Téquila

A tool for federated authentication and access control.

Claude Lecommandeur
EPFL - SIC
claude.lecommandeur@epfl.ch

Table of Contents

Overview

EPFL is using a tool for a long time called Gaspar. This tool is the repository for authentication information. Web servers use it to authenticate their users.

The need has recently arise to use such a tool to authenticate people across several organizations. Tequila was conceived to achieve this goal.

A Tequila cell is a set of Tequila servers that agreed to trust themselves and to agree on the meaning of users attributes. Each server manages its own set of users and is willing to authenticate its users and give away the value of users related attributes (name, id, ...) on request of another client in the cell.

Tequila holds no data itself, neither authentication data, nor user attributes, it delegates these data to so called 'connectors', authentication connector data connectors.

How does it work

The job of Tequila is to authenticate people in a federated network of organizations. Each organization must provide a way to authenticate their users, and to fill a predefined set of attributes for these users.

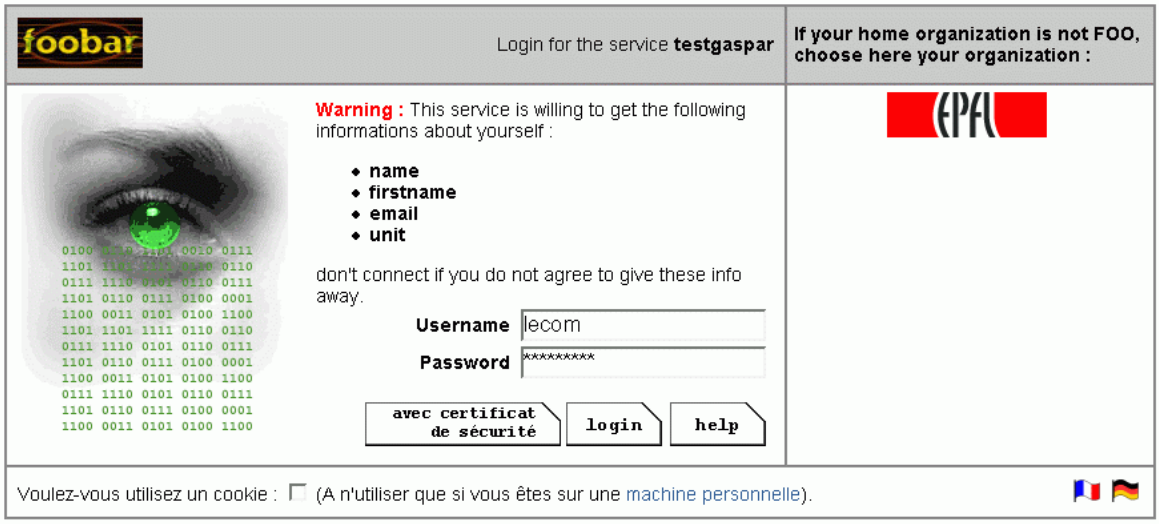
Each organization runs its own Tequila server (or several). This local server is interfaced with the local information system. The local information system must provide a way to authenticate its users via a username/password, and must provide the necessary attributes for all its users.

The communication between Tequila and the information system is made via so- called 'connectors'.

By default a LDAP connector is given. This connector supposes that all the users have an entry in the local LDAP directory with an associated password. The username must be mapped to the uid attribute in the LDAP schema.

For the other attributes, a table of mapping between Tequila attributes and LDAP attributes can be used.

When a web application wants to authenticate a user, it redirects her to the local Tequila server. The local Tequila server presents the following screen to the user :



The local organization logo is shown at the top left corner (here foobar). The right column lists all the organizations participating in the Tequila cell.

If the user is not a member of the local organization, he chooses his home organization by clicking on its logo on the right. The local Tequila then redirects her to her home Tequila server.

This schema ensures that people gives away their password only to their home server, they don't need to trust the local server.

The user is also shown which attributes the service she is authenticating to is willing to know about her. At this point, she still can decide not to disclose any information.

The user home server then authenticates her (either with his username / password, or with his SSL certificate, or even with a cookie previously set by Tequila itself), creates a session in the originating web application, and finally redirects the user to the initial application.

The application sees that there is a session opened for that user with the desired attributes, and gives him access to its protected resources.

Installing Tequila

Prerequisites

1.1 On the server side, you need Perl 5, and a few Perl packages :

- Net::LDAPS
- IO::Socket::SSL

And if you want RSA support :

- Crypt::OpenSSL::Random
- Crypt::OpenSSL::RSA

If you want to support cookies (and you want) :

- Crypt::RC4

1.2 On the client side :

- Perl 5, of course
- Apache if you want to use the mod_tequila module.
- Tomcat if you want to use the java client.
- Crypt::OpenSSL::Random and Crypt::OpenSSL::RSA if you want to use RSA.

Installing

Tequila tries to install itself in /var/www since it supposes that Apache is installed here.

The client needs only 4 files and a directory :

```
/var/www/Tequila/Sessions
/var/www/cgi-bin/Tequila/Client.pm
/var/www/cgi-bin/tequilalogin
/var/www/cgi-bin/testtequila
/usr/lib/apache/mod_tequila.so
```

(testtequila is just there as an example and for testing purpose).

mod_tequila.so is the Apache module that is used to protect non scripts documents, either

directories or data files.

/var/www/Tequila/Sessions/ is the place where sessions files will reside by default.

The server needs a little more files :

```
/var/www/cgi-bin/tequila
/var/www/cgi-bin/margarita
```

The main script and an administration script useful to setup a new server on tune it up.

```
/var/www/cgi-bin/Tequila/AuthConnector.pm
/var/www/cgi-bin/Tequila/DataConnector.pm
/var/www/cgi-bin/Tequila/LdapAuthConnector.pm
/var/www/cgi-bin/Tequila/LdapDataConnector.pm
/var/www/cgi-bin/Tequila/MysqlDataConnector.pm
/var/www/cgi-bin/Tequila/NullAuthConnector.pm
/var/www/cgi-bin/Tequila/NullDataConnector.pm
```

Various connectors used by the server to authenticate users and fill user attributes.

There is also a set of images, styles and doc files.

Configuring

i. Client

There is not much to do to configure the client. If you have standard Apache and Perl installation, Tequila should work out of the box.

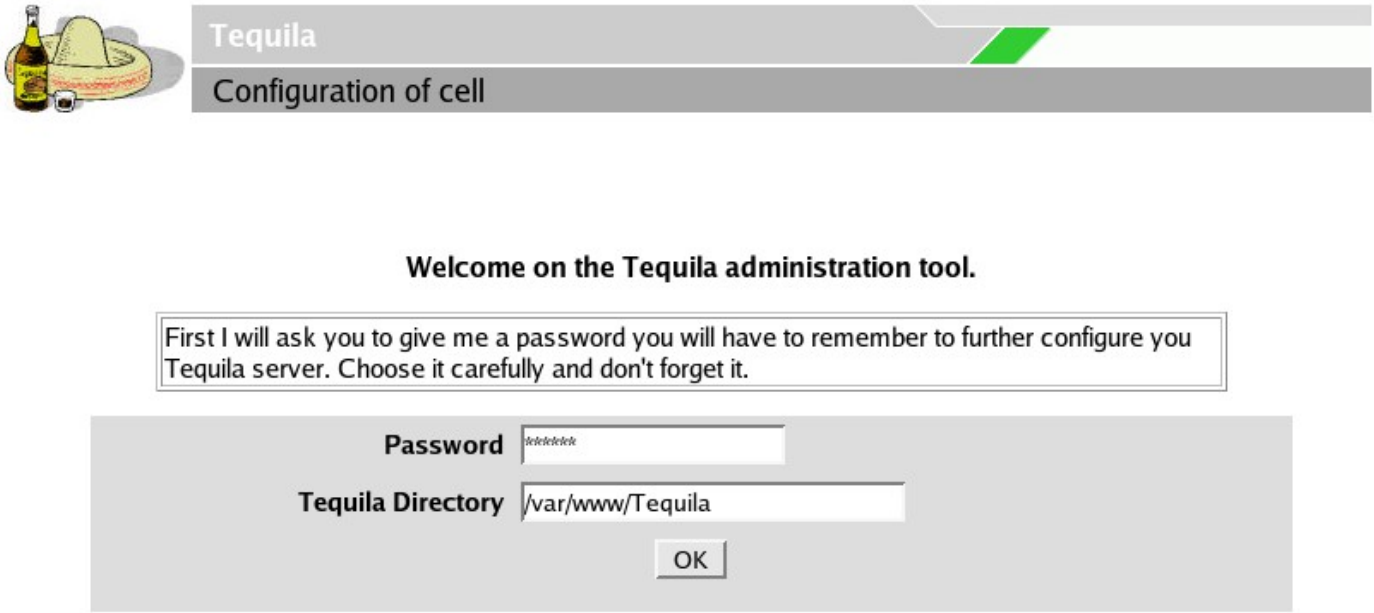
/var/www/Tequila/Sessions/ is the place where sessions files will reside. Tequila tries to locate its server by itself, it will first try to find what the local domain is, and use 'tequila.l*ocaldomain*'. If you are not willing to use these defaults, you can create a file called /etc/tequila.conf' with the following content :

```
#
#
TequilaServer:your_tequila_server
#
SessionsDir:your_sessions_directory
#
```

ii. Server

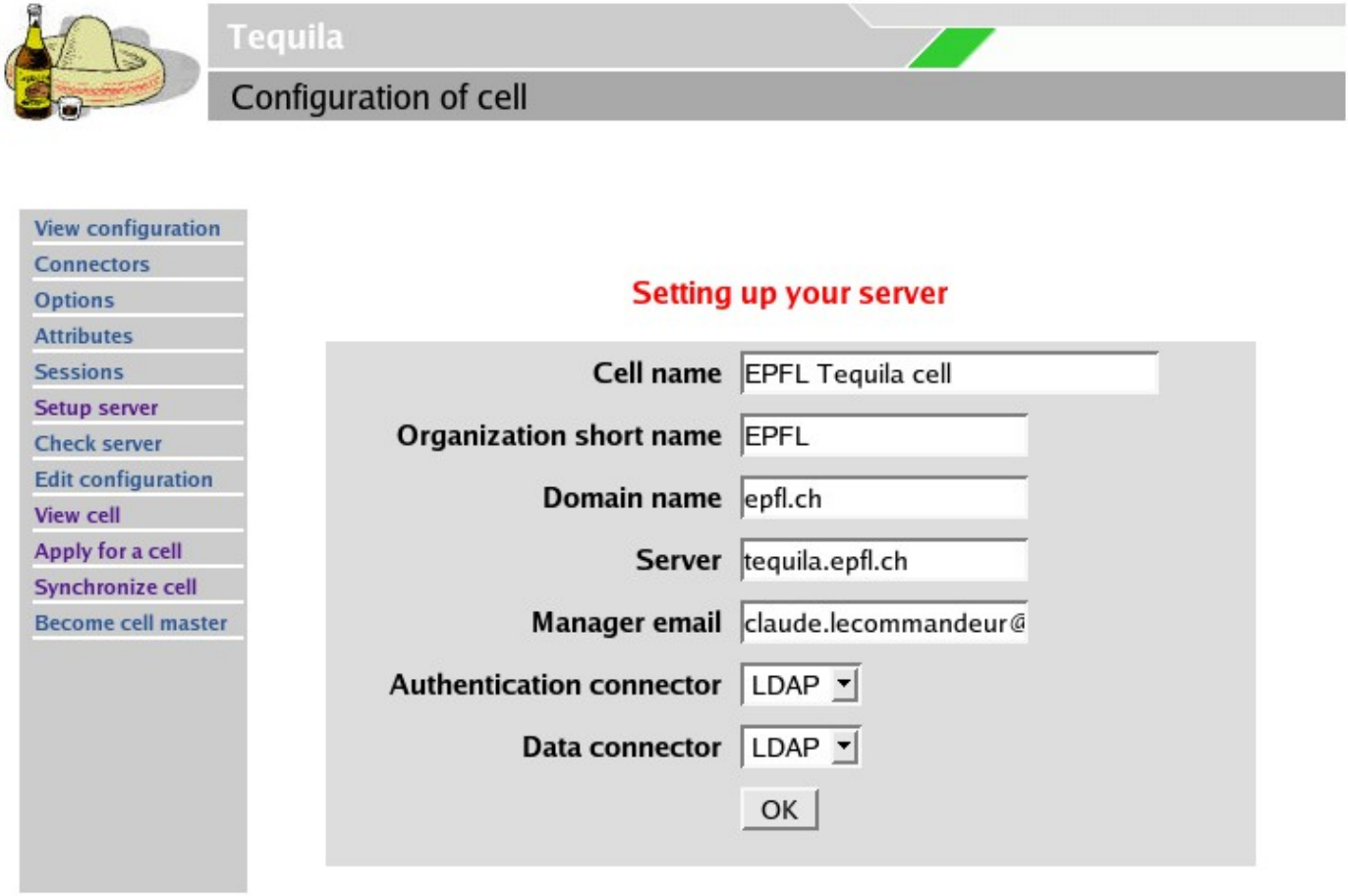
Configuring the server is not hard too with the **margarita** tool. After installing the client and the server rpm file, go to your favorite browser and fetch ***your_server*/cgi-bin/margarita.**

First, it will ask you for an initial password. It is important that you configure your server asap after installing it, since any password will do :



Second, it will ask you if you want to setup a new cell, or join an existing cell :

The first time, you will probably choose a new cell. Now we go in the configuration :



- **Cell Name** : Any string will do, but this should be something people can understand.

- **Organization short name** : The acronym of your organization, no spaces or funny characters.

- **Domain name** : Your Internet domain name. A single Tequila server cannot serve more than one domain.

- **Server** : The server name. The default is the name as given by 'hostname'.

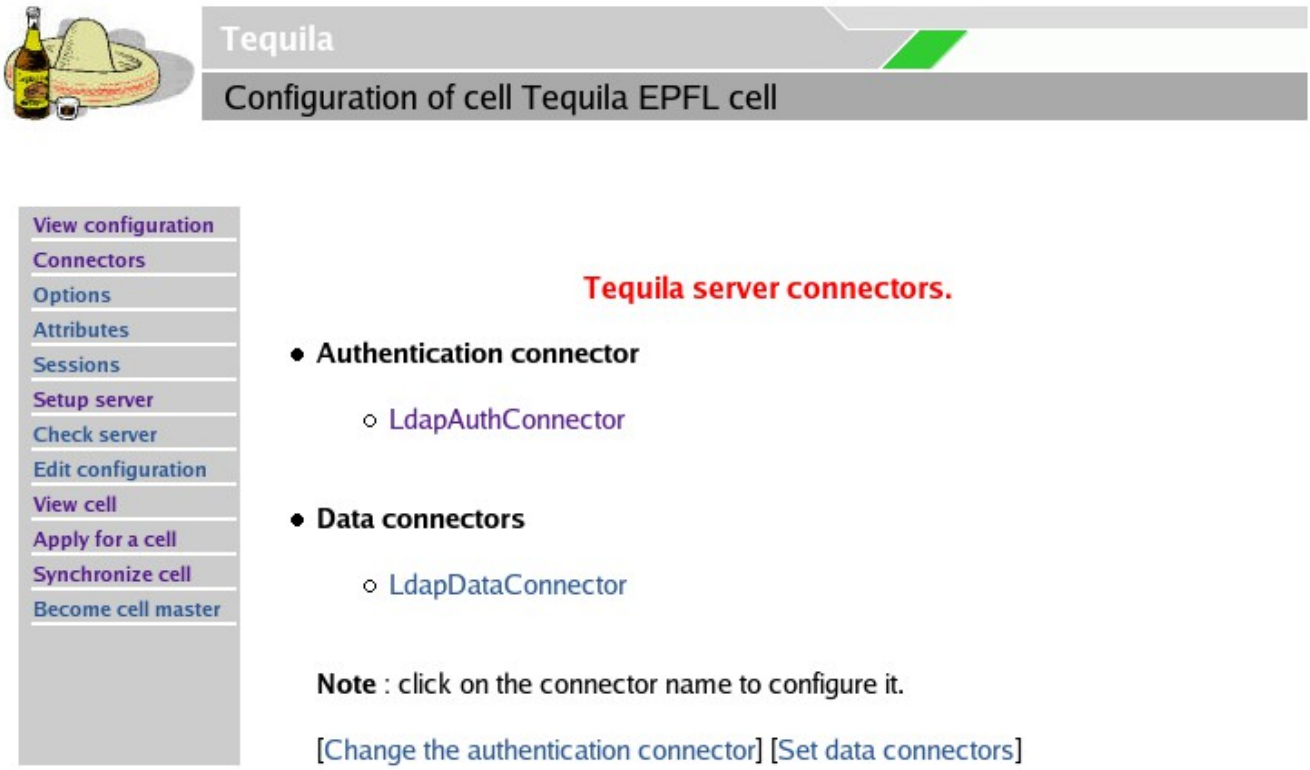
- **Manager email** : Probably your address.

- **Authentication connector** : Choose what kind of server you will use for authentication, right now, only LDAP is supported.

- **Data connector** : Choose what kind of server will hold users data. At this stage, only one data connector is possible, but later on, you can use any number of data connectors. Right now, only LDAP is supported.

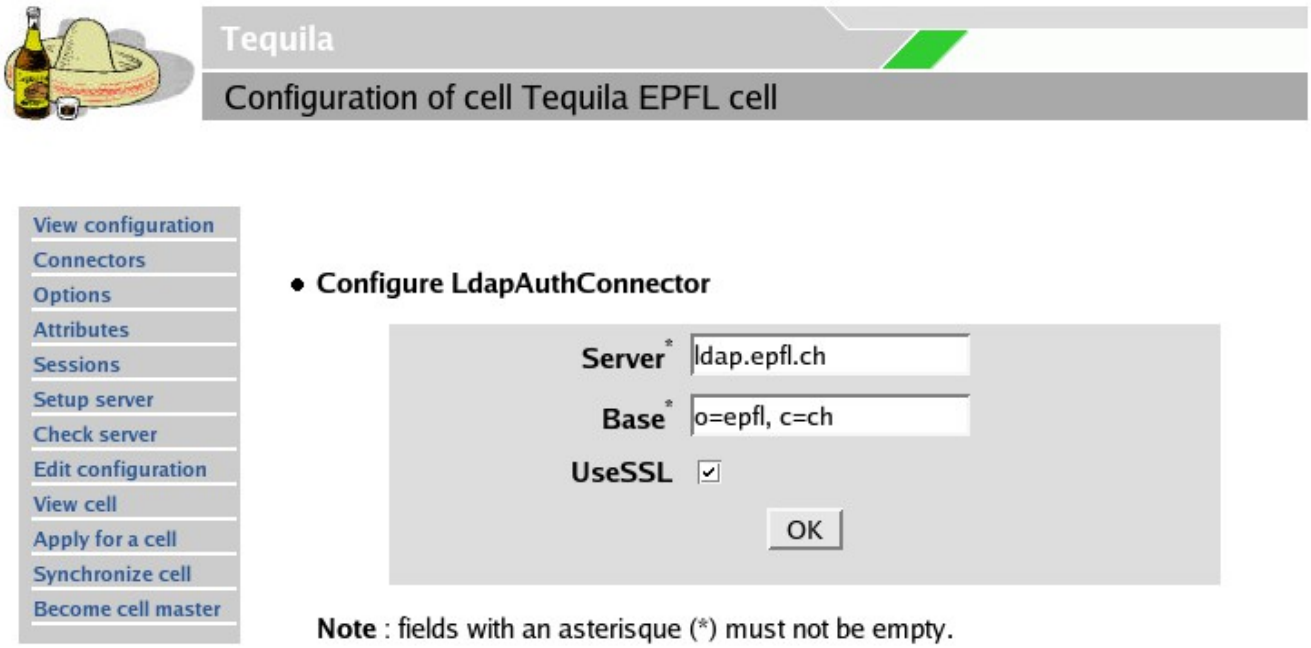
Configuring connectors :

Click on the '**connectors**' command :



At this stage, you can change the authentication connector, set the data connectors (remember that you can have several data connectors if you users data are in several databases.

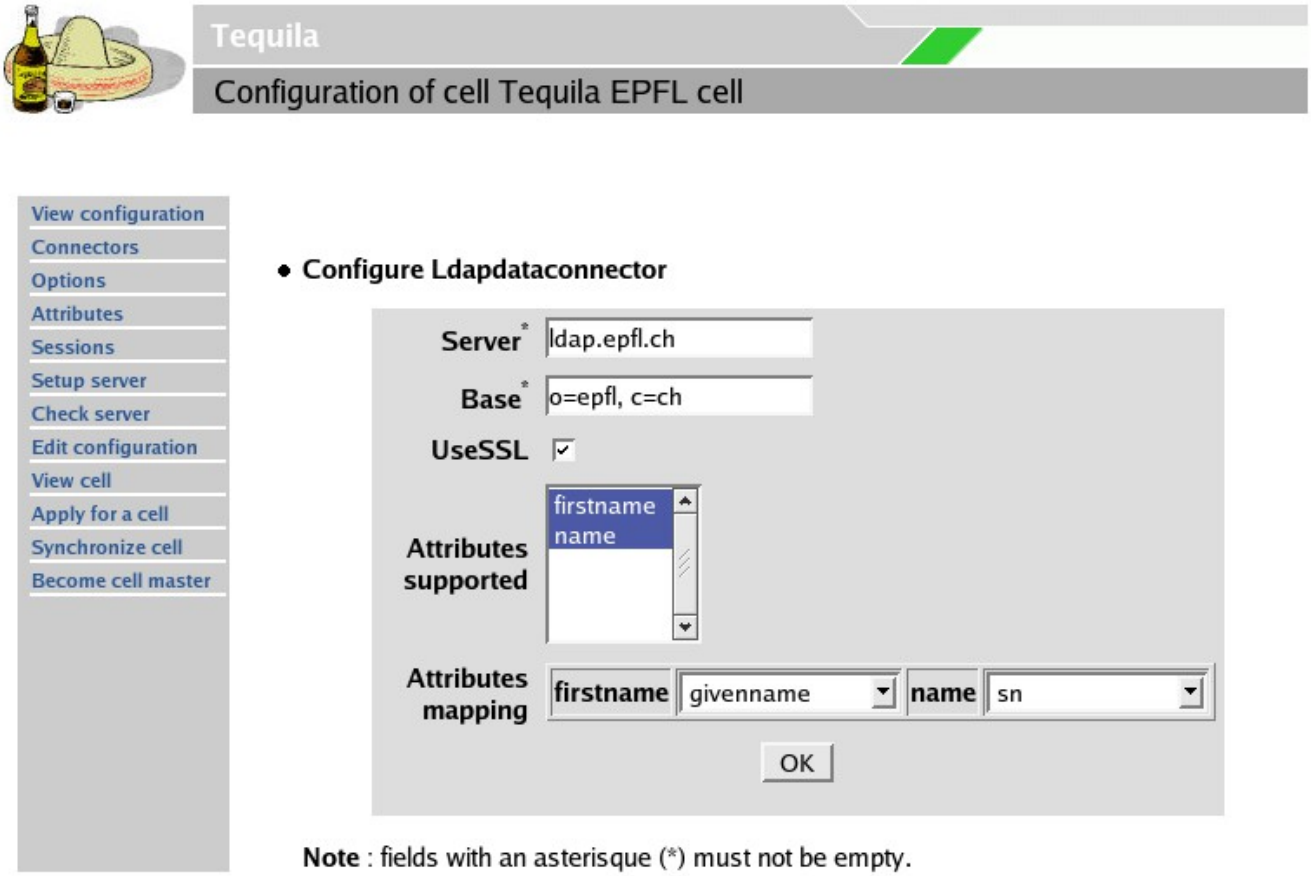
iii. LDAP authentication connector :



You just have to configure :

- the name of your LDAP server, generally **ldap.*your_domain***
- your search base, sometimes '**o=*your_organization*, o=*your_country***', and sometimes '**dc=*your_domain***'.
- Whether you want or not to use SSL (LDAPS protocol) to speak to your LDAP server. I advise you do use it.

iv. LDAP data connector

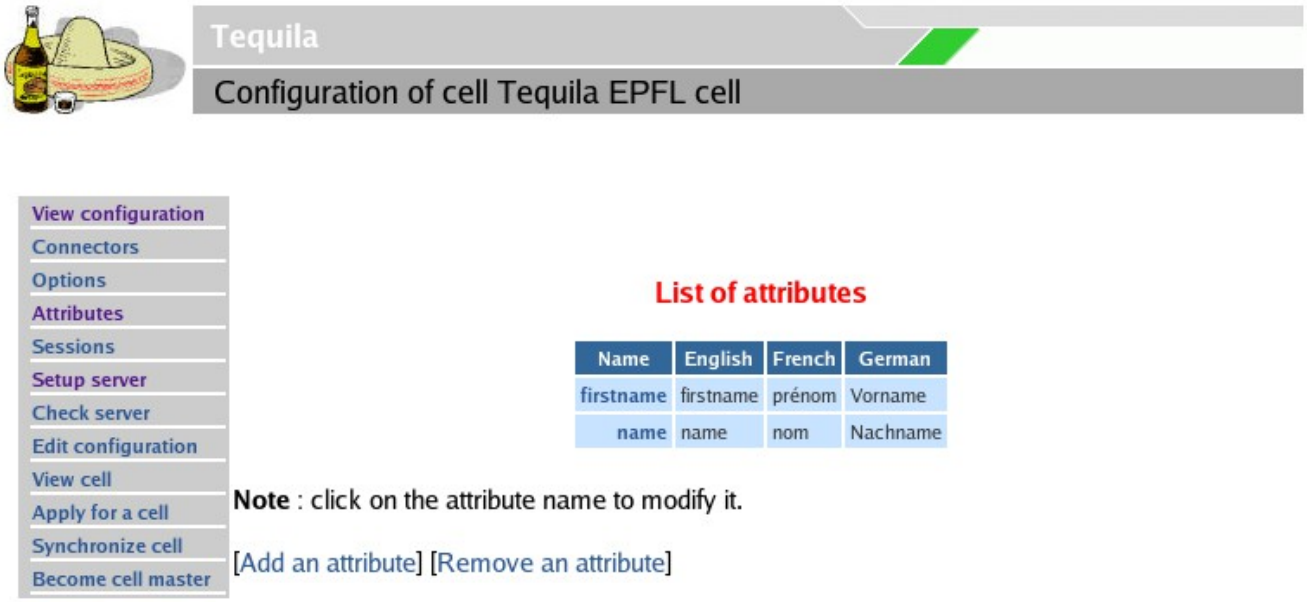


The 3 first fields are the same as for the LDAP authentication connector, but they have not necessarily the same values, you can authenticate on e server and fetch users data on another.

For each server supported attribute, you must tell whether this particular connector supports it, and which LDAP attribute value represents it.

Configuring attributes :

Click on the '**Attributes**' command :



Ath this stage, you can add, or suppress an attribute. For an attribute to be actually usable, it must be supported by a connector. So, if you add an attribute, you have then to configure a connector to support it.

Synchronize your cell data with the master

Use the '**Synchronize cell**' command, and this is done.

Files

If you prefer, you can directly configure files instead of using **Margarita**.

v. Tequila.conf

Each line has the form :

keyword: *value* [*value value* etc...]

Comments begin with a sharp character.

Possible keywords are :

- **CellName***: cell_name*

  The name of your cell. Mandatory.
- **Organization** :

  The name of your organization. Mandatory.
- **Server**: *server_name*

  The name of the Tequila server. Mandatory.
- **Domain**: *domain_name*

  The name of the local Internet domain. Mandatory.
- **Status**: *status*

  Either *master* or member. M *mandatory.*
- ***ServerManager****: email_address*

  The email address of the local server manager. Mandatory.
- **UseCertificates***: value*

  If *value* is 'on', tequila will propose SSL client authentication to its clients. In this case remember to have tequilac linked to tequila in your server cgi-bin.
- **UseCookies***: value*

  If *value* is 'optional', Tequila will propose to use a cookie for further authentication. If value is 'on', Tequila will use cookies unconditionally. This avoids unnecessary login when you are confident about your host security. Remember you need the Crypt::RC4 Perl module on the server.
- **AuthConnector** *: value*

  The authentication connector. Value is the name of the perl module that implements the authentication connector. All authentication connectors must inherit from the AuthConnector module. Only one AuthConnector is allowed. Mandatory.

- **DataConnector** : *value*

  The data connector. Value is the name of the perl module that implement the data connector. All data connectors must inherit from the DataConnector module. Multiple DataConnector can be used. Mandatory.

- **DefaultLanguage** :

  The default language of the Tequila interface at your organization. Possible values are francais, deutsch and english. Optional, default is english.

- **Attribute** :

  Describes the user attribute this Tequila server is managing. The format is :

  Attribute: *attribute_name English_name French_name German_name*

  Use as many **Attribute** line as you need for all the supported attributes.

Example :

```
#
# Tequilaserverconfiguration.
#
CellName:EPFL Tequilacell
#
Organization:EPFL
#
Status:master
#
Server:tequila.epfl.ch
#
Domain:epfl.ch
#
ServerManager:claude.lecommandeur@epfl.ch
#
AuthConnector:LdapAuthConnector
#
DataConnector:LdapDataConnector
#
DefaultLanguage:francais
#
#        Attr      English      French       Deutsch
#
Attribute:name        name          nom          name
Attribute:firstname   firstname     prénom       firstname
Attribute:email       email         email        email
Attribute:title       title         fonction     title
Attribute:unit        unit          unité        unit
Attribute:office      office        bureau       office
Attribute:phone       phone         téléphone    phone
Attribute:username    username      username     username
Attribute:uniqueid    username      uniqueid     uniqueid
Attribute:unixid      unixid        unixid       unixid
Attribute:groupid     groupid       groupid      groupid
#
```

vi. orgs

The list of all organizations known to this Tequila server. One organization per line.

Example :

```
#
# Members of the Tequila network.
#
FOO:slpc1.epfl.ch:claude.lecommandeur@epfl.ch:member
EPFL:tequila.epfl.ch:epfl.ch:claude.lecommandeur@epfl.ch:master
#
```

vii. passwd

The crypt'ed administrator's password.

viii. LdapAuthConnector.conf

Configuration of the LDAP authentication connector. The format of the lines is :

keyword: *value*

The possible keywords are :

- **Server** : Name of the LDAP server. Mandatory.
- **Base** : Search Base of the server. Mandatory.
- **UseSSL** : Use or not LDAPS to authenticate. Possible values are 'on' and 'off', 'on' is recommended. Mandatory.

Comments begin with a sharp character.

Example :

```
#
# Configuration of the LDAP authentication connector for Tequila.
#
Server:ldap.epfl.ch
Base:  o=epfl,c=ch
UseSSL:on
#
```

ix. LdapDataConnector.conf

Configuration of the LDAP data connector. The format of the lines is :

keyword: *value [value value etc...]*

The possible keywords are :

- **Server** : Name of the LDAP server.
- **Base** : Search Base of the server. Mandatory.
- **UseSSL** : Use or not LDAPS to authenticate. Possible values are 'on' and 'off', 'on' is recommended. Optional.
- **Supports** : List of attributes supported by this connector. Mandatory.
- **Mapping** : Which LDAP attribute correspond to which Tequila attribute. Mandatory.

Comments begin with a sharp character.

Example :

```
#
# Configurationof the LDAP Data connectorforTequila.
#
Server:ldap.epfl.ch
Base:  o=epfl,c=ch
UseSSL:off
#
Supports:name firstnameemailtitleunitofficephone \
        username uniqueidunixidgroupid
#
Mapping: name        sn
Mapping: firstname   givenname
Mapping: email       mail
Mapping: title       title
Mapping: unit        ou
Mapping: office      roomNumber
Mapping: phone       phone
Mapping: username    uid
Mapping: uniqueid    uniqueIdentifier
Mapping: unixid      uidnumber
Mapping: groupid     gidnumber
#
```

x. rc4key

Just the key itself. About 8 characters is enough. Necessary only if you want to use the cookie option.

xi. privkey

The X509 private key of the server. Necessary only if you are using the RSA option.

xii. keys/*.key

The X509 public key of the servers in the Tequila network. Necessary only for servers using the RSA option. The name of the file is : *orgname.*key

Client Interfaces

Tequila offers several interfaces to services willing to authenticate their users. Presently, Perl scripts and Java servlets have special support, but it is always possible to use the raw interface in any language.

The Perl interface

The Perl interface is just one perl module : Tequila::Client.pm. It manages sessions for the application.

xiii. Methods

constructor

The constructor.

authenticate

The main entry. This method tests whether there is a valid session opened for this user. If yes, it fills its attributes with the user attributes values and returns, else, it does the URL redirect to the local Tequila server and exits (take care, in this case it doesn't return).

opensession

Opens a new session. Uses the CGI QUERY_STRING or an array or keys as argument to fill the session attributes.

loadsession

Loads an existing session. The unique argument is the session key.

purgesessions

Purge all timed out sessions.

killsession

Remove a session. The unique argument is the session key.

request

Tells Tequila which attributes the service wants to know about the user. The arguments are the list of attribute names. The use must accept to release these attributes or refuse to authenticate.

wish

Tells Tequila which attributes the service wishes to know about the user. The arguments are the list of attribute names. The user can choose which of these attributes she wants to release, but she knows they are not mandatory.

wantright

Telle Tequila to require the user has this specific right to accept authentication. This is not supported by all servers. If the authentication succeeds, the application is returned the list of units the user has the right for, and the list of the units the user is administrator for this right.

wantrole

Telle Tequila to require the user has this specific role to accept authentication. This is not supported by all servers. If the authentication succeeds, the application is returned the list of units the user has the role for.

require

You can ask the Tequila server for a filter to be satisfied instead of requiring attributes values. Example : firstname=claude&org=EPFL. This server feature is mainly used by the mod_tequila Apache module.

setlang

Set the language of the interface. The 3 possible values are : 'francais', 'deutsch' and 'english',

usersa

Tells Tequila to store all the user attributes in the *urlaccess* and signs it with its RSA private key. *Urlauth* is not needed in this case. This can be considered as an on the fly certificate.

setdirectask

Whether Tequila should directly propose the login screen to the user (argument is true) or a message telling her she must authenticate.

setserver

Sets the home Tequila server. Normally this value is read from a configuration file.

setorg

Sets the home organization of the server. It alse sets the server (either to the value set in the configuration file, or to "tequila.*org_name*.ch".

setservice

Sets the service name.

setsessionsdir

Sets the session directory. Normally /var/www/Tequila/Sessions/.

setsessionsduration

Sets the duration of the sessions. Default is 10 minutes.

xiv. Instance variables :

Generally, most if not all of these variables are set automatically, read this only of necessary.

urlauth

The URL where the Tequila server should go to open the session on the client.

urlacces

The URL where the Tequila server should redirect the client browser in case of successful authentication.

service

Service name. The Tequila server displays it to the user to inform her.

request

The list of user attributes requested.

wish

The list of user attributes wished (the user is not forced to give them).

require

The filter that Tequila must the user verifies.

language

Language of the user interface.

localserver

Local server name.

sessionsdir

Session directory.

sessionmax

Session duration.

directask

Whether the user is automatically redirected to the Tequila server, or whether she is asked before.

usersa

Does the client requests the server to stuff all the attributes values in a signed URL, or does it create a session with *urlacces.* If 'on', *urlacces* is not necessary.

Key

The authentication key generated by the server.

Org

The organization that did the actual authentication. The client script can read this to see where the user is coming from.

User

The user name. The actual user can be considered as *user@org*

Host

The host the user is coming from. Beware it can be a proxy.

Attrs

The array of attributes. All requested and possibly wished attributes are keys of this array, the values are the values of these attributes.

xv. Example

```
use Tequila::Client;

my $tequila= new Tequila::Client();
$tequila>setService('Tequilatest');
$tequila>request("name","firstname","unit","where");
$tequila>setDirectAsk();
$tequila>setOrg('EPFL');
$tequila>useRSA(1);
$tequila>authenticate();

my  $org= $tequila>{Org};
my $user= $tequila>{User};
my $host= $tequila>{Host};
```

The module interface

The mod_tequila Apache module is designed to protect non script data, files and/or directories (whole trees), or even a whole external WWW server.

It is an Apache module, so you configure it in your *httpd.conf*. The module doesn't require any attribute from the server, instead it asks the server whether a particular filter is satisfied. A filter is a set of conditions acting over the user attributes.

xvi. Module commands:

Global commands :

TequilaLogLevel value

The more *value* is the more log you get.

TequilaLog *filename*

The name of the log file. Default is /etc/httpd/logs/tequila.log

TequilaServer *server_name*

The name of the local Tequila server. Default is tequila.*local_domain*.

TequilaSessionDir *directory*

The name of the sessions directory. Default is /var/www/Tequila/Sessions.

TequilaSessionMax *value*

Sessions duration in seconds. Default is 3600 (1 hour).

Location specific commands :

TequilaRewrite *new_location*

The current location will be rewritten as *new_location* in case of successful authentication (and the filters are satisfied).

TequilaAllowIf *condition*

The condition must be met to give access to the current location. The syntax of the condition is :

attribute=*value*&attribute=*value*&...

The condition will be met if and only if all the user's attributes match the given values. You can give as many TequilaAllowIf directives as you want for each location. All these directives will be or'ed, that means that if only one condition need to be met for access to be given.

If the equal ('=') is replaced with '=~', pattern matching is done instead of strict equality. In any case the comparison is done ignoring case.

Example :

```
AddModulemod_tequila.c
```

```
<IfModulemod_tequila.c>
  TequilaLogLevel     2
  TequilaLog          /etc/httpd/logs/tequila.log
  TequilaServer       tequila.epfl.ch
  TequilaSessionDir   /var/www/Tequila/Sessions
  TequilaSessionMax    3600


  <Location/restricted/oscar/>
    TequilaRewrite/var/www/restricted/oscardoc/
    TequilaAllowIffirstname=claude&name=lecommandeur
    TequilaAllowIforg=EPFL
  </Location>
```

This means that the location /restricted/oscar/ will be protected by **Tequila**. Only users of EPFL and Claude Lecommandeur will have access to it. If access is given, files will be given from /var/www/restricted/oscar/. For example :

http://server.epfl.ch/restricted/oscar/Oscar.html will fetch

/var/www/restricted/oscar/Oscar.html

Beware to use only relative links in all html files in /var/www/restricted/oscardoc/ if you want to have the access rights to be inherited.

```
  <Location/restricted/www/>
    TequilaRewritehttp://www.epfl.ch/
    TequilaAllowIforg=EPFL
    TequilaAllowIforg=FOO
  </Location>

</IfModule>
```

In this example, the access is checked for a whole remote site (http://www.epfl.ch/). Only people from EPFL or FOO will be allowed to access. In this case, it is even more difficult to have only relative links in all the documents on the remote server (this is not the case on the server in this example).

The Java Interface

The java interface is designed to be used by Java servlets. It contains a class : Tequila.Client, and a very little servlet : Tequila.OpenSession.

The Java interface is not complete.

xvii. Specification of class Tequila.client.

Constructor :

Takes 2 String arguments urlauth, urlacces, same as the Perl module.

Authenticate (HttpServletRequest request, HttpServletResponse response)

Same as Perl module.

TODO : finish the list.

xviii. Example

```
publicclassTequilaTestextendsHttpServlet{
   publicvoiddoGet(HttpServletRequestrequest,HttpServletResponseresponse)
                        throwsIOException,ServletException{

      int port= request.getServerPort();

      int port= request.getServerPort();
      Stringus = request.getServerName();
      Stringme = request.getServletPath();
      Stringpi = request.getPathInfo();
      Stringcp = request.getContextPath();
      if(pi== null)pi = "";
      Stringurlauth= "http://"+us+ ":"+port+ "/examples/servlet/OpenSession";
      Stringurlacces= "http://"+ us + ":"+ port+ cp + me + pi;
      String    key = request.getParameter("key");
      Client tequila= new Client(urlauth,urlacces);

      tequila.setService("TequilaTest");
      tequila.setSessionDuration(7200 * 1000);
      tequila.request("name");
      tequila.request("firstname");
      tequila.request("email");
      tequila.request("unit");

      response.setContentType("text/html");
      PrintWriterout = response.getWriter();
      out.println("<html>");
      out.println(" <head>");
      out.println("   <title>TequilaTest</title>");
      out.println(" </head>");
      out.println(" <body>");

      Sessionsession= tequila.authenticate(request,response);
      if(session== null)return;

      out.println("<h3>TestTequila:</h3>");
      out.println("<pre>");
      out.println("     org = " + session.org);
      out.println("    user = " + session.user);
      out.println("    host = " + session.host);
      Enumerationattrs= session.attrs.keys();
      while(attrs.hasMoreElements()){
```

```
            Stringattr= (String)attrs.nextElement();
            out.println(attr+ " = " + session.attrs.get(attr));
        }
        out.println("</pre>");
        out.println(" </body>");
        out.println("</html>");
    }
}
```

The raw interface

A client can directly access the Tequila services via HTTP redirection. But it must manage sessions itself. The client just has to redirect the user browser to a specially coined URL of the form :

https://*tequilahost*/cgi-bin/tequila/auth?urlauth=*urlauth*&urlacces=*urlacces*&service=*service*&usersa=*1*&request=*request*&language=*language*

Only urlacces is mandatory, generally this is the URL of the client itself.

If *usersa* if not there (or empty or 0) urlauth is mandatory too.

```
AddModulemod_tequila.c
<IfModulemod_tequila.c>
  TequilaLogLevel     2
  TequilaLog          /etc/httpd/logs/tequila.log
  TequilaServer       tequila.epfl.ch
  TequilaSessionDir   /var/www/Tequila/Sessions
  TequilaSessionMax    3600

  <Location/restricted/oscar/>
    TequilaRewrite/var/www/restricted/oscardoc/
    TequilaAllowIffirstname=claude&name=lecommandeur
    TequilaAllowIforg=SWITCH
  </Location>
```

This means that the location /restricted/oscar/ will be protected by **Tequila**. Only users of SWITCH and Claude Lecommandeur will have access to it. If access is given, files will be given from /var/www/restricted/oscardoc/. For example :

http://server.epfl.ch/restricted/oscar/Oscar.html will fetch

/var/www/restricted/oscardoc/Oscar.html

Beware to use only relative links in all html files in /var/www/restricted/oscardoc/ if you want to have the access rights to be inherited.

```
  <Location/restricted/www/>
    TequilaRewritehttp://www.epfl.ch/
    TequilaAllowIforg=EPFL
    TequilaAllowIforg=FOO
  </Location>

</IfModule>
```

In this example, the access is checked for a whole remote site (http://www.epfl.ch/). Only people from EPFL or FOO will be allowed to access. In this case, it is even more difficult to have only relative links in all the documents on the remote server (this is not the case on the server in this example).

Local Authentication

Zenon wants to enter a temple. The warden asks him who he is :

- *Warden* : Who are you ?
- *Zenon* : I am Zenon.
- *Warden* : Prove it. You must go to our local city authority to have you authenticated.

Zenon goes to city authority and asks :

- *Zenon* : I want to prove my identity to the temple warden to enter the temple.

  The city authority has a special device that can read fingerprints and Zenon can rapidly prove his identity.
- *Authority* : So take this special item manufactured only here and hands it to the temple warden, I'll phone him that the holder of this item is actually Zenon and he will let you enter.

Zenon goes back to the temple

- *Zenon* : I am Zenon and the city authority gave me this (he shows the item).

The warden looks at the item, looks in his list of authorized people whether Zenon is in it and reply.

- *Warden* : Ok, you can enter, I have been phoned that you were coming.

Remote authentication

Now Zenon wants to enter to a temple that is not in his town :

- *Warden* : Are you a good mathematician ?
- *Zenon* : Yes, I am.
- *Warden* : Prove it. You must go to our local city authority to have this proved.

Zenon goes to city authority and asks :

- *Zenon* : I want to prove to the temple warden that I am a good mathematician to enter the temple.
- *Local authority* : You are not from this city, and we don't know you here, goto the certification authority of your city and ask them.

Zenon goes to his home city authority and asks for the proof.

- *Home authority* : Ok, we know you, I know this temple warden, take this special item, I'll phone him that the holder of this item is a good mathematician. He will let you enter.

RSA signature

Same as the first scenario, but the local authority has not yet been installed the phone. The local city authority signs a paper that certifies that the holder is Zenon.

This is also valid for remote certification.