EUROPEAN MIDDLEWARE INITIATIVE

LCMAPS-PLUGINS-C-PEP

Document version:	**1.0.0**
EMI Component Version:	**1.1.13**
Date:	**April 28, 2011**

This work is co-funded by the EC EMI project under the FP7 Collaborative Projects Grant Agreement Nr. INFSO-RI-261611.

NAME

lcmaps-plugins-c-pep - LCMAPS plug-in PEP-C Policy Enforcement Point Daemon client

SYNOPSIS

lcmaps_c_pep.mod *--pep-daemon-endpoint-url <url>* [*--resourcetype rb|ce|se|wn*] [*--actiontype queue|execute-now|access*] [*--pep-c-debug*] [*--[no]-check-certificates*] [*--capath <dir>*] [*--cafile <path>*] [*--cert <path>*] [*--key <path>*] [*--passfile <path>*] [*--pass <plaintextpassword>*] [*--resourceid <uri>*] [*--actionid <uri>*] [*--profile <profile name>*]

DESCRIPTION

The LCMAPS plug-in **lcmaps_c_pep** utilizes the PEP-C library to contact the PEP daemon. It will send the user credentials and, if applicable, the pilot job credentials and extra information to the PEP daemon.

The PEP daemon will process the request and query the PDP, PAP, EES chain for a policy decision. The PEP daemon will return a Permit statement with a Unix account. The Unix account must be composed of a Unix User ID and Unix Group ID. Optionally Unix Secondary GIDs may be returned. All of these IDs must be returned in numerical form. The results will then be published in the LCMAPS framework.

The plug-in will use the credentials loaded in the LCMAPS framework for the primary authorization decision. The returned Unix account will reflect this identity. Additionally to this identity, in a multi-user pilot job scenario, the X509_USER_PROXY environment variable is read to add information about the identity that executes the pilot job framework and triggered the execution of this plug-in. This probably with the use of gLExec.

OPTIONS

--actiontype queue|execute-now|access

The action type option will declare the type of action that is intended to be performed. The **queue** option signifies an execution to a queue. That's mostly due to a submission of a computer job to a queue. The **execute-now** option signifies the direct execution of a command or a job. Use cases for this action are the LCG-CE's fork-queue or gLExec where there is no (significant) delay for the operation's execution.. The **access** option signifies the access of a file at a storage facility of any kind. The true type of (file) access, like reading, writing, execution or listing, is not declared because this would be too detailed and poses practical limitation in the interaction with different storage system.

--pep-daemon-endpoint-url <url>

This will configure the plugin to contact the PEP daemon at <url>. Multiple endpoints can be set in the order specified by the **--endpoint-strategy** option.

--resourcetype rb|ce|se|wn

The resource type option will identify this PEP by its type of resource. The possible types that can be signified are rb, ce, se and wn. The **rb** option is to signify a Resource Broker or Workload Managment System or an differently named high level scheduler. The **ce** is to signify a Computing Element as a front-end node to a compute cluster like a LCG-CE or CREAM-CE. The **se** option is to signify a Storage Element, like DPM, dCache, Castor, StoRM or something else. The **wn** is to signify a Worker Node, like an LCG-WN, a compute node with gLExec on it for example.

--pep-c-debug

This option will trigger the PEP-C library to log at the maximum verbosity level and will let output detailed message, like HTTP protocol interactions with the PEP daemon, to be written to stderr. Without this option the detailed information is directed to /dev/null. Most error messages can be retrieved in sufficient detail from the PEP-C library in a different way. If you enable this option, the option_client_keypassword will be printed in clear-text to the invoker on stderr!

--[no]-check-certificates

This option will trigger the PEP-C library to disable or enable SSL validaiton on the connection to the PEPd server. By default, SSL validaiton is enabled. Using --no-check-certificates will ignore any SSL validation errors on the PEPd interconnect, but does not influence any checks on the invoking user or the target proxy.

--capath <directory>

Path to a directory containing the trust anchors. Each trust anchor must be in PEM format, and should be named according to the OpenSSL c_hash convention. The default will be one of $X509_CERT_DIR, $HOME/.globus/certificates, or /etc/grid-security/certificates (in that order).

--cafile <file>

Path to a file containing the trust anchors.

--cert <file>

Specify the file that contains the client certificate used to connect to the PEPd. When this option is set, client authentication is implicitly enabled, and a corresponding key file (--key option) must be provided.

--key <file>

Specify the file that contains the client private key used to connect to the PEPd. When this option is set, client authentication is impli citly enabled, and a corresponding certificate chain (--cert option) must be provided.

--passfile <file>

Specify the file that contains the password used to decrypt the private key for the PEPd handshake. This option is mutually exclusive with the --pass option.

--pass <password>

Specify the password in clear-text that will be used to decrypt the private key for the PEPd handshake. This option is mutually exclusive with the --passfile option. Using a password file (with appropriate permissions) is preferred.

--resourceid <urn/url>

This option will set the resourceID in the request protocol message.

--actionid <urn/url>

This option will set the actionID value in the request protocol message.

--profile <profile name>

The name of the profile that the plug-in must use. Currently supported Profiles names are "http://glite.org/xacml/profile/grid-wn/1.0" (default) and "http://authz-interop.org/profile/1.1". For more information, see the SUPPORTED PROFILES section.
The profile setting of "http://glite.org/xacml/profile/grid-wn/1.0" (default) requires the installation and availability of the libpepc.so library version 1.3.0-1 or newer.

ENVIRONMENT

X509_USER_PROXY

The value of the X509_USER_PROXY environment holds the path to the proxy certificate. This is not the primairy identity on which the authorization decision is based on. This proxy certificate identifies the Pilot Job executor. This identity is responsable for pulling a pilot job payload associated with a proxy onto a Worker Node during a job execution.

NOTES

When an https end point is used for the PEP daemon, client-side authentication can be enabled by specifying a file with a certificate chain and a file with the associated private key. If an https end point is specified but no certificate or key is provided, an anonymous secure connection is established. The server identity is always verified using the trust anchor repository specified by the --capath or --cafile options. If neither of these is specified, the plugin will use the directory referred to by the X509_CERT_DIR environment variable, or fall back to /etc/grid-security/certificates. If neither directory can be found, the system default trust anchor store is used.

SUPPORTED PROFILES

There are different profiles to which the plug-in can adhere. Each profile describes the way in which the XACML attributes and obligations are to be used and what each of their intentions are. In the previous versions (all versions before 1.0.1-1) of this plug-in only the attributes and obligations mentioned in the "An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids" document, version 1.1 (October 09, 2008) were used. Since January 13th the first draft of the "XACML Grid Worker Node Authorization Profile, Version 1.0" has been submitted for review. As there is a strong push to support the attributes and obligations stated in the Grid WN profile this profile is the new default.

NAMESPACE USAGE

The namespace usage complies with "An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids" document, version 1.1 (October 09, 2008).

EDMS : https://edms.cern.ch/document/929867
FNAL CD : http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=2952

The Subject section of the request message will use the namespace "http://authz-interop.org/xacml/subject/cert-chain" to set the proxy certificate of the (real) user (job) credentials. The value is a base64 encoded string of the proxy certificate.

The action-id namespace is by default: "http://authz-interop.org/xacml/action/action-id".

It's value is set by the parameter value provided through the **--actionid** option, which can be anything as we do not check if the provided string value is a proper URL/URN. The option **--actiontype** also sets the action-id namespace, but the provided values **access-file**, **queue** and **execute-now** are expanded internally to the following namespace values:
http://authz-interop.org/xacml/action/action-type/access-file
http://authz-interop.org/xacml/action/action-type/queue
http://authz-interop.org/xacml/action/action-type/execute-now
When both the **--actionid** and **--actiontype** options are set, the value of the **--actionid** will take presidence.

The resource-id namespace is by default: "http://authz-interop.org/xacml/resource/resource-id".

It's value is set by the parameter value provided through the **--resourceid** option, which can be anything as we do not check if the provided string value is a proper URL/URN. The option **--resourcetype** also sets the resource-id namespace, but the provided values **ce**, **se**, **wn** and **rb** are expanded internally to the following namespace values:
http://authz-interop.org/xacml/resource/resource-type/se
http://authz-interop.org/xacml/resource/resource-type/ce
http://authz-interop.org/xacml/resource/resource-type/wn
http://authz-interop.org/xacml/resource/resource-type/rb
When both the **--resourceid** and **--resourcetype** options are set, the value of the **--resourceid** will take presidence.

The dns-name of the host is set in the Resource section of a request and set in the namespace "http://authz-interop.org/xacml/resource/dns-host-name". The value is the FQDN of the resource which sends out the request.

The Environment section of the request message will use the namespace "http://authz-interop.org/xacml/subject/cert-chain" to set the proxy certificate of the pilot job executor’s credentials. The value is a base64 encoded string of the proxy certificate.

The registered Obligation handlers are trigger by the namespaced IDs:

http://authz-interop.org/xacml/obligation/uidgid
http://authz-interop.org/xacml/obligation/secondary-gids
http://glite.org/xacml/obligation/local-environment-map/posix

The obligation identified by "http://authz-interop.org/xacml/obligation/uidgid" can only be set once in a Response message. The obligation handler will trigger an error state when this obligation is stated multiple times in a response message. The obligation requires that two attributes are set. These attributes must provide the Unix UserID and Unix Primary Group ID in numerical form. The attribute with the namespace "http://authz-interop.org/xacml/attribute/posix-uid" is expected to be filled with an integer value of the Unix User ID which must exist on the system on which the Response message is going to be processed. The attribute with the namespace "http://authz-interop.org/xacml/attribute/posix-gid" is expected to be filled with an integer value of the Unix Group ID which must exist on the system on which the Response message is going to be processed.

The obligation identified by "http://authz-interop.org/xacml/obligation/secondary-gids" can be set zero or multiple times in a Response message. The obligation requires that at least one or more attribute are set. These attributes must provide the Unix Group ID in numerical form. This may be the primary Unix Group ID, but it is intended to only be filled with the secondary Unix Group IDs. The attribute with the namespace "http://authz-interop.org/xacml/attribute/posix-gid" is expected to be filled with an integer value of the Unix Group ID which must exist on the system on which the Response message is going to be processed. The obligation requires that at least one or more attribute are set. These attributes must provide the Unix Group ID in numerical form. This may be the primary Unix Group ID, but it is intended to only be filled with the secondary Unix Group IDs. The attribute with the namespace "http://authz-interop.org/xacml/attribute/posix-gid" is expected to be filled with an integer value of the Unix Group ID which must exist on the system on which the Response message is going to be processed.

The obligation identified by "http://authz-interop.org/xacml/obligation/username" can only be set once in a Response message. The obligation handler will trigger an error state when this obligation is stated multiple times in a response message. The obligation requires that one attribute is set. These attributes must provide the Unix Username in string form. The attribute with the namespace "http://authz-interop.org/xacml/attribute/username" is expected to be filled with an string value of the Unix Username which must exist on the system on which the Response message is going to be processed. The attribute declaring the username will result in system lookup to acquire the Unix User-ID, Unix primary Group-ID and the list of secondary Group-IDs of which the username is a member. This information will then be stored for further processing in LCMAPS.

The obligation identified by "http://glite.org/xacml/obligation/local-environment-map/posix" can be set once Response message. The obligation handler will trigger an error state when this obligation is stated multiple times in a response message. The obligation requires that two attributes are set, optionally more. The two required attributes are identified with "http://glite.org/xacml/attribute/user-id" and "http://glite.org/xacml/attribute/group-id/primary". They state the name of the User account and the name of the primary Group on the system. Ootionally a list of "http://glite.org/xacml/attribute/group-id" attributes can be set which state the secondary Group affifiliation on the system. These names are resolved in the obligation handlers to the actual Unix User ID and Unix Group IDs and registered in the LCMAPS system for enforcement.

A source code rebuild with the define 'USE_STANDARDIZED_NAMESPACE' will force the default namespace usage to for the resource-id to "urn:oasis:names:tc:xacml:1.0:resource:resource-id" and for the action-id to "urn:oasis:names:tc:xacml:1.0:action:action-id". As this is not used by the interoperability group as the default, and would thus break interoperability, we've choosen not to promote these settings.

EXAMPLE

The following example config file can be used for LCMAPS:

```
### gLExec on the WN for the AuthZ WG PEP-C to PEP-D interaction only
# default path for the modules
path = /usr/lib64/modules

# Plugin definitions:
posix_enf = "lcmaps_posix_enf.mod"
          "-maxuid 1"
          "-maxpgid 1"
          "-maxsgid 32"

verifyproxy = "lcmaps_verify_proxy.mod"
           "-certdir /etc/grid-security/certificates"

pepc = "lcmaps_c_pep.mod"
      "--pep-daemon-endpoint-url http://localhost:8080/PEPd/authz"
      "--resourceid http://cnaf.infn.it/wn"
      "--actionid http://glite.org/xacml/action/execute"
      "--capath /etc/grid-security/certificates"
      "--pep-certificate-mode implicit"


# Policies:
# AuthZ WG PEP-C PEP-Daemon interaction
glexec_get_account:
verifyproxy -> pepc
pepc -> posix_enf
```

BUGS

There are no bugs found (yet).

FILES

/etc/lcmaps/lcmaps.db
/usr/lib64/modules/lcmaps_c_pep.mod
/usr/lib64/modules/liblcmaps_c_pep.a
/usr/lib64/modules/liblcmaps_c_pep.so
/usr/lib64/modules/liblcmaps_c_pep.so.0
/usr/lib64/modules/liblcmaps_c_pep.so.0.0.0

SEE ALSO

lcmaps(3), **glexec**(1)

AUTHOR

Writen by Oscar Koeroo

COPYRIGHT