Enable log-out from Apache 2.4

Task and problem

We have a multi domain server running Apache 2.4 on Ubuntu 16.04. Multiple applications and information services are offered to users, one of them being SVN. The services require TLS / HTTPS and logging-in with user:password.

There are cases where users, from the server's point of view, have multiple identities, i.e. multiple user:password pairs. They need those to get access to different information spaces (directories in the end), different SVN repositories and so on.

The problem shouldn't exist

The first and correct objection*) is that this schizophrenia should not imposed to users as persons. Eventually, the semantics of these multiple identities are groups with certain rights
a) the user should be (made) member of and
b) group rights should be applied to him **) by the application.
Here b) is no problem for Apache and a solvable one for SVN.

On the other hand a) requires a functioning ID management, flexible enough to create, handle and delete groups (for projects, lectures, development groups etc.). And you have to have the server running Apache and else having sufficient access to this ID management.

More often than not at least one of these conditions fail or require undue bureaucracy. In consequence, we see those pseudo users (semantically meaning a group) very often in organisations of all kinds and sizes. And we here fall back to this "french plumbing" en lieu de solid ID management, too.

Having human users having multiple IDs in our servers has a prize: A user changing development projects, lectures, infos will have to log-out and log-in as another one to get to the other sub-site or even see it in lists.

Note *): Another point is: The objection is valid from an Windows NT (and beyond; some would say Posix) point of view. Unix/Linux has no usable groups. There are no groups in groups. Being user1 in group1 and considering a file owned by user1 and group1 070 gives me less rights than 700 [sic!]. Those architectural bugs, never mended but considered as sacred by over half century age, also lead to the organisational fault of miss-using user as group. And Linux' Posix add-ons are just add-ons – and evidently not in wide use.

Note**): Where ever He or Him is used in this text or in [27] [29] read She or Her.

The login example

Let's look at a real example, also using notorious user-as-group approach. Your environment is different, but this example will most probably fit for this reports topic in most cases.
Here's the first excerpt from /etc/apache2/sites-available/weAut_ssl.conf:

```
<VirtualHost *:443>
  ServerName weinert-automation.de
  # omitted basic configurations of:  ServerAdmin,  DocumentRoot,
  #  ErrorDocument, ServerAlias, SSLEngine  .... etc. pp.
  <IfModule mod_authnz_external.c>
    AddExternalAuth pwauth /usr/sbin/pwauth
    SetExternalAuthMethod pwauth pipe
    AddExternalGroup unixgroup /usr/sbin/unixgroup
    SetExternalGroupMethod unixgroup environment
  </IfModule>

# infos needs no authentication; subfolders get stricter by .htaccess
```

```
  <Directory /var/www/sites/weAut/infos>
    Options +FollowSymLinks
    IndexOptions +ShowForbidden
    AllowOverride FileInfo AuthConfig Indexes

    AuthType Basic
    AuthName "weAut restricted"
    AuthBasicProvider external
    AuthExternal pwauth
    GroupExternal unixgroup
    Require all granted
  </Directory>

# userInfos needs any valid authentication
  <Directory /var/www/sites/weAut/userInfo>
    IndexOptions +ShowForbidden
    AllowOverride FileInfo AuthConfig Indexes

    AuthType Basic
    AuthName "weAut restricted"
    AuthBasicProvider external
    AuthExternal pwauth
    GroupExternal unixgroup
    Require valid-user
  </Directory>

 <Location /svn>
  DAV svn
  SVNParentPath /var/www/repos
  SVNListParentPath on
  SVNIndexXSLT /conf/svnindex.xsl

  AuthType Basic
  AuthName "weAut restricted"
  AuthBasicProvider external
  AuthExternal pwauth
  GroupExternal unixgroup
  require valid-user
  # Enable authorisation via mod_authz_svn (used "by repo", only):
  <IfModule mod_authz_svn.c>
    AuthzSVNAccessFile /etc/apache2/dav_svn.authz
  </IfModule>

 </Location>
</VirtualHost>
```

Obviously, besides SVN, we have two information sub-sites:
 directory info
 directory userInfo

The first one lets anybody in by "Require all granted";
the second one lets any authenticated user in by "require valid-user".

By "AllowOverride ...AuthConfig .." we give subdirectories stricter access rules and access to different users and groups in the particular .htaccess file.

Without "IndexOptions +ShowForbidden" logged-in users wouldn't even get sub-folders listed, they haven't actually access to.

Here in our example, users and groups are those of the OS, i.e. of Ubuntu 16.04 by the lines:

```
   AuthBasicProvider external
   AuthExternal pwauth
   GroupExternal unixgroup
```

And we have some 20 SVN repositories under locaction svn, which Apache can nicely list as an extra. In our case we let any authorised users in by "require valid-user" and make the restrictions on a "by repository" basis in the file dav_svn.authz. This, alas, is an extra SVN configuration text file, but thanks to its simple syntax, its handling can be automated by some bash and Java (Frame4J) acrobatic.

Hence, having a user with multiple IDs (user:password) pairs, being logged in for a development project, would have to log-out to be able to log-in for another project or information service.

There's no logout

Well, there's no login either. As https is a stateless protocol, there is (basically) no such thing from our Apache server's point of view. What we experience as being logged in is done solely by the browsers and their remembering a set of login credentials entered by the user. The browser resends them every time the user refreshes the page or navigates to another one.

As the log-in is handled by the browser, one has to make the browser log-out. On most browsers stopping the programme or going to a set up menu and ordering a "forget past" will do so. Neither can be expected of nor imposed to user just wanting to do his work.

Recipes to log out depending on certain browser behaviour

There are many log out recipes to be found, some of them really clever and complicated. Almost all use a special name:password pair – mostly babtised "log:out" for obvious reasons. And most recipes give the user a button or link labelled "logout" going to "https://log:out@...".

Variant a: log:out non existing

Here going to "logout" effectively goes to an error page, from where the user has to go where he wanted to and then log in there.

Landing on an error page annoys the user and is considered as a site bug. To avoid this, these recipes involve most complicated conditional rewritings of the error page. This makes those solutions hardly comprehensible nor maintainable and lastly unreliable.

The recipe's basic assumption is:
Going to a page with a non-existing user and coming back will make the browser log-out.
This seems true for most browsers, but by no way guaranteed behaviour.

Variant b: log:out existing but no rights

Here going to the "logout" link effectively will bring the user to the page href'd, from where he may navigate back or to whatever page by clicking links or by automatic re-direction on time-out.
Using an existing user will change the browsers set of credentials.

But now we are logged in again and going back to where "valid-user" was O.K. reveals no difference to the state before. There's no log-out effect in the sense no one logged in.

Variant c: log:out existing, target page in other realm

The alert reader will have noticed our two info sub-sites as well as the SVN repos being in the same realm. This has the benefit of single-sign on.

Putting the log-out page of variant b in another "log-out" realm, we hope to effectively log-out from the productive realm just used.

Well, this hope may not die in certain cases. But browsers and even browser versions behave differently here. Some keep respectively cache just one set of credentials. Some keep one per server, some keep one per base URL (even when pointing to the same IP) and some do keep one per realm.

In the latter case with variant c we did no log-out at all but have created another log-in living ever after until the browser dies. This behaviour, by the way, is an extra reason to stick with one and only one realm when having no strong reasons against.

The solution

A solution falling in category b) – existing user, same realm, no rights – seems the way to go, promising to avoid all the browser dependencies named. The problem with variant b) was log:out being now a "valid-user" wherever such is asked for. This counteracts the log-out semantics.

But that can easily be cured by an extra directory for the log-out page.
We supplement /etc/apache2/sites-available/weAut_ssl.conf with:

```
#logout needs log:out
  <Directory /var/www/sites/weAut/logout>
    # Options +FollowSymLinks +SymLinksIfOwnerMatch
    IndexOptions +ShowForbidden
    AllowOverride FileInfo AuthConfig
    AuthType Basic
    AuthName "weAut restricted"
    AuthBasicProvider file
    AuthUserFile extraUsers
    Require user log
  </Directory>
```

We do not make log (password out) an Ubuntu user, but jail himand only him (!) in an extra Apache user file (we may have an arbitrary number of) by

```
htpassed  -c /etc/apache2/extraUsers log
```

After gotten to know it, the trick is obvious:

- log:out is a valid user allowing log-in to the log-out page.
- As using the same realm he really logs out any other user there.
- log:out is not in the same domain as used for all other services.

In our exemplary configuration the last point says: log:out isn't a Ubuntu user and, hence, won't be recognised as valid user in all other directories or services.

Resume

We have solution for the Apache log-out problem avoiding browser dependencies and complications seen with many other recipes.

For the basic Apache 2.4 on Ubuntu 16.04 installation used here as example, please see [29].

The log-out solution by itself presented here depends not on Ubuntu as base for Apache and SVN.

References and Abbreviations

Please find those in

[29] Albrecht Weinert, Ubuntu for remote services, Report, November 2016,
(the full story): a-weinert.de/pub/ubuntu4remoteServices.pdf

We keep those in one (the biggest Ubuntu server) document for consistency, so far.
Also, see the policy on "Using names" there.

If you see this in print you may look at

[27] Albrecht Weinert, Enable log-out from Apache 2.4, Report. March 2017,
This paper (actual version): a-weinert.de/pub/enableApache24logout.pdf

Table of Content

Dr. Albrecht Weinert is computer science professor at
Bochum University of Applied Sciences or Hochschule Bochum.
He is founder and director of
MEVA-Lab – Laboratory for versatile distributed applications –
as well as of the service provider weinert – automation.
albrecht@a-weinert.de

Rev. 03 01.08.2017