Table of Contents

Proxy Restrictions

Rationale

There is a certain probability that a proxy certificate, which has been delegated to a worker node (WN) gets stolen. To mitigate the harm what can be caused by a stolen proxy certificate one can restrict its usage.

There are already some restrictions in place: first of all the expiration time of the proxy and the 'limited' attribute in the DN, which disallows the proxy to be used in job submission again.

Here we propose some further optional extensions that can restrict the usage of a proxy *from* a given site and *to* some set of sites.

And for auditing purposes another extension to record the host names [and IP addresses?] where the delegation happened.

Matching Rules

The restrictions are a simplified version of the Appache httpd's 'Allow' directive, where the following type of restrictions are applicable:

A (partial) domain-name

Example:

restrict-from
: example.org

restrict-from
: .net example.edu

Hosts whose names match, or end in, this string are allowed access. Only complete components are matched, so the above example will match 'foo.example.org' but it will not match 'fooexample.org'.

IPv4 address (with mask)

Example:

restrict-from
: 10.1.2.3

restrict-from
: 10.1.0.0/16 The full IPv4 address has to be the source IP address, while with the netmask, only the first 16 bits have to match.

IPv6 address (with mask)

Example:

restrict-from
: 2001:db8::a00:20ff:fea7:ccea

restrict-from
: 2001:db8::a00:20ff:fea7:ccea/10 Similar to the IPv4 case, however with IPv6 addresses.

Restrictions can be added in any certificate in the certificate chain and multiple restrictions can be added to a single proxy certificate as well.

The restrictions inside a single certificate should be 'or'-ed: if any of them matches, then the restriction is satisfied. [rationale: a farm may span multiple domains or IP address segments]

However restrictions coming from different proxies should be 'and'-ed: all restrictions have to be satisfied. [rationale: the attacked might do an extra delegation step and adds some new restrictions, however those should not invalidate the previous ones]

'From' Restriction

In the last delegation step, where the proxy is being delegated to the destination computing element (CE) one can restrict the proxy that it could only be used *from* the worker nodes of that CE.

This can be achieved by adding a 'restrict-from' optional extension, which should be checked by the services that authenticate the proxy.

Example: The destination CE is 'ce.example.org' and its worker nodes have host names like 'wn0003.farm.example.org'. In this case the delegation service on 'ce.example.org' could add the follwing restriction:

restrict-from: farm.example.org

When the proxy is used on 'wn0003.example.org' to access a storage element (SE), that service could check if the source host name matches the one in the restriction.

If the proxy gets stolen and the attacked uses it from 'ui.example.edu', the check will fail:

ui.example.edu !~ farm.example.org

'To' Restriction

The user submitting the job might have a better knowledge of what services shall be accessed on her/his behalf and may express this in the form of a *to* restriction.

This restriction shall be added by the user to the proxy, which is used to submit the job.

Example: The user knows that her/his job should access only one storage element and nothing else:

restrict-to: se.example.edu

The services receiving a request from a job, with a 'to' restriction should match their own host name and IP address with the restriction. If the service's host name or IP address does not match, it should refuse to serve the request.

Delegation Trace

In the ideal case of delegation a new proxy is created at each step, when the host is changed. To track these steps for auditing the involved delegation services may add the following fields to a delegated proxy: 'delegated-from' and 'delegated-to'.

Example: delegated-from: ui.example.edu delegated-to: scheduler.large.vo

The 'delegted-from' field is the host name from where the delegation was initiated. The 'delegated-to' field is the host name of the delegation service, where the new proxy was created.

In case of an authentication failure this trace can be added to the audit logs.

Q&A

- what happens, if the proxy is restricted by the CE, however it is used in an SRMcopy operation? The SE must act on behalf of the user, however the SE might not be inside the restricted domain.

- Shall we use proper ASN.1 structures for the restrictions and trace or a simple string representation is enough? In case of an ASN.1 structure we have to allocate our new OID arc. In case of a simple string extension we might just use VOMS' include OID: 1.3.6.1.4.1.8005.100.100.2 This would also simplify the introduction of this idea, as any user can include a simple text file under this OID using the '-include' option of 'voms-proxy-init'

TODO

- add the proposed extensions to the delegation services: GridSite (C) and gLite delegation (Java)

- add optional check for these extensions to
    - trustmanager (Java)
    - gridsite (C)
    - LCAS/LCMAPS (C)

- provide methods to retrieve the delegation trace for logs

Last edit: AkosFrohner on 2009-03-09 - 10:36

Number of topics: 1

This topic: EGEE > DelegationProxyRestrictions
Topic revision: r3 - 2009-03-09 - AkosFrohner