificates, and the conversion of certificates to appropriate formats. The extra work involved in not only acquiring the

certificate, but also the time taken for the users to familiarise themselves with the technology of Public Key Infrastructures (PKIs) make this an undesirable course of action for bringing the Grid to users.

A key issue facing the wider less-IT focused research community is also the requirement for them to convert the certificate to appropriate formats understandable by Grid (Globus) middleware, e.g. through running commands such as: *$> openssl pkcs12 -in cert.p12 -clcerts -nokeys -out usercert.pem*

Such requirements are likely to dissuade less IT-savvy researchers from engaging – especially as openSSL is not commonly available on platforms such as Windows. We note that the UK Certification Authority now suggests for researchers with Windows based PCs that they can use a Windows OpenSSL based solution [3], but this in turn requires them to install and configure additional software etc. In some circumstances this is not possible, for example if they do not have sufficient privileges on their PC (root access etc) – a not uncommon practice in certain departments and faculties at Glasgow University for example. In this case the researchers will instead have to refer to a local system administrator to help with the installation and configuration.

Assuming researchers have managed to obtain a certificate which they have converted into the appropriate format, they are then expected to remember strong 16-character passwords for their private keys with the recommendation to use upper and lower case alphanumeric characters. The temptation to write down such passwords is apparent and an immediate and obvious potential security weakness. Alternative, judgement based personalised authentication schemes drawing upon user knowledge bases are one way in which such issues can be resolved but as yet still largely a research area [4,5].

This process as a whole does not lend itself to the wider research community which the e-Science and Grid community needs to reach out to and engage with. It is a well known adage that the customer is always right. Usability and addressing researcher requirements is crucial to the uptake and success of Grid technology. End user scientists require software which simplifies their daily research and not make this more complex. Given the fact that the initial user experience of the Grid currently begins with application for UK e-Science certificates, this needs to be made as simple as possible, or potentially removed completely.

Indeed it is our experience in the UK e-Science initiative and a multitude of associated projects that this "enabling" technology is forcing people to abandon the Grid in favour of their more familiar, decentralised applications.

This situation need not be the case however. Security technologies exist that can both strengthen Grid security to include authorisation and not just authentication, and tailor the security environment to be familiar to end users, e.g. based on their local institutional usernames and passwords.

2. Technologies

2.1 Shibboleth

Shibboleth [6] is an Internet2 project designed to implement an open, standards-based protocol for securely transferring user attributes between collaborating sites. It is currently being deployed in the UK academic community as an inter-institutional infrastructure to support existing local methods of user authentication. The core of Shibboleth is an implicit trust relationship between institutions within a federation. A federation is a collection of institutions which trust each other to authenticate their users properly, and is responsible for distributing the PKI components necessary to establish trust between relying parties. Grid authentication currently relies on the nationally centralised CA described above to confirm the identity of individuals, whereas Shibboleth delegates the responsibility of authenticating users to where they belong: the user's home institution. Educational institutions in particular have reliable and established methods of authenticating their students and staff, usually by the process of matriculation for students, or human resources registration for staff members. Institution IT facilities are usually available everywhere across campus, accessible by usernames and passwords that the student will use irrespective of location. Ideally there should be a single username/password but often a variety of solutions exist depending upon the technologies. We note that Glasgow University is in the process of rolling out a unified account management system where all accounts and associated services are centrally managed using active directory technology [7].

Shibboleth extends local authentication and authorisation to allow institutions within a federation to trust local infrastructures sufficiently to allow access to large-scale resources utilising local login information without the need for a centralised authentication

mechanism. Therefore, a student would login to a resource such as the National Grid Service using their local institution login, and through the federated trust model of Shibboleth, this authentication information is valid across the Grid (assuming that the NGS is in a federation with the students institution).

The architecture of Shibboleth defines several entities which are necessary to achieve this seamless integration of separate collaborating institutional logins. The main components of Shibboleth consist of an Identity Provider (also known as a Shibboleth 'Origin'), a Where-Are-You-From (WAYF) service, and a Service Provider (also known as a Shibboleth 'Target').

The role of each of these entities is best described by an example invocation of a service that is protected by Shibboleth. A user types in the URL of the service (or portal) they wish to access. We assume that the web server that hosts the service is protected by Shibboleth and is known as the Target. Any attempts to access this page will cause the Shibboleth web server plug-in to start and to instantly redirect the user to the WAYF service. The WAYF service is generally run by the federation that the institution is subscribed to, but can also be run locally. The WAYF service presents a drop-down list to the user that contains all the participating institutions (or projects) that are subscribed to within the federation. The user chooses their home institution from this list and is then redirected to the home institution's Origin. This is a specific Shibboleth resource based at the user's home institution that integrates Shibboleth with the existing local login mechanism. The Origin will provide an interface for the user to login as if they were at their home institution. If successful, the Origin passes the user back to the original Target they were trying to access, at which point the resource can decide whether or not to grant access based on the information passed to it by the user's home institution Origin. It is important to note that much of this is transparent to the end users: they attempt to access a Target, are redirected to where a WAYF where they select where they are from, and subsequently log-in to their local institution. The authentication information and any security attributes that are needed and/or allowed to be released occur transparently to the user. They are either authorised to access the Target (in which case the service is displayed to them) or they are unauthorised (in which case they will get an appropriate not-authorised 404 exception raised).

Shibboleth itself is deployed as a plug-in to Apache web server and is instantiated when a user attempts an HTTP request on a Shibboleth-protected directory. Assertions are exchanged between the Origin and Target using SAML v1.1 [9], which allows for secure transmission of authentication tokens and user attributes across the federation. Shibboleth allows for any authentication mechanism that can be interfaced with Apache, in our case we query a separate LDAP directory containing user login information and attributes using the mod_auth_ldap Apache module, although authentication methods such as .htaccess, or Kerberos may be implemented.

2.2 PERMIS

The PERMIS (Privilege and Role Management Infrastructure Standards Validation) [10] software realises a Role Based Access Control (RBAC) [11] authorisation infrastructure implemented in a Java API. It offers a scalable alternative to centrally allocated PKI certificates through the issuance of locally generated X.509 Attribute Certificates (ACs). PERMIS allows institutions to implement a Privilege Management Infrastructure (PMI), which enforces access control on resources based not only on a user's identity, but on specific privileges or roles that they have been allocated. Access control in most current Grid systems predominantly make use of the GSI [12] gridmap file, which is a list of all valid users on the system and their appropriate user accounts. Although simple to use, the Access Control List (ACL) approach of GSI faces scalability issues to administer and manage when the number of grid users scales up to hundreds or thousands. RBAC systems instead allocate roles to users based on their function within their institution (e.g. Lecturer, Student, Administrator). The identity of system users will change as new people are employed, and older employees leave, however the roles and subsequent privileges that these people hold will generally remain unchanged. So access control policies or lists do not need to be constantly updated according to changes in personnel, rather, users are assigned these pre-existing roles so they can be instantly absorbed into the PMI with no changes to the system itself.

The PERMIS system uses XML policies which define the rules by which a resource will grant its users access to its services. A PERMIS policy contains the following definitions:

- Subjects that may be assigned roles

- Sources of Authority (the local managers trusted to assign roles)
- The roles themselves and their hierarchy
- The resource targets and the allowed actions
- The actions each role is authorised to take

Roles are assigned to users by issuing them with X.509 ACs. The roles are allocated by a Privilege Allocator (now called Attribute Certificate Manager) tool which binds the role to the user, digitally signs the certificate and stores it in an LDAP repository.

The PERMIS API can be applied to any application, and through the SAML AuthZ API callout [13], can be used to provide a generic mechanism for linkage of Grid services and authorisation infrastructures. Currently versions of the Globus Toolkit since version 3.3 have supported this API. This API also allows bypassing of the relatively coarse-grained authorisation control of GSI.

A new tool for the final phase of the DyVOSE Project provides dynamic delegation of authority. This tool and its use in the project are described in detail in Section 4.

2.3 Grid Portals

In the same way a Web portal such as Yahoo or MSN aggregates site information, web pages, and indexes, a Grid portal provides a framework for providing single-point access to Grid services. A portal is a web application that runs in a Tomcat container, and a portlet is an application, e.g. a Grid service that is accessible within the portal. The JSR-168 specification [14] provides a framework and associated APIs through which portlets can be developed and successfully deployed across a variety of different portal technologies, and is exploited in our work.

A typical scenario is that the user navigates to the portal page, and then, based on their identity and the authorisation policies in place, the portal displays the relevant portlets that the user may interact with. In this way, a virtual organisation may be formed by aggregating required, separate Grid Services into one front-end web page.

Entry to a portal is generally provided by username and password. In the case of a Grid portal, this username/password information will be used to establish the user's identity and generate a proxy certificate for subsequent Grid job submission. It is therefore necessary for the portal to carry an Access Control List of its valid users. This presents problems similar to that which RBAC systems are meant to solve (i.e. scalability issues).

Several Grid Portal solutions are available. IBM's WebSphere [15] is a highly robust solution, however it makes use of proprietary software that is difficult to interface with newer applications. GridLab's GridSphere software [16] is an open source solution that can run out of a Tomcat container providing a portal that can be altered to interface with our own Grid based applications.

Numerous other portal technologies such as PURSE and GAMA are currently being explored at the NeSC. [18,19]

3. Background Projects

The JISC funded ESP-Grid project is investigating the extent to which Shibboleth can offer solutions to the unresolved issues regarding Grid security. The National e-Science Centre in Glasgow has the remit to build a demonstrator showing how Shibboleth can be used for secure access to Grid resources.

The Dynamic Virtual Organisations in e-Science Education (DyVOSE) [19] is a two-year JISC funded Core Middleware Program project which aims to investigate the use of advanced security infrastructures in the e-Science education domain. As part of the project deliverables, an Advanced MSc. Course Module in Grid Computing was written and delivered at the Department of Computing Science in the University of Glasgow at the beginning of the 2004/05 session. This course is also running again in early 2006, with expanded lab work.

In the previous teaching of the module, a thorough exploration was made of the PERMIS authorisation software for forming a static PMI in a grid infrastructure. Specifically, the students were asked to implement a GT3.3 Grid service which searched and sorted a text file (the complete works of Shakespeare) by submitting jobs to our local Condor pool. Once deployed, they were required to protect the service using PERMIS by implementing the correct callouts from GT3, and authoring a security policy in XML which would enforce access control to certain members of the class. In this case, two teams of students were created, with the goal of allowing all users to search the file, but allowing only users in the student's own team to sort it. Experiences in running the course, and knowledge gained from using the software was reported at CLAG 2005 in Cardiff [20].

The final phase of DyVOSE involves implementing simple dynamic delegation of authority. This involves creating an infrastructure together with the e-Science MSc course being run at the University of Edinburgh, using an extended version of the PERMIS software incorporating a Delegation Issuing Service (DIS).

In the rest of this paper we outline how we have exploited Shibboleth, PERMIS, Globus and GridSphere technologies to provide decentralised security, and outline our plans for exploiting the dynamic delegation of authority capabilities implemented in PERMIS in the upcoming Grid Computing module.

4. The Delegation Issuing Service (DIS) and Dynamic Delegation
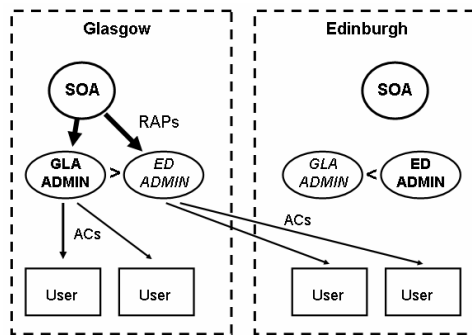
Figure 1: Diagram illustrating Simple Dynamic Delegation. Glasgow has issued a Role Allocation Policy to Edinburgh, allowing it to issue Glasgow roles (in ACs) to Edinburgh students (and vice versa).

The DIS (Delegation Issuing Service) [21] is a web service that can issue attribute certificates on behalf of the logged-in privilege holder (normally the Source of Authority (SoA)). The DIS service in this case is a Java component hosted in a Tomcat container running Axis SOAP server. In this case, the client is a trusted server acting as a proxy between DIS users and the DIS server itself. DIS extracts and writes the ACs to a dedicated LDAP server. In the static PMI model, any entity that issues roles or privileges to users must have a PKI key pair. This restriction implies that the SoA is responsible for all privilege allocations within the VO. In the Simple Dynamic Delegation case, the SoA has the ability not only to issue roles, but to delegate the right to issue roles to its subordinate entities. These entities are not required to hold a PKI key

pair in order to issue ACs to their subordinates however the SoA will by definition restrict the roles that its subordinate authority will be able to issue. This is known as issuing a Role Allocation Policy (RAP) to the subordinate. The RAP enforces the "top-down" security architecture so that subordinate authorities will always have lower privilege than their superiors. Using the fact that multiple SoAs may be trusted in the local security policy in the static PMI case, we can extend this idea of Dynamic Delegation using the DIS across institutional boundaries. This is illustrated in Figure 1.

A static PMI has been implemented at the University of Glasgow as part of the first year of the Grid Computing module. It consists of a SoA which has the job of issuing ACs to Glasgow users. The SoA holds an X.509 certificate which is the root of trust for the Glasgow PMI and it is this certificate that signs subordinate ACs. The SoA by definition can assign any role to any user, but within the context of the Glasgow PMI has two roles available: *studentteam1* and *studentteam2*. The PMI is protecting a Grid Service which grants access to users holding one of these roles, and denies access to the users holding the other role. Any role other than these two will not be accepted by the local XML policy and therefore no access will be granted. As a final check, the role must have been issued and signed by a SoA trusted in the XML policy which for a static PMI is simply the local SoA.

As part of the second year of the Grid Computing module, the Grid Service was extended to grant access to a set of users at the University of Edinburgh, which is running its own PMI. The simplest approach in this case is to include the Edinburgh SoA in the trusted SoA policy statement, and tell the Edinburgh SoA that it may issue either *studentteam1* or *studentteam2* to its users. Edinburgh users wishing to use the Glasgow resource would then present their *studentteam* role when attempting access, the Glasgow PMI would see that this role was signed by the Edinburgh SoA and access would be granted as this SoA appears in the trusted SoA list of the XML policy. This approach is undesirable as the Glasgow PMI is, in effect, allowing an Edinburgh student to act as a Glasgow student and inherit all of the Glasgow user's privileges. The fundamental point to be made here is that it should never be the case that an external user should have the same rights (role) as a local user. This would undermine the local security policy and the integrity of the PMI.

Another approach is to use Dynamic Delegation. In this case a third role would be defined alongside the *studentteam* roles. In the static PMI model, this could be achieved by adding the Edinburgh SoA to the policy trust list, and adding an Edinburgh role to the policy using whatever name they have chosen, e.g. "EdStudent". This again is an undesirable situation as we are now hard-wiring roles from another institution into our local security policy.

A more secure way is to define a Glasgow role that exists for non-Glasgow users. We simply called this role "external", and it is defined in the XML policy as being subordinate to the *studentteam* roles. This role is to be granted to users from Edinburgh (or anywhere else in the VO) who wish to use the Glasgow resource. Through the hierarchy of the XML role policy, any privileges that the external role holds will be inherited by the *studentteam* roles, yet the *studentteam* roles will have higher privileges which "external" cannot inherit. This hierarchy allows distinct levels of trust to be implemented based on a user's function and location within the VO without surrendering local policy integrity. Edinburgh is now free to assign this role of external to any of its users who wish to access the Glasgow resource. Although this sounds better than the previous model, there is still the issue that we are freely allowing an external (albeit, trusted) entity to allocate a Glasgow role to its users. Obviously we would like to retain more control over the issuing of this role by external institutions. This is the job of the Delegation Issuing Service, the GUI for which is shown in Figure 2.
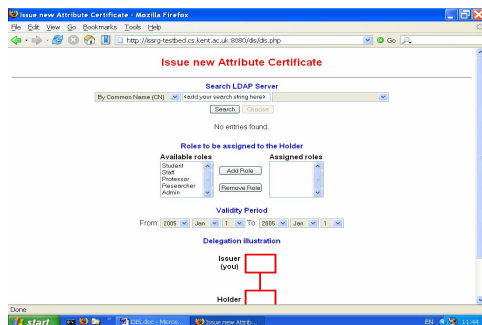
Figure 2: The Delegation Issuing Service (DIS) user interface.

The DIS allows us to realise a Simple Dynamic Delegation infrastructure by providing a framework within which a home PMI may securely delegate some of its administration to external entities. Considering our implementation above, the Glasgow PMI has defined a subordinate external role in its policy

and has added the Edinburgh SoA as a trusted authority. The Glasgow SoA creates an empty Edinburgh Admin entry in their LDAP server then logs in to the DIS service as its "root" user. The Glasgow SoA issues an X.509 AC (effectively a RAP) to the local Edinburgh Administrator entry stating that they have been delegated the authority to grant the "external" role to any of their users. The Edinburgh Administrator user (the Edinburgh SoA) can now log in to the DIS server where they will be allowed to issue ACs containing the external role, but no other privileges, to Edinburgh users. These ACs will be valid for access to Glasgow resources as they have been created through the constraints of delegation and the Edinburgh SoA is trusted in the XML policy. There have been resources set aside in Edinburgh to enable "two-way" delegation through an identical setup in Edinburgh, with Glasgow students being assigned the "external" role.

Note that in the current software implementation, it is necessary for these Edinburgh ACs to be held in the Glasgow LDAP server. This limitation is being addressed by the next version of the PERMIS software. The subset of users that the RAP applies to may also be constrained by a Target Access Policy (TAP) which states further constraints on the privileges delegated down to subordinates. This software should be available in mid-2006 [22].

As far as the student end experience goes, the search/sort Grid Service will be identical to the static PMI version. All the changes are done at the administrator level however the students will have to include the definition of the function of the "external" role in their XML policy. This is, as of now, not finalised, however the external role could be offered a restricted number of invocations of the Grid service or some similar constraint.

5. Implementation Issues

To explore the integration of Shibboleth and associated Grid security technologies, we have initially focused upon the Shibboleth enabling of the previous Grid Computing students project assignment. The software infrastructure to realise the protected Grid Portal was built first on the lab machines used by the Glasgow MSc students, then deployed to production machines. For hosting the Web Portal and the Shibboleth Service Provider software, a platform of Fedora Core 1 was chosen. This OS kernel is compatible with GT3.3. The Shibboleth Identity Provider was deployed using Fedora Core 4. This choice of

operating system was made to facilitate easy implementation of OpenLDAP, which requires the latest version (2.2.26) for the PERMIS DIS to work. Also, the versions of Apache Web Server, PHP and mod_auth_ldap are compatible with the DIS requirements.

The connecting of Shibboleth to GridSphere can be done in several ways. Ideally, we would like Shibboleth to protect the directories that GridSphere uses for its dynamically created portal pages. Since Shibboleth is an Apache module, and GridSphere is deployed in a Tomcat container, the use of a connector such as mod_jk2 was a desirable (yet potentially fraught) solution. It turned out that the implementation of the Apache/Tomcat connector was quite straightforward. Shibboleth was configured using the Location tag in httpd.conf to protect the /gridsphere directory that GridSphere writes its pages to. Appropriate firewall changes were made so that Shibboleth was the only way to access the portal, as without it, it is easy to point directly to port 8080 and find the portal front page.

Shibboleth provides user information through HTTP headers. Since GridSphere uses its own login system, we needed some way to turn off this generic login and use the user information that Shibboleth provides. The login portal for GridSphere was switched off, and a small PHP script was used to parse the HTTP headers for Shibboleth information. In order to be compatible with PERMIS, a new schema was defined that passes the attributes "attributeCertificateAttribute" (the signed AC), and "permisRole" which is just a text version of the unsigned role and was used for test purposes. The user's name was passed to the portal as well to provide a personalised login, although bearing in mind the Shibboleth trust model, releasing user's identities is fundamentally optional, and perhaps undesirable in a highly secure system.

GridSphere now holds the role of the user currently logged in. It is this role that is presented to the PERMIS API (operating in push mode) when making the access decision. But in our example, Gridsphere only shows the services they is allowed to use according to the site policy, so it would be impossible for them to even attempt to invoke a service they are not authorised to.

Figure 3 shows the GUI and the Shibboleth attributes that have been returned. Here we retrieve the distinguished name of the user and the attributes indicating which role(s) the person has in this federation. These attributes are dynamically retrieved by the Shibboleth

infrastructure and used by the local security infrastructure to allow or deny access to the resources, i.e. the Grid service portlet.

Figure 3: A Shibboleth-Protected GridSphere portal displaying the user attributes received by the Identity Provider

This infrastructure has successfully demonstrated how dynamic delegation and recognition of attribute certificates can be achieved. The design of the next Grid Computing module main programming assignment is currently under development. The scenario we are expecting the students to support will be based upon inter-institutional resources at Edinburgh and Glasgow and associated security policies (utilising the delegation issuing service) at Glasgow and Edinburgh. In detail, we expect to ask the Glasgow students to develop a Grid service that can utilise the Condor pools at Glasgow and Edinburgh, but that the priority of their jobs at Edinburgh will be lower than their job priority at Glasgow, due to their external role at Edinburgh having a lower privilege. The application / Grid service itself may well be based upon getting the students to crack a 40-bit Data Encryption Standards (DES) key. Whilst it is not our intention to educate students to misuse Grid resources for key cracking, we think that the security flavour of this problem is in suited to the general project of Grid security which we have been exploring in DyVOSE.

6. Conclusions and Future Work

The scenarios implemented thus far in DyVOSE and extended as part of the ESP-Grid project have illustrated how a simple GT3.3 service deployed in the GridSphere portal can be protected by the Shibboleth infrastructure with fine grained authorisation access control performed by the PERMIS software. This GT3.3 service uses the Condor Pool for its job submission, and does not rely on proxy certificates for propagating jobs on the Grid. Further work for ESP-Grid will integrate a proxy certificate solution using MyProxy, which should allow job submission to large scale resources where user identity and proxy certificates ala classic Grid security solutions can also be supported. The use of eduPerson standard attributes for Shibboleth will be investigated, as we have tailored our own attribute set for the work described above.

The PERMIS dynamic issuing service has allowed us to implement a simple dynamic delegation PMI between Edinburgh and Glasgow on this Grid Service. This service was deployed in a GridSphere portal and, using the Apache JK2 connector, was protected by the Shibboleth Apache module. Shibboleth provides user attributes to the PERMIS software when a service is invoked, doing away with any access control lists on the resource side.

Thus we have demonstrated that Shibboleth can be utilised in a grid context, and that simple dynamic delegation can be achieved between two sites in a VO. This model where fine grained security infrastructures can be seamlessly linked has numerous application areas where the existing weaker authentication-based security models is not suited, e.g. the e-Health and commercial domains.

7. References

[1] UK National Grid Service http://www.ngs.ac.uk/

[2] J.Jensen, "The UK e-Science Certification Authority", Proceedings of the UK e-Science All-Hands Meeting, Nottingham, UK, September 2003

[3] OpenSSL for Windows Platform (http://www.openssl.org/related/binaries)

[4] J. Liddell, K.V. Renaud, A. De Angeli, "Authenticating users using a combination of sound and images", HCI 2003, Bath, UK, September 2003

[5] K. Renaud, "Quantifying the quality of web authentication mechanisms: a usability perspective", Journal of Web Engineering, 3(2):95-123, 2004.

[6] Shibboleth, http://shibboleth.internet2.edu

[7] C. Rusbridge, "Authentication and Access Control – final report", (http://www.gla.ac.uk/infostrat/wgroups/wg1/iswg-1-frep.html)

[8] Shibboleth Architecture Technical Overview, http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-oberview-latest.pdf

[9] OASIS. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1, Sep 2003 http://www.oasis-open.org/committees/security/

[10] Privilege and Role Management Infrastructure Standards Validation project www.permis.org

[11] D. W. Chadwick, A. Otenko, E. Ball. "Role-based access control with X.509 attribute certificates", *IEEE Internet Computing*, March-April 2003, pp. 62-69

[12] Globus Security Infrastructure (GSI), http://www.globus.org/security

[13] D.W.Chadwick, A.Otenko, V.Welch, "Using SAML to link the Globus Toolkit to the PERMIS authorisation infrastructure". Proceedings of 8th Annual IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Windermere, UK, 15-18 September 2004

[14] JSR-168 Portal and Portlet Specification, http://developers.sun.com/prodtech/portalserver/reference/techart/jsr168/pb_whitepaper.pdf

[15] IBM WebSphere, http://www-306.ibm.com/software/websphere

[16] GridSphere, http://www.gridsphere.org

[17] PURSE – Portal based User Registration Service, (http://www.grids-center.org/solutions/purse)

[18] GAMA – Grid Account Management Architecture, (http://grid-devel.sdsc.edu/gama)

[19] Dynamic Virtual Organisations in e-Science Education project (DyVOSE) www.nesc.ac.uk/hub/projects/dyvose

[20] R.O. Sinnott, A.J. Stell, J. Watt, "Experiences in Teaching Grid Computing to Advanced Level Students", Proceedings of the 5th IEEE International Symposium on Cluster Computing and the Grid, Cardiff, 19th-12th May 2005

[21] D.W.Chadwick etc al, "Delegation Issuing Service", *NIST 4th Annual PKI Workshop*, pages 62-73, Gaithersberg, USA, April 2005

[22] OpenPERMIS, http://www.openpermis.org

Author/s:
Watt, J.;Ajayi, O.;Jiang, J.;Koetsier, J.;Sinnott, R. O.

Title:
A Shibboleth-protected privilege management infrastructure for e-science education

Date:
2006

Citation:
Watt, J., Ajayi, O., Jiang, J., Koetsier, J., & Sinnott, R. O. (2006). A Shibboleth-protected privilege management infrastructure for e-science education. In Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID), Singapore.

Publication Status:
Published

Persistent Link:
http://hdl.handle.net/11343/28870