Authorization and virtual organisations with Apache, SSL and GACL

Frederik Orellana
Niels Bohr Institute
Copenhagen University
November 2009

Contents

1 INTRODUCTION.....3

2 SUPPORTED GACL ELEMENTS.....3

3 VIRTUAL ORGANISATION CACHING.....4

4 APACHE CONFIGURATION DIRECTIVES.....5

5 IMPLEMENTATION.....6

6 BIBLIOGRAPHY.....6

1 Introduction

In a previous paper [SECURITY], the inherent problems of traditional grid security were discussed and a simpler model was proposed. In the present paper we describe a first step towards realising this model, namely an implementation of GACL authorisation [GACL] that works with newer versions of Apache and OpenSSL. Concretely, we have written a new Apache module called mod_gacl, which can be used together with the standard Apache modules mod_dav and mod_ssl to implement a file server that enforces GACL access control and is virtual organisation aware.

2 Supported GACL elements

mod_gacl implements only a subset of the full GACL specification [GACL_SPEC]:

- VOMS directives are not supported,
- only directory permissions are supported, i.e. only ".gacl" files are parsed - any files of the form ".gacl-my_file" are ignored,
- only the following actions are supported: read, list, write, admin – these are mapped on the HTTP/WebDav methods GET, PROPFIND, PUT/MKCOL.

For reference we now summarise the supported subset of GACL:

The general form of a ".gacl" file is:

```
<gacl>
    <entry>
        [WHO block]
        [WHAT block]
    </entry>
    ...
</gacl>
```

The WHO block must be in one of the following three forms:

```
<any-user>
</any-user>
```

```
<person>
    <dn>/O=Grid/O=My Organisation/CN=Some User</dn>
</person>
```

```
<dn-list>
    <url>https://my.server.com/my_vo.txt</url>
```

```
</dn-list>
```

The WHAT block must have the following form:

```
<allow>[ALLOW/DENY block]</allow>
<deny>[ALLOW/DENY block]</deny>
```

where the ALLOW/DENY block must consist of one or several of the elements:

```
<read/><list/><write/></admin>
```

3 Virtual organisation caching

When a HTTP request is made in a directory with a ".gacl" file containg `dn-list` elements, mod_gacl reads the text files from the URLs and caches the content by creating a file ".gacl_vo". This file is a GACL file containing a WHO block with lists of `person` entries, each list accompanied by the same WHAT block as that accompanying the `dn-list` used to generate the list of persons.

If, for example a ".gacl" file reads

```
<gacl>
    <entry>
        <dn-list>
            <url>https://my.server.com/my_vo.txt</url>
        </dn-list>
        <allow><read/></allow>
    </entry>
</gacl>
```

and https://my.server.com/my_vo.txt reads

```
/O=Grid/O=My Organisation/CN=Some User 1
/O=Grid/O=My Organisation/CN=Some User 2
```

Then, the ".gacl_vo" file generated by mod_gacl will read

```
<gacl>
    <entry>
```

```
        <person>
            <dn>/O=Grid/O=My Organisation/CN=Some User 1</dn>
        </person>
        <allow><read/></allow>
    </entry>
    <entry>
        <person>
            <dn>/O=Grid/O=My Organisation/CN=Some User 2</dn>
        </person>
        <allow><read/></allow>
    </entry>
</gacl>
```

If a“.gacl_vo” file is newer than a configurable time-out, it is read, parsed and honoured just like the “.gacl” file. If it is older than the time-out, it is attempted regenerated from the URL(s).

4 Apache configuration directives

mod_gacl is configured like any other Apache modules through directives given in an Apache configuration file. mod_gacl implements support for the following directives:

DefaultPermission [permission string]

Specifies default permission for directories with no “.gacl” file.

Must be one of none, read, exec, list, write, admin.

Notice: it also seems to affect directory listings: if set to none

listings are not allowed - even by DNs allowed in the “.gacl” file.

GACLRoot [path]

Specifies alternative path to use when checking for “.gacl” files.

If given, e.g. the request https://my.server/some/dir/file.txt

will cause mod_gacl to consult GACLRoot/some/dir/.gacl for permissions.

If not given, ServerRoot/some/dir/.gacl will be consulted.

VOTimeoutSeconds [seconds]

Number of seconds to cache dn-lists.

AuthScriptFile [path to the program]

Specifies the program that caches the dn-lists (virtual organizations)

given in the .gacl files. This path should be an absolute path or relative to the ServerRoot.

The directives given below are implemented by standard Apache modules, but are understood by mod_gacl and are *mandatory*. AuthType must be set to "Basic". AuthName can be provided to prompt a browser dialog.

AuthType	Basic
AuthName	"authentication realm"
Require	valid-user

5 Implementation

mod_gacl is an Apache-2 only module, implemented in plain c, making heavy use of the examples in the The Apache Modules Book [GACL] and moreover directly using the GACL library [GACL] for parsing ".gacl" files.

6 Bibliography

SECURITY, Frederik Orellana, Christian Ulrik Søttrup, Anders Wäänänen, Daniel Kalici and Michael Grønager, "The case for a simpler security model in grid computing", *in preparation*

GACL, Andrew McNab, "Grid-based access control for Unix environments, Filesystems and Web Sites", Talk from the 2003 Computing in High Energy and Nuclear Physics (CHEP03), La Jolla, Ca, USA, March 2003, 3 pages, http://arxiv.org/abs/cs.DC/0306030

GACL_SPEC, Andrew McNab, GACL specification, http://www.gridpp.ac.uk/authz/gacl/notes-0.1.5.html. See also http://www.nordugrid.org/documents/gacl_mini_howto.html

MODULES, Nick Kew, "The Apache Modules Book: Application Development with Apache", Prentice Hall PTR, 2007, ISBN 0132409674, 9780132409674,

http://books.google.com/books?id=HTo_AmTpQPMC&printsec=frontcover