

# BackupPC

From ArchWiki

**BackupPC** is a high-performance, enterprise-grade system for backing up Unix, Linux, WinXX, and MacOSX PCs, desktops and laptops to a server's disk. BackupPC is highly configurable and easy to install and maintain.

Given the ever decreasing cost of disks and raid systems, it is now practical and cost effective to backup a large number of machines onto a server's local disk or network storage. For some sites this might be the complete backup solution. For other sites additional permanent archives could be created by periodically backing up the server to tape.

## Contents

- 1 Installation
- 2 Apache configuration
  - 2.1 Edit Apache configuration
    - 2.1.1 General settings
    - 2.1.2 Single-purpose Apache settings
    - 2.1.3 Multi-purpose Apache settings
      - 2.1.3.1 The webserver user and the suid problem
- 3 Alternative nginx configuration
- 4 Alternative lighttpd configuration
- 5 Accessing the admin page
- 6 Website view problem
- 7 See also

## Installation

Install `backuppc` (<https://www.archlinux.org/packages/?name=backuppc>) from the official repositories. Install `rsync` (<https://www.archlinux.org/packages/?name=rsync>) and `perl-file-rsyncp` (<https://www.archlinux.org/packages/?name=perl-file-rsyncp>) if you want to use `rsync` as a transport.

Start **backuppc** `systemd` daemon and, if you wish to have running at boot time enable it.

## Apache configuration

BackupPC has a web interface that allows you to easily control it. You can access it using Apache and `mod_perl` or a C wrapper but other webserver like `lighttpd` (<https://www.archlinux.org/packages/?name=lighttpd>) works too. Install `apache`

(<https://www.archlinux.org/packages/?name=apache>) from the official repositories.

## Edit Apache configuration

BackupPC's web UI needs to run as the user `backuppc`, but Apache normally runs under the user `http`. There are several ways to fix this. The two demonstrated here are common for single-purpose servers (Apache is only used to serve the BackupPC UI) or for multi-purpose servers (Apache may also server other websites under the regular `http` user).

Setting up Apache for single-purpose use is a bit easier but not as flexible.

### General settings

Edit `/etc/backuppc/config.pl`. Set administrator name:

```
-----  
$Conf{CgiAdminUsers} = 'admin';  
-----
```

Next, we need to add a users file and set the admin password:

```
-----  
# htpasswd -c /etc/backuppc/backuppc.users admin  
-----
```

The BackupPC-Webfrontend is initially configured so that you can only access it from the localhost. If you want to access it from all machines in your network, you have to edit `/etc/httpd/conf/extra/backuppc.conf`. Edit the line:

```
-----  
allow from 127.0.0.1  
-----
```

to:

```
-----  
allow from 127.0.0.1 192.168.0  
-----
```

where you have to replace `192.168.0` to your corresponding IP-Adresses you want to gain access from. After one of the configuration steps below has also been performed, [re]start the Apache service.

### Single-purpose Apache settings

Install `mod_perl` ([https://aur.archlinux.org/packages/mod\\_perl/](https://aur.archlinux.org/packages/mod_perl/))<sup>AUR</sup> from the official repositories.

Edit the Apache configuration file to load `mod_perl`, tell Apache to run as user `backuppc` and to include `/etc/httpd/conf/extra/backuppc.conf` :

```
-----  
/etc/httpd/conf/httpd.conf  
-----  
LoadModule perl_module modules/mod_perl.so  
User backuppc  
-----
```

```

Group backuppc
Include conf/extra/backuppc.conf

```

## Multi-purpose Apache settings

Instead of globally changing the Apache user and group like in the example above, we will instead make Apache run just the BackupPC CGI script as the backuppc user and leave the default user alone. This method uses `mod_cgi` to call a wrapper written in C instead of using the extra `mod_perl` dependency. You still need to have `perl` (<https://www.archlinux.org/packages/?name=perl>) itself installed so the wrapper can run the BackupPC scripts.

Make sure Apache can run CGI programs (the line loading `mod_cgi` is not commented) and that it reads the BackupPC configuration by including it in `/etc/httpd/conf/extra/backuppc.conf` :

```

/etc/httpd/conf/httpd.conf
...
LoadModule cgi_module modules/mod_cgi.so
...
Include conf/extra/backuppc.conf

```

## The webserver user and the suid problem

The current setup of BackupPC, the webserver needs to run as backuppc user and this can be a problem on many setups where the webserver is used for other sites. In the past one could suid a Perl script, but it was blocked globally due security problems several years ago. To workaround that, `perl-suid` was used, but again blocked due the same problem more recently, scripts cannot be run securely with suid bit. Still there is another way, this time using a simple binary program that is suid as a launcher, that will run the backuppc Perl scripts already with the correct user. This isolates the Perl script from the environment and it is considered safe.

You need to replace the original backuppc CGI with the below C code compiled program and move the backuppc CGI to another place.

Move the real CGI `/usr/share/backuppc/cgi-bin/BackupPC_Admin` to the lib directory `/usr/share/backuppc/lib/real-BackupPC_Admin.cgi` .

Save the C code below to a file named `wrapper.c` (please update the CGI path if needed) and compile it with:

```

$ gcc -o BackupPC_Admin wrapper.c

```

## The wrapper C code:

```

#include <unistd.h>
#define REAL_PATH "/usr/share/backuppc/lib/real-BackupPC_Admin.cgi"
int main(ac, av)
char **av;
{

```

```
    execv(REAL_PATH, av);
    return 0;
}
```

Place the new binary `BackupPC_Admin` in the `cgi-bin` directory and chown the binary CGI to `backuppc:http` and set the suid bit:

```
# chown backuppc:http /usr/share/backuppc/cgi-bin/BackupPC_Admin
# chmod 4750 /usr/share/backuppc/cgi-bin/BackupPC_Admin
```

Do not forget to clear the suid bit on the original Perl script if it was set (or the CGI page will not load):

```
# chmod 0755 /usr/share/backuppc/lib/real-BackupPC_Admin.cgi
```

Keep your web server with its usual user and backup should now be able to run correctly.

## Alternative nginx configuration

Install `fcgiwrap` (<https://www.archlinux.org/packages/?name=fcgiwrap>). Enable and start `fcgiwrap.socket`.

```
/etc/nginx/sites-available/backuppc
-----
server {
    listen <your_server_port>;
    server_name <your_server_name>;

    root /usr/share/backuppc/html;
    index /index.cgi;

    access_log /var/log/nginx/backuppc.access.log;
    error_log /var/log/nginx/backuppc.error.log;

    location / {
        allow 127.0.0.1/32;
        # allow 192.168.0.0/24;
        deny all;

        # auth_basic "Backup";
        # auth_basic_user_file conf/backuppc.users;
    }

    location /backuppc {
        alias /usr/share/backuppc/html/;
    }

    location ~\.cgi$ {
        include fastcgi_params;
        fastcgi_pass unix:/run/fcgiwrap.sock;

        fastcgi_param REMOTE_ADDR    $remote_addr;
        fastcgi_param REMOTE_USER    $remote_user;
        fastcgi_param SCRIPT_FILENAME /usr/share/backuppc/cgi-bin/BackupPC_Admin;
    }
}
```

And symlink to sites-enabled:

```
# ln -s /etc/nginx/sites-available/backuppc /etc/nginx/sites-enabled
```

Change fcgiwrap executing user in systemd fcgiwrap.service file to user: backuppc

## Alternative lighttpd configuration

```
/etc/lighttpd/lighttpd.conf

server.port          = 81
server.username     = "backuppc"
server.groupname    = "backuppc"
server.document-root = "/srv/http"
server.errorlog      = "/var/log/lighttpd/error.log"
dir-listing.activate = "enable"
index-file.names     = ( "index.html", "index.php", "index.cgi" )
mime.type.assign    = ( ".html" => "text/html", ".txt" => "text/plain", ".jpg" => "image/jpeg", ".p

server.modules = ("mod_alias", "mod_cgi", "mod_auth", "mod_access" )

alias.url           = ( "/BackupPC_Admin" => "/usr/share/backuppc/cgi-bin/BackupPC_Admin" )
alias.url           += ( "/backuppc" => "/usr/share/backuppc/html" )

cgi.assign          += ( ".cgi" => "/usr/bin/perl" )
cgi.assign          += ( "BackupPC_Admin" => "/usr/bin/perl" )

auth.backend        = "plain"
auth.backend.plain.userfile = "/etc/lighttpd/passwd"
auth.require        = ( "/BackupPC_Admin" => ( "method" => "basic", "realm" => "BackupPC", "require" => "user=

/etc/lighttpd/passwd

admin:yourpassword
```

And create log file:

```
# touch /var/log/lighttpd/error.log
# chown backuppc:backuppc /var/log/lighttpd/error.log
```

## Accessing the admin page

Before accesing de admin page you have to specify which users/groups will be able to edit BackupPC's configuration.

```
/etc/backuppc/config.pl

$Conf{CgiAdminUserGroup} = '<authorized groups>';
$Conf{CgiAdminUsers}     = '<authorized users>'; # <-- set to '*' if the webserver is not autenticating
```

Browse to [http://localhost/BackupPC\\_Admin](http://localhost/BackupPC_Admin) respectively [http://your\\_backuppc\\_server\\_ip/BackupPC\\_Admin](http://your_backuppc_server_ip/BackupPC_Admin).

## Website view problem

Due an Apache directive, the web interface may not shown properly. If that is your case, just modify the line in your `/etc/httpd/conf/httpd.conf` that avoids `.htaccess` and `.htpasswd` from viewed for clients or change directory name `/usr/share/backuppc/html` for `/usr/share/backuppc/files` and update `/etc/httpd/conf/extra/backuppc.conf` with the new path, as it follows:

```
-----  
/etc/httpd/conf/extra/backuppc.conf  
-----  
Alias          /BackupPC/images      /usr/share/BackupPC/files/  
-----
```

## See also

- BackupPC Home page (<http://backuppc.sourceforge.net/index.html>)
- BackupPC documentation (<http://backuppc.sourceforge.net/faq/BackupPC.html>)

Retrieved from "https://wiki.archlinux.org/index.php?title=BackupPC&oldid=467935"

Category: System recovery

- 
- This page was last modified on 8 February 2017, at 13:07.
  - Content is available under GNU Free Documentation License 1.3 or later unless otherwise noted.