Audited Credential Delegation: A Usable Security Solution for Grid Environments

Ali N. Haidar, Stefan J. Zasada, and Peter. V. Coveney*

Centre for Computational Science, UCL

TeraGrid '11: Extreme Digital Discovery
July 18-21, 2011, Salt Lake City, Utah, USA
Overview of this Talk

- Motivation
- ACD & Relevance to TG/XSEDE
- ACD Architecture
  - Components
  - Administration
  - Pattern of interactions
- Usability
- Future Plans
Motivation

- Usability of grid security mechanisms is essential for the uptake of any solution. Current solutions are based on PKI but are hampered by usability considerations
  - Process of acquiring X509 certificates is time consuming - at least 3 days
  - Configuring grid tools to use the certificate
  - Generating proxies prior to accessing remote resources.

- So users ignore grid or often share the private key of a single personal certificate to get on with their tasks. This
  - Weakens the security environments
  - Breaches the *de facto* security model
    » one-user one-certificate
What do scientists want?

- Run scientific tasks on distributed resources (i.e. High Performance Computers, Grids) and get the results of running these tasks

- Use familiar/usable security mechanisms to perform their jobs. The mechanism should
  - not to be a barrier to their progress,
  - be seamlessly integrated with their desired ways of working
What do Resource Providers want?

- Secure access to their resources:
  - Who is requesting access to their resource? **Authentication**
  - Is the user allowed to run tasks on their resources? **Authorisation**

- Find users responsible for running named tasks on their resources (**Auditing**) in case of misuse:
  - Security breach
  - Usage of CPU allocations /billing

- Avoid consequences of security breaches:
  - Negative publicity and fines
Current approach for a scientist to access grid resources

To access grid resources a scientist will need to:

- Acquire a digital certificate (average 3 days); (application, RA confirmation, issuing)
- Get authorisation from resource providers (TeraGrid, XSEDE, PRACE, DEISA, NGS); takes between 3 days to 2 weeks BUT only done once!
- Configure the client applications e.g. Globus Toolkit, UNICORE or AHE Client to access the grid with a certificate (exception web portal)
- All in all a lengthy and complicated process, deters many potential end-users from exploiting grid resources.

ACD: What is it?

- **Audited Credential Delegation (ACD)** is a security solution that provides:
  - Alternative authentication methods to certificate based approach. Current prototype supports username/password mechanism; Shibboleth is underway (expected release Oct 2011).
  - "Hides" X.509 digital certificates from end-user; it does not replace it
  - Authorisation based on parameterised Role Based Access Control
  - Auditing

ACD: Relevance to XSEDE (1)

- Currently implemented as a security extension to the Application Hosting Environment (AHE2.0) middleware: http://www.realitygrid.org/AHE/

- Tested and deployed on TG/XSEDE, NGS and DEISA as part of a comprehensive usability study

- Combination of ACD+AHE is currently on the Science Gateways list of XSEDE https://www.xsede.org/gateways-listing

- Enables construction of VOs that permit scientists to run pre-configured applications on remote grid resources using familiar credentials such as username-password.
ACD: Relevance to XSEDE (2)

XSEDE has several Gateway desirable characteristics that ACD+AHE meets.

- Usable security by supporting authentication via:
  - Local username-password (using best practices)
  - Shibboleth: e.g. US InCommon Federation (www.incommon.org) & UK Access Management Federation (www.ukfederation.org.uk)

- Developed in Java and Web services technologies open standards
  - Ease of integration into XSEDE infrastructure
  - Platform independent (Java)

- Flexible and evolvable: Service Oriented Architecture
  - E.g. Authentication as service; Authorisation as service
ACD: Relevance to XSEDE (3)

- Adopted the recommendations of the Open Web Application Security consortium (OWASP) for developing secure software. The implementation addresses:
  - known issues in username/password mechanisms (using hashes and salts) and weak passwords
  - SQL Injections
- Model of ACD has been developed based on formal notation, which is used for building safety critical systems.
  - Ensures correctness of the design and robustness against known vulnerabilities listed by OWASP.
- Tight auditing by recording users’ requests to perform tasks on grid resources to maintain accountability

ACD Usability Study

- **Usability:** we have completed a comprehensive usability study that involved:
  - Comparing AHE+ACD, AHE (GUI), AHE command line, Unicore and Globus
  - 40 users from different faculties (Physics, Computer Science, Medical school, Business School, Chemistry, Cancer Institute, Law School 😊)
  - Task: run a simulation on Grid (NGS) using the above middlewares and use credentials given to them (username/password, X509 Certificate)
  - **Result:** AHE+ACD scored best in respect of:
    » Time needed to run the task
    » Ease of Configuring the tool
    » Ease of running the whole task

Setting up virtual organizations (VOs) at a specific time.

Requires acquisition of certificate for the VO group in the usual lengthy way by an expert-user (the admin of the group).
ACD: Assign user to VO and roles instantly

- Grid Projects are referred to as VOs.
- Quick and easy user enrolment e.g. add user to one or more VOs;
Easier user management for administrators e.g. add/remove users from VOs; assign/remove roles to/from users; reset user passwords
ACD: Functionalities - Admin Screen

- Set up virtual organisations (VOs)
- Add/remove users to/from VOs,
- Renew/update certificate used by VOs
- Auditing issued proxies used to access grid resources
**Authentication Service**
- Local database
- Kerberos
- Shibboleth

**Auditing Service**
- Audit Log
  - Records who accessed what, when, from where, and outcome

**Credential Repository**
- Certificate → Key
- Proxy → Key
- Proxy → UserID
- ProjectName → UserID
- ProjectName → Certificate

**Authorisation Service**
- UserRole: UserID → [Role]
- RolePermission: Role → [Task]

**Construction of VO**
- [1] task, u1,pw1
- [2] record success/Fail
- [5] task
- [6] generate proxy
- [7] Upload proxy
- [9] Download proxy
- [10] proxy & run Task on grid

**AHE Server**
- Workflows
- Application 1
- Application 2
- Application n

**NGS myproxy Server**
- GRID
- NGS
- DEISA
- TeraGrid

**Note:** All communications between AHE Client, AHE Server, myproxy server and Grid resources uses HTTPS.

**Note:** Task, u1, pw1, record success/Fail, record Granted/Denied, task, u1, record success/Fail, record Granted/Denied, task, u1.
ACD Components

- **Credential Repository stores:**
  - Digital certificates associated with VOs and their private keys
  - Proxies issued to individual users (for auditing)
  - Existing VOs and their members

- **Local Database stores:**
  - Usernames and passwords for users who want to access grid resources.

- **Authorisation Component:**
  - Resource providers still have full control over their resources
  - Controls the operations of AHE and ACD interfaces using Role Based Access Control e.g. who can invoke submitting a job on grid resources from the VO; who can reset passwords on ACD; who can monitor jobs submitted and who can terminate a job.

A user “John” who wants to run an application on a TG resource issues a request from AHE+ACD Client. The request is intercepted by:

1. **Authentication component** to establish the identity of the requestor (e.g. valid username-password)
2. **Authorisation component** to check whether the user is permitted to perform the operation requested (e.g. member of UCL-TG VO)
3. **Auditing component** which records sensitive requests (e.g. running applications for billing)
4. If the request is authorised, a proxy is issued for the request on user’s behalf and the operation is executed on the TG/XSEDE, ..., resources
From XSEDE’s end, it is the named VO (UCL-TG VO) that submitted the task, not “John Smith”.

In case of misuse how does XSEDE know which individual was the holder of the proxy?

**AUDITING:** TG/XSEDE administrator passes the public key of the proxy to the AHE+ACD VO administrator who can identify the name of the user from the relation Proxy → UserID in the Credential Repository component. In this way, requests from within the combined ACD+AHE are audited.
Future plans

- Support for Shibboleth to enable authentication using local University credentials (expected Oct 2011).
- Will be part of the Virtual Physiological Human Network of Excellence (VPH-NoE) Toolkit (Expected 1st release June 2012);
- Will be used by p-medicine, VPH-SHARE and ContraCancrum EU projects to securely access patient data and grid resources including XSEDE.
  - [http://www.contracancrum.eu/](http://www.contracancrum.eu/)
  - [http://www.p-medicine.eu/](http://www.p-medicine.eu/)
  - [http://www.vph-share.eu/](http://www.vph-share.eu/)
Thank You

Questions?