Design and Application of a Scalable Virtual Organization Privileges Management Environment

Nanbor Wang <nanbor@txcorp.com>
Gabriele Garzoglio <garzoglio@fnal.gov>
Balamurali Ananthan <bala@txcorp.com>
Steven Timm <timm@fnal.gov>
Tanya Levshina <levshin@fnal.gov>

Tech-X Corporation
Fermi National Accelerator Laboratory

ISGC 2011, Taipei, Taiwan
March 23, 2011

Funded by US DOE OASCR
Grant #DE-FG02-07ER84733
Outlines

• Project motivations
  – What SVOPME tries to address
• eXtensible Access Control Markup Language (XACML) and domain-specific policy templates
• VO-side implementation and support
• Grid-site implementation and support
• Extending new policy templates
• Current progress and deployment
• Conclusions
What are VO Privileges?

Virtual Organizations:
- VOs use shared resources
- VOs need to define resource usage policies for different users within the VOs
  - Example 1: Production team members submit jobs with higher priority
  - Example 2: Software team members can write to disk area for software installations but others can’t
- However, VOs do not manage/configure Grid sites

Grid Sites:
- Grid sites provide resources
- Grid sites don’t define VO’s usage policies
- Grid sites enforce and manage user privileges
- Grid sites do not allow others (such as VO admins) to change the site configurations

Site and VO Challenge: Enforcing heterogeneous VO privileges on multiple Grid sites to provide uniform access to VOs based on their policies across the Grid (ad hoc solution: verbal communication)
State-of-the-Art
User Privilege Management

VO Services manage only VO groups and roles, and user memberships.

A VO passes only information about groups and roles to sites.

GUMS maps a user (with group/role membership) to a local ID.

Privileges management and enforcement points are scattered at different places, using mechanisms provided by the resources. This can be as straightforward as setting the permissions to certain directories. Grid sites manage and control these configurations. They involve tweaking GUMS and all the local resources configurations.

The OSG Authorization Infrastructure
Motivations of SVOPME

- With the growth in Grid usage, both the numbers of VOs and Grid-sites increase
  - More opportunistic usage
  - Many Tier-3 sites lack the necessary man-power to keep up with VOs
- Propagating privilege policies by verbal communication between VO and Grid site admins no longer scales
- SVOPME fills the gap by
  - Providing the tools and infrastructure to help
    - VOs express their policies
    - Sites provide proper supports to VOs
  - Reuse proven administrative solutions

Addressing scalability
Employing eXtensible Access Control Markup Language (XACML)

- An XACML policy definition consists of
  - A “Target” describing where the policy applies to, by specifying
    - Subjects: a list of users requesting access
    - Resources: a list of target resources
    - Actions: a list of intended actions
  - A list of “Rule”s that grant/deny access under specific “Condition”s defined in the Rule
    - Also possible: “NotApplicable” or “Intermediate”

- An XACML request describes the kind of access
  - Like Target, it consists of subject, resource(s), and actions(s) desired

- SVOPME uses XACML to replace the verbal communication between VOs and sites
  - Avoid ambiguity by using XACML
  - Ensure conformance by using test requests
Utilizing XACML to Describe and Verify VO Privilege Policies

**VO administrators**
- Document the VO privilege policies in XACML format
- Generate a set of corresponding test requests

**Site administrators**
- Synthesize a set of equivalent privilege policies from the site configuration
- Verify conformity to a VO’s privilege policies programatically
  - Download all the test requests of the VO
  - Issue all requests with site policies should all result to “Permit”
Domain Specific Privilege Policies

- **XACML** is a very generic XML-based language for specifying access control policies
  - Not very human-readable
  - Too many variations to express the same policy

- **Thus, without some restrictions, it can be hard to**
  - Express the privilege policies consistently
  - Know what site configurations to look for
  - Synthesize local configuration policies

- **SVOPME** therefore defines a set of common privilege policies for the VOs and sites
  - Confines the problems
  - Allows us to design a set of tools targeting these policies
  - Easy to expand

- **Defining common policies as XACML templates enables:**
  - VO policy editors
  - Grid configuration probes
  - Policy Comparison
  - Grid configuration advisory
SVOPME Currently Support These Types of Policies (VOs can define)

- **Account Type Policy**: Run job from Group(G) and Role(R) using Pool (unique)/ Group (shared) accounts.
- **Account Mapping Policy**: Must have accounts for all users in the Group (G) and Role (R) sharing a pool account
- **Relative Priority Policy**: Jobs from Group (G1) and Role (R1) should have higher priority than those from user of Group (G2) and Role (R2).
- **Preemption Policy (Batch system)**: Jobs from Group (G) and Role (R) should be allowed to execute for n consecutive hours without preemption.
- **Package Installation Policy (Storage)**: Allow Group (G) and Role (R) to install software in $OSG_APP (assuming there is NO space reserved for any VO)
- **Unix Group Sharing Policy (Batch system)**: Accounts belonging to /Group/Role=A and /Group/Role=B must share the same unix Group ID
- **File Privacy Policy (Storage)**: Files Privacy Policy: Users belonging to /Group/Role=A expect privacy for their files
- **Job Suspension Policy (Batch system)**: Do not suspend / resume jobs submitted from /Group/Role=A
- **Disk Quota Policy (Storage)**: Assign disk quota of X GB and Y MB to accounts mapped to /Group/Role=A
VO Policy Editor and Compiler

- VO Administrator can create and edit a set of VO policies
- Two ways of composing/editing privilege policies
  - GUI editor
    - Interactively direct administrator how to create/edit policies
    - Overview of all policies
  - Policy compiler
    - Compile text-based domain-specific policy text file into XACML format
- Reject redundant and contradicting policies
- Also create/maintain the corresponding test requests
- A small utility (voms-client) to ensure the use of correct VO FQANs

# Example domain-specific privilege policy file
Amp1 AccountMapping /TECH-X/Role=Production group
Amp2 AccountMapping /TECH-X/Role=Test pool true
PPn Priority /TECH-X/Role=Software /TECH-X/Role=Production
VO Policy Data Management

• The Editor stores the policies and verification requests under predefined directories

• The requests are published as bundles that site can access over the net (they are not pushed to sites)

• Request Archiver collects and zips up all test requests into a time-stamped zip file
  – Time-stamped request zip archives are made available to site via a simple web page
  – Sites can scan the page and determine the latest version

• VO admins and users can use Comparer Client to contact and check a site’s support to VO policies
  – Sites need to support comparer web service interface (describe later)
SVOPME VO Tool Overview

VO Privilege Policy Editor

VO Administrator Creates/Edits

VO Privilege

Uses

VO Privilege Policies

XACML VO Privilege Policies

VO HTTP Server

VO Administrator

Comparer Client Invokes

Comparer services hosted at Grid sites

VOMS Client Uses

Retrieves VO Groups/Roles

VO Administrator Uses

VOMS Server

XACML VO Requests

Reads

Request Archiver Creates

Time-stamped Zip Archives

Published via

Comparer services hosted at Grid sites

Uses

Latest
Mechanism for Synthesizing Grid Site Privilege Policies

“Grid Probes” in a nutshell
- Policy building and configuration crawling functions are separated
- Depending on the target privilege, different info is necessary: there are multiple crawling executables
- Invoked by different cron tasks with different privileges
- Dump the info as simple text files at a specific directory
- Allow site-specific probes
- Site administrators decide how often to run the probes

Policy Builder
- Parses the intermediate configuration info
- Synthesizes the effective privilege policies of a site into XACML policies
- Does not rebuild if no configuration change

Configuration checked
- Condor/GUMS config
- Filesystems/SE
  - Disk quota/directory permissions
Grid Configuration Probes

• GUMS
  – GUMS web service
  – Get a list of VO users/FQAN to local user ID mappings

• Priority
  – Dump all users/FQANs priority assignments from Condor

• File privacy
  – Local user ID’s home directories and their permissions are recorded

• Unix group
  – Local user ID’s group memberships are recorded

• Job runtime
  – Check the MaxJobRetirementTime of the Condor scheduler’s headnode configuration

• Job suspension
  – Record FQANs that are configured as WANT_SUSPEND==FALSE in Condor.

• OSG_APP
  – Check if OSG_APP is set on the site. If so, record its permission and ownership

• Disk quota
  – Check allotted quotas for FQAN mapped local user IDs (requires root privilege)
Sites Cache VO’s Verification Requests

- Sites decide which VO\'s they want to support
- A utility \"VO Request Retriever\" helps manage local caches of VO verification requests
  - Checks if the local caches of VO verification requests are up-to-date using timestamps
  - Download and cache new set of verification requests if needed
  - Organize multiple VO request caches into different subdirectories
Analyzing and Verifying Site Configurations

- **Policy Advisor**
  - Test compliance by testing the verification requests one-by-one
  - Allow verification of a single VO
  - Since all requests and policies are based on our XACML profiles, reports and advises can be derived

- **Policy Compare**
  - Web/Grid service interface for submitting test requests
  - Users can check support to specific VO/policies
  - Effective hiding site configurations from outside
VO/Grid Policies Advisor

- Provide advices for the **Grid site administrator** on what amendments need to be done on the Site; such that the Grid site complies with the VO policies

- Example output:
  - VO requested 3 accounts for VISITORS role via VO policies
  - Site-policies derived from GUMS do not match

---

**VO/Grid Grid Accounts Policy Advices**

No matching Grid Accounts Policy was found for /TECHX/VISITORS on the Grid site. Create a mapping in GUMS config such that /TECHX/VISITORS be mapped to at least 3 account(s)

TECHX/Role=VO-Admin mapped to 1 account(s) (techxVOadmin) on the Grid site, is not sufficient enough. *Needs to be mapped to at least 3 accounts.*
VO/Grid Policies Comparer

- Policy Comparer Grid Service
  - Allow VO users to check privilege policy compliance at a site
  - Instead of cached verification requests, users supply a list of verification requests related to policies of interests
  - SVOPME provides a policy comparer client as part of the VO tools
  - Currently only provide text reports – should provide a mechanism that further automates the information gathering
  - VO should aggregate results from multiple sites

- Example output:

VO/Grid Grid Accounts Policy Comparison

/TECHX/Role=User is mapped to 1 account(s) on the Grid site. Passed!

No Account Mapping Policies for /TECHX/VISITORS were found on the Grid site.
SVOPME Architecture

**VO Privilege Policy Editor**
- VO Privilege Policy Editor
- VO Administrator

**XACML VO Privilege Policies**
- VO Administrator

**XACML VO Requests**
- VO Administrator

**SVOPME Application**
- SVOPME Application
- XACML Document
- Text Document

---

**Grid Site**

**Grid Probe**
- Synthesizes
- Storage Elements
- Computing Elements

**XACML Effective Site Access Policies**
- Grid Probe
- Site Administrator

**Policy Comparator**
- Uses
- Suggested Changes
- Compliance Report

---

**VO**
- Uses
- VO Privilege Policy Editor

---

**Grid Site**
- Storage Elements
- Computing Elements
- Grid Probe

---

**SVOPME Application**
- SVOPME Application
- XACML Document
- Text Document

---

**Grid Site**
- Grid Probe
- Synthesizes
- XACML Effective Site Access Policies

---

**Policy Comparator**
- Uses
- Suggested Changes
- Compliance Report

---

**VO Privilege Policy Editor**
- VO Privilege Policy Editor
- VO Administrator

---

**XACML VO Privilege Policies**
- VO Administrator

---

**XACML VO Requests**
- VO Administrator

---

**SVOPME Application**
- SVOPME Application
- XACML Document
- Text Document
Advantages of SVOPME

**VOs**
- No need to run ad-hoc jobs to figure out what policies are enforced and what not
- Provides templates to define commonly used policies
- Automates most of the communication with Sites that support the VO
- Provides the basis for the negotiation of privileges at sites that provide opportunistic access

**Sites**
- Sites can advertise and prove that a VO is supported
- Sites that want to support a VO have a semi-automated mechanism to enforce the VO policies
- Privilege enforcement remains responsibility of the Site, informed by formal VO policy assertions
Extending Meta-Policies

• It is possible to extend SVOPME to support new privilege policy profiles.

• Extending the utilities is done
  – Using generic classes
  – Using interfaces
  – Using class loader

• Steps to extend SVOPME
  – Define what access right the policy type would control (subject, action, etc.)
  – Define how the XACML policy would look like
  – Extend the VO Editor to support the policy type
  – Extend Grid Probe to crawl relevant resource configurations and build the Grid policy based on the Grid Probe findings.
  – Extend Policy Comparer/Advisor to interpret the test results
Current Progress

- VO and site tools are packaged
  - Zip files
  - Pacman packages

- Performed experiment deployment on FermiGrid Integrated TestBed (ITB)
  - Emulated “DZero” and “Engage” VO’s privileges as examples
  - Was able to detect several anomalies
  - Prompted the use of multiple probes to adapt to different site configurations/requirements

- Working with Engage VO to set up VO publishing point
  - Engage VO encompasses many smaller groups that want to take advantage of OSG
  - Policies can change often

- Working with US-ATLAS and US-CMS Tier-3 sites
  - Often have less human resources to maintain the site
  - We will work with the VOs to create sample policies
Conclusions

• SVOPME ensures uniform access to resources by providing an infrastructure to define, propagate, verify, and enforce VO policies at Grid sites
• SVOPME integrates with the OSG Authorization Infrastructure
• We continue to enhance SVOPME design and implementations
• We are working with interested VOs and sites to deploy SVOPME in a production environment
• Question, comments, or suggestions? https://ice.txcorp.com/trac/svopme/