

RUNNING PHP AS A SEPARATE PROCESS

Lukman HDP (lukman[at]id.go.bpk)

Pada server-server hosting yang memuat banyak virtual host, biasanya apache dikonfigurasi menggunakan *Name-Based Virtual Host*. Untuk membuat konfigurasi virtual host ini tidaklah sulit, termasuk mengintegrasikan dengan PHP. Saya juga sudah beberapa kali membuatnya untuk beberapa site monitoring.

Ada dua cara populer untuk menginstall PHP agar terintegrasi ke dalam Apache, yang pertama adalah membuat PHP secara static masuk ke dalam *binary* apache, dan yang kedua membuat PHP menjadi modul yang dipanggil oleh apache.

Kedua cara tersebut memiliki kesamaan dalam hal user yang menjalankan proses, skrip-skrip PHP akan berjalan sebagai user yang sama dengan user yang menjalankan Apache.

Sehingga, jika dalam satu server ada user A dan B, maka user A harus meng-*set permission* pada file-filenya minimal 755 agar file-file tersebut bisa dibaca oleh user Apache.

Tapi dengan demikian, tentu saja user B juga akan bisa mengintip file-file user A (karena *permission* yang 755 tadi) termasuk juga file config.php (atau file-file konfigurasi database yang lain).

Masalah lain yang timbul adalah, jika apache+php running sebagai satu user process (ambil contoh user www), kemudian dalam web kita ada fasilitas yang isinya meng-*create* sebuah file, maka meskipun program/skrip PHP tersebut milik user A, tapi file baru yang di-*create* tersebut adalah terdaftar sebagai milik dari user www, sehingga mungkin akan ada masalah dalam *editing* maupun *deleting* file tersebut.

Workaround yang mungkin adalah suexec. Apache sudah mengintegrasikan fasilitas suexec ini ke dalam program webservernya, walaupun secara default, fasilitas ini tidak diaktifkan. Suexec intinya adalah menjalankan program-program CGI dengan *privilege* user masing-masing yang memiliki file CGI tersebut. Termasuk pula PHP, jika ingin diperlakukan demikian, maka PHP harus di-*compile* sebagai program CGI, bukan sebagai modul apache.

Ada beragam tutorial di internet yang bisa dijadikan panduan untuk ini, tapi dari kemarin tidak ada satupun yang berhasil saya coba. Akhirnya saya menemukan suatu *interface* mediator yang bernama SuPHP.

Berikut langkah yang berhasil saya lakukan, ini berhasil pada kondisi saya, mungkin perlu sedikit penyesuaian jika lingkungan anda berbeda dengan lingkungan saya. Kondisi mesin saya adalah HP ProLiant 3x0 dengan sistem operasi Linux Fedora Core 5.

1. Menginstall [Apache](#)

Sesuai yang tertera pada halaman [FAQ SuPHP](#), bahwa versi Apache yang mereka support adalah Apache versi 2.0 dan 1.3. Sehingga dalam hal ini saya menggunakan apache versi httpd-2.0.59. Bisa didownload di <http://httpd.apache.org>.

```
[lukman@webserver src]$ wget http://apache.cbn.net.id/httpd/httpd-2.0.59.tar.bz2
[lukman@webserver src]$ tar -jxvf httpd-2.0.59.tar.bz2
[lukman@webserver src]$ cd httpd-2.0.59
[lukman@webserver httpd-2.0.59]$ ./configure \
--prefix=/usr/local/apache \
--enable-so \
--disable-userdir

[lukman@webserver httpd-2.0.59]$ make
[lukman@webserver httpd-2.0.59]$ su
Password:
[root@webserver httpd-2.0.59]# make install
```

Opsi penting yang tidak boleh ditinggalkan adalah `--enable-so` karena suPHP nanti akan menjadi modul bagi Apache. Jadi gambarannya, Apache akan memanggil modul yang bernama suPHP, modul suPHP akan memanggil PHP(CGI) untuk mengeksekusi script.

Selanjutnya buat user untuk menjalankan Apache:

```
[root@webserver ~]# groupadd apache -g 80
[root@webserver ~]# adduser apache -u 80 -g apache -d /usr/local/apache -M -s
/sbin/nologin
```

Edit file `/usr/local/apache/conf/httpd.conf` dan sesuaikan baris berikut untuk menentukan user yang menjalankan apache:

```
User apache
Group apache
```

2. Menginstall [PHP](#)

Sebagai catatan, instalasi PHP yang kita lakukan harus menghasilkan PHP versi CGI, bukan dynamic ataupun static module buat Apache. Saat ini saya masih prefer menggunakan PHP versi 4 daripada versi 5.

```
[lukman@webserver src]$ wget http://id2.php.net/distributions/php-4.4.3.tar.bz2
[lukman@webserver src]$ tar -jxvf php-4.4.3.tar.bz2
[lukman@webserver src]$ cd php-4.4.3
```

```
[lukman@webserver php-4.4.3]$ ./configure \  
--prefix=/usr/local/php \  
--disable-apxs \  
--disable-apache \  
--disable-cli \  
--enable-cgi \  
--with-mysql \  
--with-zlib \  
--with-bz2 \  
--with-gdbm \  
--with-db4 \  
--with-jpeg \  
--with-png \  
--with-freetype \  
--with-gd \  
--disable-ipv6 \  
--enable-magic-quotes
```

```
[lukman@webserver httpd-2.0.59]$ make
```

```
[lukman@webserver httpd-2.0.59]$ su
```

Password:

```
[root@webserver httpd-2.0.59]# make install
```

Opsi penting yang tidak boleh terlupakan adalah `--disable-apxs` dan `--enable-cgi` yang artinya men-disable fungsi modus terhadap Apache dan sebagai gantinya mengaktifkan fungsi modus CGI.

Untuk memeriksa apakah PHP yang sudah anda install adalah versi CGI, anda bisa mengeceknya menggunakan perintah berikut:

```
[lukman@webserver ~]$ /usr/local/php/bin/php -v  
PHP 4.4.3 (cgi) (built: Aug 14 2006 12:12:04)  
Copyright (c) 1997-2006 The PHP Group  
Zend Engine v1.3.0, Copyright (c) 1998-2004 Zend Technologies
```

Pastikan yang ter-*built* adalah `cgi` dan bukan `cli` atau yang lain

3. Menginstall SuPHP

Dari situs SuPHP: *suPHP is a tool for executing PHP scripts with the permissions of their owners. It consists of an Apache module (mod_suphp) and a setuid root binary (suphp) that is called by the Apache module to change the uid of the process executing the PHP interpreter.*

```
[lukman@webserver src]$ wget http://www.suphp.org/download/suphp-0.6.1.tar.gz  
[lukman@webserver src]$ tar -zxvf suphp-0.6.1.tar.gz  
[lukman@webserver src]$ cd suphp-0.6.1
```

```
[lukman@webserver suphp-0.6.1]$ ./configure \  
--prefix=/usr/local/suphp \  
--sysconfdir=/usr/local/suphp/etc \  
--with-apxs=/usr/local/apache/bin/apxs \  
--with-setid-mode=owner \  
--with-min-uid=500 \  
--with-min-gid=500 \  
--with-apache-user=apache \  
--with-logfile=/var/log/httpd/suphp_log \  
--with-php=/usr/local/php/bin/php
```

Keterangan atas beberapa opsi di atas adalah:

--sysconfdir=/usr/local/suphp/etc : file konfigurasi akan ditaruh di folder

/usr/local/suphp/etc

--with-apxs=/usr/local/apache/bin/apxs : lokasi file apxs milik apache

--with-setid-mode=owner : PHP akan dijalankan sesuai dengan user pemilik file tersebut

--with-min-uid=500 : userid minimal yang bisa digunakan (default minimal userid pada linux)

--with-min-gid=500 : userid minimal yang bisa digunakan (default minimal userid pada linux)

--with-apache-user=apache : user yang menjalankan apache

--with-logfile=/var/log/httpd/suphp_log : lokasi log file

--with-php=/usr/local/php/bin/php : lokasi file binary PHP

Program ini defaultnya diinstall di /usr tapi karena saya ingin semuanya teratur pada masing-masing folder (agar mudah dalam melakukan proses uninstall) maka saya menginstall ke dalam suatu folder menggunakan --prefix=/usr/local/suphp. Untuk menyesuaikan hal tersebut, kita harus mengedit file `src/apache2/mod_suphp.c`.

Sesuaikan path-nya karena kita menginstall tidak pada direktori standard (baris 106):

```
#define SUPHP_PATH_TO_SUPHP "/usr/local/suphp/sbin/suphp"
```

Baris 316 dan 317, cari entry yang berbunyi:

```
ACCESS_CONF
```

ganti menjadi

```
RSRC_CONF | ACCESS_CONF
```

Selanjutnya baru kita *compile* dan install

```
[lukman@webserver suphp-0.6.1]$ make
```

```
[lukman@webserver suphp-0.6.1]$ su
```

Password:

```
[root@webserver httpd-2.0.59]# make install
```

Selanjutnya copy contoh file konfigurasi dan edit seperlunya

```
[lukman@webserver suphp-0.6.1]$ cp doc/suphp.conf-example  
/usr/local/suphp/etc/suphp.conf
```

4. Konfigurasi Apache agar bisa menggunakan SuPHP

Edit file `/usr/local/apache/conf/httpd.conf` dan tambahkan *entry* seperti berikut:

```
LoadModule suphp_module modules/mod_suphp.so  
suPHP_Engine on  
suPHP_AddHandler x-httpd-php  
AddHandler x-httpd-php .php
```

5. Melakukan test instalasi

Jalankan daemon apache:

```
[root@webserver root]# /usr/local/apache/bin/apachectl start
```

Kemudian buat sebuah file yang `index.php` dan taruh di `/usr/local/apache/htdocs`, isi file tersebut adalah sebagai berikut:

```
<? echo `id`; ?>
```

Ubah kepemilikan file menjadi milik salah seorang user system yang valid:

```
[root@webserver root]# chown lukman:lukman /usr/local/apache/htdocs/index.php
```

Kemudian menggunakan browser, bukalah website anda. Jika DNS sudah dikonfigurasi, ketik `http://namaserver/` atau jika belum menggunakan DNS, anda bisa menggunakan nomor IP-nya `http://192.168.10.37/`

Tampilan pada browser anda seharusnya menghasilkan:

```
uid=500 (lukman) gid=500 (lukman) groups=500 (lukman)
```

Silahkan mencoba mengganti kepemilikan file, maka output dari `id` akan menghasilkan `uid` yang berbeda-beda.

Dengan demikian kita telah berhasil menginstall PHP yang *running* sebagai *user process* tersendiri.

6. Daftar Pustaka

1. <http://httpd.apache.org/docs/2.0/>
2. <http://www.php.net/docs.php>
3. <http://www.suphp.org/Documentation.en.html>

7. Thanks to

“Mak Kenthit”

8. Change Log

15 Agustus 2006