

ArchiveOverHTTP

Az archiválás célja az, hogy olyan adatokat mentünk le, amelyre valószínűleg sohasem lesz szükség, de ha mégis, akkor nagy biztonsággal helyreállítható legyen.

Ez a leírás azt feltételezi, hogy kevés file-t kell szalagra írni. Amennyiben sok állományról van szó, akkor az alábbi lehetőségek közül választhatunk:

- `tar` (zip, rar stb.) segítségével egy file-t csinálunk belőle
 - Ha kevés a helyünk lokálisan, akkor használjuk a `tar --remove-files` opcióját arra, hogy az archiválandó adatokat "belemozgassuk" az archívumba, pl így:
 - `tar -c --remove-files -f cgp.tar cgp`
- kihasználjuk a rekurzív archiválás lehetőségét (DAV vagy SFTP használatával)

Kiszolgálás Apache webserverral

Az Apache nem hatékony nagy statikus állományok kiszolgálására, mivel ehhez nagyon sok memóriát foglal(hat). Erre megoldás a [mod_xsendfile](#) használata, amellyel egy egyszerű script segítségével hatékonyan szolgálhatjuk ki a file-okat, és tetszés szerint autorizálhatunk. Az autorizáció különösen fontos akkor, ha érzékeny adatokat akarunk menteni, hiszen nem szeretnénk, ha egy publikus webszerverről bárki le tudná azokat tölteni.

Webszerver konfiguráció

Tetszőleges működő http(s) webservert konfigurálunk. Az alábbi két sort szúrjuk be:

```
XSendFilePath /path/to/archive/  
XSendFile on
```

Fontos, hogy az `XSendFilePath` a webservert *DocumentRoot*-ján kívül legyen, ellenkező esetben xsendfile-on kívüli módon is elérhetőek lehetnek az anyagok.

Kiszolgáló PHP script

```
<?php
```

```
# Place the IP addresses in this array that are authorized to download the
```

```
# archives
$trusted_networks = array(
    "193.225.50.212",
    #TODO IPv6!
);

# You should not allow directory listing, and you probably want to be explicit
# about what you serve through this script
$served_files = array(
    "proba.txt",
    "cgp-metadata.txt",
    "cgp.tar",
    "szfe-orphaned_20130318.tar",
    "maildirs2-orphaned_20130318.tar",
);

# The directory where the archives reside
$dir = "/srv/maildirs/archive";

$ip = $_SERVER['REMOTE_ADDR'];
$file = $_GET['file'];

if (!$file) {
    echo "Missing 'file' parameter";
    exit(1);
}
if (!preg_match('/^[a-z0-9A-Z\-\_]+\.\?[a-z0-9A-Z\-\_]*$/',$file)) {
    echo "Invalid 'file' parameter";
    exit(1);
}

if (in_array ($ip, $trusted_networks)) {
    if (in_array ($file, $served_files)) {
        header("Content-Type: application/octet-stream");
        header("Content-Disposition: attachment; filename=\"\$file\"");
        header("X-Sendfile: $dir/$file");
        exit(0);
    } else {
        echo "Sorry, this file can not be served!";
        exit(1);
    }
}
```

```
    }  
  } else {  
    echo "Your IP ($ip) is not authorized!\n";  
    exit(1);  
  }  
  
exit(0);
```

A scriptet nevezzük el mondjuk `archiver-get.php`-nek. A programhoz való hozzáférést az Apache további eszközeivel is szűkíthetjük (Basic Auth, IP korlátozás stb), de ezek már redundáns védelmet nyújtanak.

Tiszta URL-ek

Ugyan az archiver támogatja a query stringek (`?file=foobar.txt`) használatát, azonban ez kényelmetlen, mivel át kell nevezni a végeredményt ahhoz, hogy értelmes nevet kapjunk. Ezért mindenképpen érdemes egy egyszerű `mod_rewrite` trükköt használni:

```
<IfModule mod_rewrite.c>  
  RewriteEngine on  
  RewriteRule ^(.*)$ /archiver-get.php?file=$1 [L]  
</IfModule>
```

A `.htaccess` file-t én az archiver könyvtárban helyeztem el, ezután az URL így nézett ki:

- <https://szerverneve.niif.hu/archiver/proba.txt>

Debian megjegyzés

Nagy file-ok kiszolgálására a `mod_xsendfile`-nak legalább a 0.11-es verzióját kell használnunk. Ez (a szöveg írása idején) nincs a Debian Squeeze-ben, azonban minden további nélkül használhatjuk itt is a wheezy-s bináris deb csomagot.

Változat #1

document-uploader hozta létre 2025-08-07 12:11:57 CEST

document-uploader frissítette 2025-08-07 12:11:57 CEST