Table of Contents

Security in CREAM

Authentication

Authentication in CREAM is managed via the trustmanager.

The Trust Manager is the component responsible for carrying out authentication operations. It is an implementation of the J2EE security specifications. Authentication is based on PKI. Each user (and Grid service) wishing to access CREAM is required to present an X.509 format certificate. These certificates are issued by trusted entities, the Certificate Authorities (CA). The role of a CA is to guarantee the identity of a user. This is achieved by issuing an electronic document (the certificate) that contains the information about the user and is digitally signed by the CA with its private key. An authentication manager, such as the Trust Manager, can verify the user identity by decrypting the hash of the certificate with the CA public key. This ensures that the certificate was issued by that specific CA. The Trust Manager can then access the user data contained in the certificate and verify the user identity.

Authorization

Authorization in the CREAM CE can be implemented in two different ways (the choice is done at configuration time):

- Authorization with ARGUS
- Authorization with gJAF

Argus is a system meant to render consistent authorization decisions for distributed services (e.g. compute elements, portals). In order to achieve this consistency a number of points must be addressed. First, it must be possible to author and maintain consistent authorization policies. This is handled by the Policy Administration Point (PAP) component in the service. Second, authored policies must be evaluated in a consistent manner, a task performed by the Policy Decision Point (PDP). Finally, the data provided for evaluation against policies must be consistent (in form and definition) and this is done by the Policy Enforcement Point (PEP). Argus is also responsible to manage the Grid user - local user mapping.

gJAF (Grid Java Authorization Framework) provides a way to invoke a chain of policy engines and get a decision result about the authorization of a user. The policy engines are divided in two types, depending on their functionality. They can be plugged into the framework in order to form a chain of policy engines as selected by the administrator in order to let him set up a complete authorization system. A policy engine may be either a PIP or a PDP. PIP collect and verify assertions and capabilities associated with the user, checking her role, group and VO attributes. PDP may use the information retrieved by a PIP to decide whether the user is allowed to perform the requested action, whether further evaluation is needed, or whether the evaluation should be interrupted and the user access denied. In CREAM CE VO based authorization is supported. In this scenario, implemented via the VOMS PDP, the administrator can specify authorization policies based on the VO the jobs' owners belong to (or on particular VO attributes). When gJAF is used as authorization mechanism, the Grid user - local user mapping is managed via glexec,

For what concerns authorization on job operations, by default each user can manage (e.g. cancel, suspend, etc.) only her own jobs. However, the CREAM administrator can define specific super-users who are empowered to manage also jobs submitted by other users.

Please note that for CREAM, `proxyX` and `proxyY` with the same distinguish name (DN) are considered belonging to two different users if they have different primary FQAN.

-- MassimoSgaravatto - 2011-04-18

This topic: CREAM > CreamSecurity
Topic revision: r5 - 2011-07-21 - EricFrizziero