FastCGI

Amennyiben úgy szeretnék Shibboleth SP-t beüzemelni, hogy biztonsági, vagy bármilyen egyéb megfontolásokból az FastCGI-ként fusson, úgy az alábbiakban leírtak szerint kell(ene) a rendszernek működnie.

FastCGI-ről

CGI (Common Gateway Interface) alkalmazás esetén a program futása a következőképpen néz ki: jön a kérés a kiszolgáló felé, mire az továbbítja a kérést a CGI programnak. Jelen esetben ez a PHP értelmező lesz, ami futtatja a programot, és az eredményt visszaadja a kiszolgálónak, majd ezután eltűnik a memóriából. Egy következő kéréskor ugyanez a folyamat ismétlődik, amiből egyből láthatjuk, hogy ez a módszer nem túl hatékony, a PHP értelmezőt állandóan be kell tölteni, majd ki kell pakolni a memóriából.

A FastCGI ezt a be/ki töltögetést küszöböli ki: a PHP értelmező bent marad a memóriában addig, amíg szükség van rá. Sőt, egyszerre több értelmezőt is a memóriában tart, így egyszerre több kérést is ki tud szolgálni a szerver.

A CGI program külön szálként fut a kiszolgálón, elkülönítve tőle, és a többi hasonló száltól. Ha az egyik kedves felhasználó programja megöli a CGI értelmezőjét, akkor sincs semmi gond, az meghal, de a többi attól még vígan dolgozik, és szolgálja ki a lapokat.

Forrás: Weblabor

Lighttpd, mint webszerver

> **FIGYELEM!**
>
> Az alább leírt módszer csak elviekben működik, a gyakorlatban - a szükséges patchelés ellenére sem - sikerült működésre bírni. A hiba oka egy **lighttpd bug**, melyről a fejlesztők is tudnak, ám javítása nem szerepel a napirenden, mivel hivatalosan a patch megjavítja, ám a gyakorlat ezt cáfolja. Az érthetőség kedvéért hibajelenség pontos leírása a telepítési folyamat leírása után található.

Lighttpd

A figyelmeztetésben is emlegetett, és az alább részletezett patch-elés miatt nem a legfrissebb, hanem az 1.4.15-ös lighttpd-t használjuk.

Beszerzés

```
wget http://www.lighttpd.net/download/lighttpd-1.4.15.tar.gz
tar -xf http://www.lighttpd.net/download/lighttpd-1.4.15.tar.gz
```

A szükséges patch letöltése

```
wget http://redmine.lighttpd.net/attachments/download/91/fastcgi-authorizer-fixes.diff
```

Patchelés (a letöltött lighthttpd kitömörített könyvtárában vagyunk):

```
patch -p0 < fastcgi-authorizer-fixes.diff
```

Majd a patchelés kiegészítéseként be kell szúrnunk egy sort a `src/mod_fastcgi.c` fájlba.

```
                            con->http_status == 0)) {
                               /*
                                * If we are here in AUTHORIZER mode then a request for
autorizer
                                * was proceeded already, and status 200 has been returned. We
need
                                * now to handle autorized request.
                                */
+++                             con->http_status = 0;
                               if ( ! buffer_is_empty(host->docroot) ) {
                                       /*
                                        * Serve local file if they specified
                                        * a docroot
                                        */
```

Fordítás és telepítés (`configure` paraméterei természetesen tovább finomíthatók)

```
./configure --libdir=/usr/lib/lighttpd --with-openssl --with-openssl-libs=/usr/lib
make
make install
```

Telepítés utáni beállítás:

```
vim /etc/lighttpd/lighttpd.conf
```

server.modules szekcióba be kell írni

```
"mod_fastcgi",
```

majd ugyanebben a fájlban meg kell adni az alapvető beállításokat mind a 80-as, mind a 443-as portra vonatkozólag

```
server.name     = "sp2.example.org"
server.document-root       = "/var/www/"

$SERVER["socket"] == "10.0.0.8:443" { // a host ip-címe
  ssl.engine = "enable"
  ssl.pemfile = "/etc/lighttpd/certs/lighttpd.pem" // ahol a domain-re vonatkozó
tanusítványunk található
}
```

Shibboleth

Beszerzés

```
dget http://ftp.de.debian.org/debian/pool/main/s/shibboleth-sp2/shibboleth-sp2_2.0.dfsg1-4.dsc
dpkg-source -x *.dsc
```

Fordítás előtti hangolás

```
vim debian/rules
```

Be kell szúrni a ./configure paramétereit meghatározó sorba:

```
--with-fastcgi=/usr/sbin/lighttpd (a lighttpd elérési útja)
```

Shibboleth + Lighttpd

Még két beállítást kell eszközölnünk, hogy a webszerver tudja, hogy shibbolethes hívások esetén mi a teendő. Először is a FastCGI beállításokat betöltetjük a lighttpd-vel (ez shibboleth-től függetlenül is kellhet, pl. php futtatása esetén). Egy egyszerű szimbolikus linket készítünk a mods-available könyvtár vonatkozó fájljáról a mods-enabled -be. A második, hogy ezt a fájlt kiegészítjük két, shibboleth specifikus sorral.

```
"/Shibboleth.sso" => ((
      "socket" => "/tmp/fcgi-resp.sock",
      "bin-path" => "/usr/lib/shibboleth/shibresponder",
      "check-local" => "disable",
      "mode" => "responder"
)),
"/"   => ((
      "socket" => "/tmp/fcgi-auth.sock",
      "bin-path" => "/usr/lib/shibboleth/shibauthorizer",
      "check-local" => "disable",
      "mode" => "authorizer"
 )),
```

Ezek után - s az itt nem részletezett Shibboleth SP beállítás után - a rendszernek további beállítások nélkül működnie kellene. Ám **kizárólag statikus tartalmat tud rendeltetésszerűen kiszolgálni, dinamikusat nem**.

A HIBAJELENSÉG

Amennyiben használni akarunk "authorizer" modult (a shibbolethnél ez ugye kell, a másik modul, a "responder" típusú mellett), akkor ehhez a fastcgi beállításoknál meg kell adni egy gyökér pontot, amelyhez viszonyítva az összes vele egy szinten lévő, ill. alatta lévő hivatkozás esetén lefut az authorizer, és megmondja, hogy jogosult, tehát mehet tovább, vagy sem. Ez a gyökérpont a shibbolethes beállításoknál a "/" kell, hogy legyen, tehát minden esetben átfut az authorizer-en egy-egy kérés. Ez nem is lenne rossz, de a bug miatt, amennyiben a fastcgi beállításoknál az authorizer megadása után bármilyen (pl. php) respondert beállítunk, akkor az az authorizer által lefedett környezetben (esetünkben a "/" miatt mindenhol) a webszerver 403 hibát dob.

Változat #2
cziernorbert hozta létre 2026-04-14 13:22:32 CEST
cziernorbert frissítette 2026-04-21 13:37:13 CEST