Open Science Grid

Running User Jobs In the Grid without End User Certificates - Assessing Traceability

Anand Padmanabhan
CyberGIS Center for Advanced Digital and Spatial Studies
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

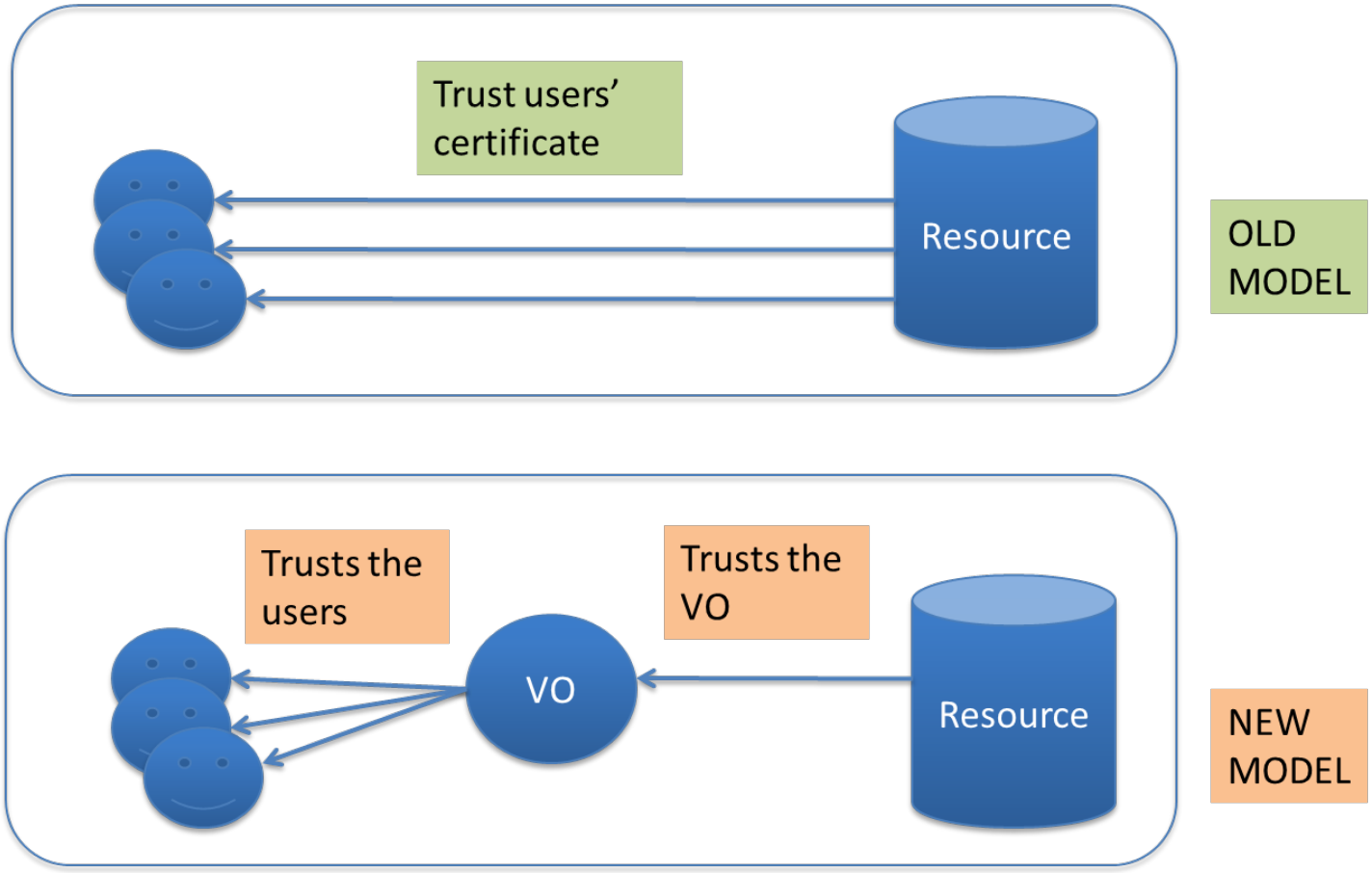
OSG Security Team
OSG AHM 2015

Open Science Grid

Motivation

- X.509 user certificates expose users to complexities of authN/authZ systems
  - First hurdle for every new Grid user
  - Represents a significant barrier to promoting ease of use and discourages potential users
- End user certificates are however the primary way to authenticate on Grids

Changing Trust Paradigm



VOs have more responsibility; Trust relations between sites and VOs needs to be higher

Changing Trust Paradigm

- OLD TRUST model
  - Sites **only trust user's with a certificate**
  - **VOs do not play a role in this trust relationship**
    - This is inefficient
- NEW TRUST model
  - Sites **trust VOs to provide the users' information when needed**
  - Sites **only know which VO the user belongs to**
  - VO **trusts its user to behave properly and takes responsibility for its user's actions**
  - Sites don't know the identity of the user upfront
    - If a problem is noticed with a user job, sites will
      - contact the responsible VO
      - expect VOs to identify and ban the problem user
  - If a problem persists, the site can ban the entire VO

Why we needed the New Trust Model? What was wrong with the Old Model

- The old model did not recognize the important role a VO played in trust relationships.
- It puts the burden of proving identities on the users.
  - VO already knows its users' identities and could provide it to sites.
- **Biggest beneficiary of the new model is the end user.**
  - If site did not need to establish trust with each user, then user did not need to deal with proving its identity and role.
  - Users do not need to obtain certs, create proxies, etc
  - Hides complexities of authN/authZ systems
  - Promotes ease of use

Does the New Trust Model Meet Our Security Needs?

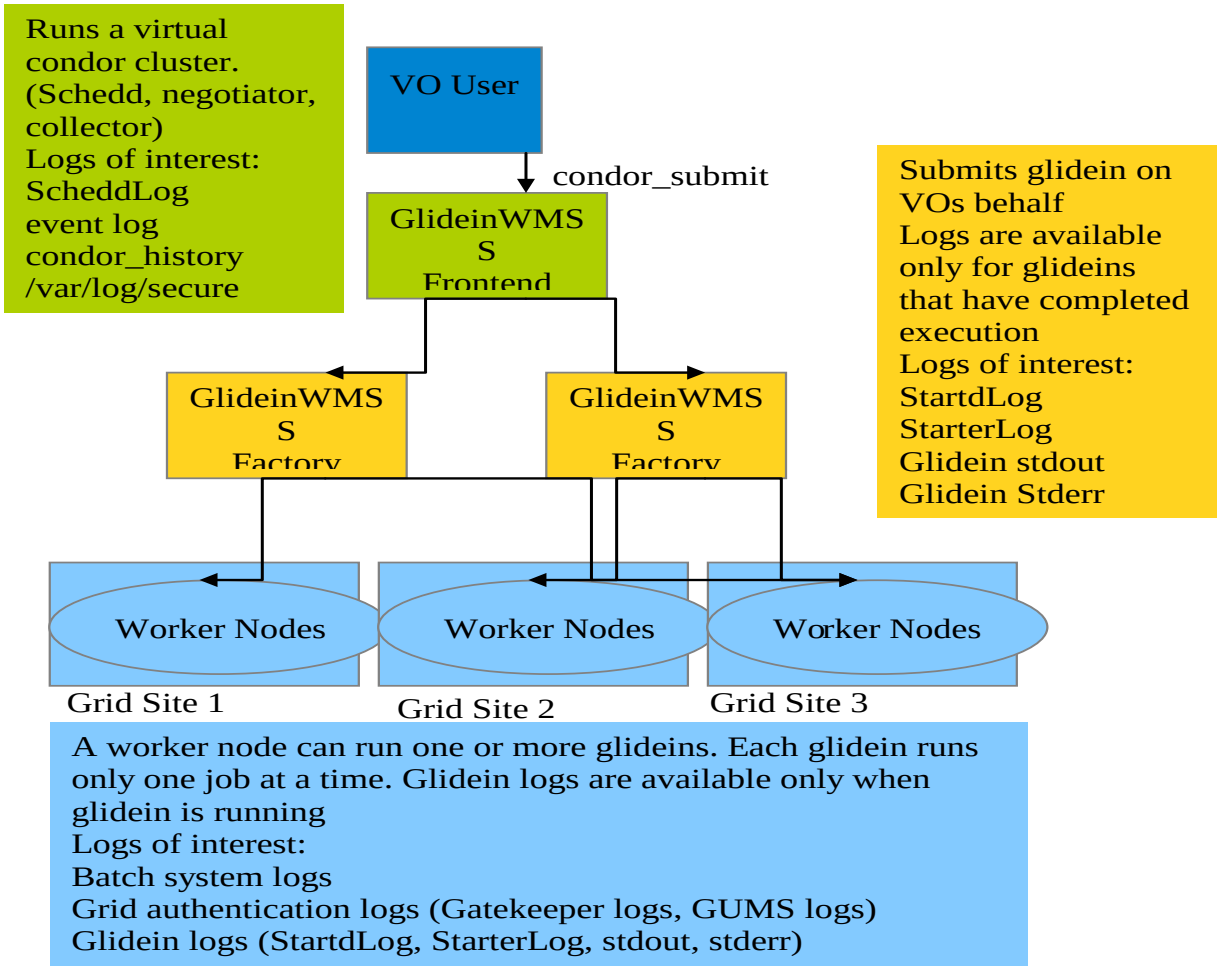
- Needs for knowing a user's identity or distinguishing a user from one another with access tokens
  - Fine-grain access privileges: Alice needs a different execution environment than Bob
  - Accountability: Holding users responsible for their actions on the grid
- Most VO members (limited exceptions like software installers exist) need the same access privileges
  - We do not hence need fine-grain privileges for most jobs
- **Accountability** – tracing a malicious job to its owner is the main reason for using certificates
  - Sites typically do not care who a user is unless a problem occurs
- We hence evaluate glideinWMS for traceability

Open Science Grid

Hypothesis

- Pilot job framework (e.g. GlideinWMS) already collect sufficient amount of data about users and jobs. So, it already has traceability information.
  - This will lead to improved usability
  - Existing pilot job framework are well positioned to support this model
- **Goals**
  - Research: Can traceability be achieved through different technical means?
  - If so, what are the security risks of using these means?

GlideinWMS Architecture



GlideinWMS Workflow

- GlideinWMS acts as a shield to protect users form complexities of the job submission on the Grid
- User accesses the Frontend and submits a job
- Frontend communicates with the Factory and requests Glideins (i.e. Pilot jobs)
- Glideins checks if the worker node suitable for the user job and then starts the user job
- Glidein job is a parent process to the actual user job and watches over the user job throughout its lifetime
  - GlideinWMS framework is documented at http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/doc.prd/index.html

GlideinWMS Details

- A worker node can run multiple Glideins and user jobs simultaneously
- An individual Glidein only starts a single job at a time
  - A Glidein can however run a sequence of jobs one after the other
    - Jobs may belong to different users
- All HTCondor logs associated with the Glidein are written on the worker nodes and are transferred back to the Glidein factory after the Glidein completes execution
- Glideins typically run for 24-48 hours

Experiment Setup

- The security team submitted jobs on OSG sites without end user certificates and evaluated if the jobs can be traced back to the submitter
- Our exercises labeled a random (non-malicious) job as malicious
- Conducted searches in both directions
  - Tracing a malicious job back to a user
  - Tracing a user to find all of her jobs
- We paid special focus to additional risks from the lack of certificates

Tracing Jobs

- Site admin identifies a problem job on a worker node at the site
- Identify the Glidein process that started the problem job
  - Use the standard error and out, Starter and Startd logs associated with Glidein
  - Search associated timelines
- Identify the VO that owns the Glidein and the problem job
  - Use Glidein DN and contact Factory operator
- With the HTCondor job id of the Glidein, look at the StartdLogs belonging to that Glidein instance
  - Find frontend where the job originated
  - HTCondor jobid at frontend
- Contact VO operating frontend to check frontend log, history files, node login logs, etc to determine user

Challenges against Traceability

- Attacker overwrites the Glidein log files
  - The user job runs in the same user account as the Glidein job, so a malicious user could hijack the Glidein infrastructure, overwrite the log files, and no useful info returned to the factory (i.e. fake logs).
  - Likelihood: Moderate. Requires some insider information
  - Mitigation: Two potential venues: 1) getting more information back to the Schedd/Frontend; and 2) making sure we always do the UID switch.
  - Currently investigating solutions

Results

- Traceability study was carefully conducted on three frontend/VOs and access to Fermilab (FNAL) resources (which previously required certificates) was enabled
  - OSG-XSEDE frontend (OSG VO)
  - CHTC frontends (GLOW VO)
    - Will present experiences today
  - HCC frontends (HCC VO)
- Careful drills were conducted and recommendations presented to FNAL security team
- Recommendations were accepted and implemented
  - Successfully opened up the use of opportunistic resources from FNAL to a number of users
- Process we went through is documented on twiki
  - E.g. https://twiki.grid.iu.edu/bin/view/Security/HCCJobTraceability

Open Science Grid

Findings

- GlideinWMS has shown to possess significant tracing capabilities
  - System can identify a unique owner for a grid job at a worker node for a given timeframe
- Some corner cases could make tracing more challenging
  - Likelihood is however low and mitigations have been recommended
- Has profound affects on trust relationships
  - Site trusts VO and VO trusts its users

What to do if my VO is Interested

- Defined a process for VOs to submit jobs under the new trust model
  - Prospective VO applies to the Security Team
  - Security Team evaluates them against a criteria
    https://twiki.grid.iu.edu/bin/view/Security/JobTraceabilityWithoutCerts
  - Example Questions
    - How do you manage your users; how do you vet their identity, give permissions and assign roles to users. How do you document and log your users activities.
    - What access control mechanisms do you employ to ensure only authorized VO members can access VO services?
    - What is the architecture of the glidein setup at your VO?
    - Does your VO operate more than one submit node (e.g. PI controlled submit node) and use flocking, if so how do you trace users between these systems
    - Do you have a centralized system for collecting logs?
    - Do you have a policy of how access is revoked?
  - If a VO demonstrates that they can manage their users effectively and take responsibility for them, they switch to operate under new trust model

OSG Procedures and Policies

- Old and new model co-exist
- Many VOs prefer to switch to the new model
  - Easier on new users
- Some VOs will continue to use the old model
- OSG leaves the decision up to the VOs
  - OSG provides the infrastructure and services to enable both models

Questions?

- Thank you