



Piscina PHP

Ziua 04

Staff 42 piscine@42.fr

Sumar:

Acest document este subiectul zilei 04 a piscinei PHP din cadrul Academy+Plus.

Cuprins

I	Instructiuni	2
II	Preambul	3
III	Exercitiu ex00: vhost	4
IV	Exercitiu ex01: session	5
V	Exercitiu ex02: create_account	8
VI	Exercitiu ex03: modif_account	10
VII	Exercitiu ex04: auth	12
VIII	Exercitiu ex05: 42chat	14

Capitolul I

Instructiuni

- Folositi doar aceste pagini ca referinta: nu plecati urechea la zgomotul de pe culoar.
- Subiectul se poate modifica pana cu cel mult o ora inainte de intalnire.
- Doar rezultatul muncii voastre, livrat in directorul vostru de lucru/livrare, va fi luata in considerare la corectare.
- Ca si in timpul piscinei C, exercitiile voastre vor fi corectate de catre colegii de piscina SI/SAU de catre Moulinette.
- Moulinette este foarte stricta la corectare. Ea este total automatizata. E imposibil sa discutati pe seama notarii facute de aceasta. Fiti foarte rigurosi pentru a evita surprizele.
- Utilizarea unei functii interzise se considera ca fiind o tentativa de trisare. Orice tentativa de acest fel va fi sanctionata cu nota -42.
- Exerciitiile sunt foarte precis ordonate, de la cele simple la cele complexe. In niciun caz nu va fi luat in considerare un exercitiu complex rezolvat inaintea unui simplu nerezolvat complet si bine.
- Nu trebuie sa lasati in directorul vostru de lucru niciun alt fisier, decat cele explicit mentionate de enuntul exercitiului.
- Aveti intrebari? Intrebati-i pe vecinii vostri din dreapta sau din stanga.
- Manualul vostru de referinta se numeste **Google**.
- Puteti discuta pe forum. Solutia la problema voastra este deja probabil, gasita. In caz contrar, va revine sarcina sa demarati investigarea.
- Cititi cu atentie exemplele. Ele pot avea solutii ce nu sunt precizate altfel prin enuntul subiectului ...
- Reflectati. Cu putin noroc, dupa Thor, dupa Alex sau Ramona!


Capitolul II

Preambul

Pana acum ar fi trebuit sa faceti subiectele zilelor precedente ale piscinei si sa vizionati videoclipul zilei 04; in plus parcurgeti chesiunile de securitate ai algoritmului hash.

Capitolul III

Exercitiu ex00: vhost


	Exercitiu: 00
vhost	
Director de lucru: <i>ex00/</i>	
Fisier(e) de iesire: <code>httpd-app.conf</code> , <code>httpd-vhosts.conf</code> si <code>bitnami-apps-vhosts.conf</code>	
Functii autorizate: <code>niciunul</code>	
Observatii: <code>n/a</code>	

Azi se continua cu PHP pe server-ul web.

- configurati un vhost “j04.local.42.fr” in `~/mamp/apps/j04/`
- copiatii fisierele `httpd-app.conf`, `httpd-vhosts.conf` si `bitnami-apps-vhosts.conf` in directorul vostru de lucru/livrare `ex00/`

Capitolul IV

Exercitiu ex01: session

	Exercitiu: 01
session	
Director de lucru: <i>ex01/</i>	
Fisier(e) de iesire: <i>index.php</i>	
Functii autorizate: <i>echo, session_start(), \$_GET, \$_SESSION</i>	
Observatii: <i>n/a</i>	

Creati o pagina `index.php` ce contine un formular permitand crearea/modificarea identicatorului (login-ul) si parola.

- formularul se va numi “`index.php`” si va utiliza metoda “GET”
- campul identicator se va numi “login”
- campul parola se va numi “passwd”
- boutonul submit se va numi “submit” si va avea valoarea “OK” (daca receptionati aceasta valoare in campul submit nu modificati valorile salvate pe sesiune)
- toate tag-urile formularului trebuie fie fiecare singure pe o linie (vezi exemplul)
- campurile completate anterior vor trebui recomplete.

Exemplu de utilizare: se incepe prin testarea daca s-a receptionat bine cookie-ul de sesiune la primul acces (lipsesc in mod deliberat liniile din headers pentru claritatea HTML-ului din formulare pentru a va lasa sa lucrati in liniste):

```
$> curl -v -c cook.txt 'http://j04.local.42.fr:8080/ex01/index.php'
> GET /ex01/index.php HTTP/1.1
>
< HTTP/1.1 200 OK
* Added cookie PHPSESSID="mfpmngdi4qjbfli03nfgrevu17" for domain j04.local.42.fr, path /, expire 0
< Set-Cookie: PHPSESSID=mfpmngdi4qjbfli03nfgrevu17; path=/
<
<html><body>
<form ...>
  Identifiant: <input ... name="login" value="" />
  <br />
  Mot de passe: <... />
  <input type="submit" ... />
</form>
</body></html>
$>
```

Apoi se trimite formularul si se observa rezultatul

```
$> curl -v -b cook.txt 'http://j04.local.42.fr:8080/ex01/index.php?login=sb&passwd=beeone&submit=OK'
> GET /ex01/index.php?login=sb&passwd=beeone&submit=OK HTTP/1.1
> Cookie: PHPSESSID=mfpmngdi4qjbfli03nfgrevu17
>
< HTTP/1.1 200 OK
<
<html><body>
<form ...>
  Identifiant: <input ... name="login" value="sb" />
  <br />
  Mot de passe: <... value="beeone" />
  <input type="submit" ... />
</form>
</body></html>
$>
```

Apoi se reincarca pagina fara a se transmite valorile in url pentru a verifica ca ele sunt totusi prezente


```
$> curl -v -b cook.txt 'http://j04.local.42.fr:8080/ex01/index.php'
> GET /ex01/index.php HTTP/1.1
> Cookie: PHPSESSID=mfpmgdi4qjbfl03nfgrevu17
>
< HTTP/1.1 200 OK
<
<html><body>
<form ...>
  Identifiant: <input ... name="login" value="sb" />
  <br />
  Mot de passe: <... value="beeone" />
  <input type="submit" ... />
</form>
</body></html>
$>
```

Ultima etapa e indepartarea cookie-urilor request-ului si se constata ca formularul este din nou gol si ca PHP-ul trimite un nou cookie de sesiune.

```
$> curl -v 'http://j04.local.42.fr:8080/ex01/index.php'
> GET /ex01/index.php?login=sb&passwd=beeone&submit=OK HTTP/1.1
>
< HTTP/1.1 200 OK
< Set-Cookie: PHPSESSID=r7u3bnggoe91negv7aqnjk0q6; path=/
<
<html><body>
<form ...>
  Identifiant: <input ... name="login" value="" />
  <br />
  Mot de passe: <... value="" />
  <input type="submit" ... />
</form>
</body></html>
$>
```


Capitolul V

Exercitiu ex02: create_account

	Exercitiu: 02
create_account	
Director de lucru: ex02/	
Fisier(e) de iesire: index.html, create.php	
Functii autorizate: echo, hash(), file_get_contents(), file_put_contents(), serialize(), unserialize(), \$_POST, file_exists(), mkdir()	
Observatii: n/a	

Creati o pagina index.html ce contine un formular ce permite crearea contului cu ajutorul unui identificator (login) si o parola. Un cont valid este creat pe baza unui identificator si parola nenule (fara campuri goale), in caz contrar returnand eroarea “ERROR\n”. In caz de succes transmiteti mesajul “OK\n”.

- formularul se va numi “create.php” si va folosi metoda “POST”
- campul identificator se va numi “login”
- campul parolei se va numi “passwd”
- butonul submit se va numi “submit” si va avea valoarea “OK” (daca nu primiti valoarea aceasta in campul submit, nu creati contul si transmiteti mesajul de eroare “ERROR\n”)
- tag-urile trebuie sa fie fiecare din ele singure pe o linie (vedeti exemplul)
- trebuie sa stocati contul astfel creat in directorul /private/passwd. Acest fisier trebuie sa fie un tablou serializat; vezi exemplul.
- fiecare cont trebuie sa fie un element al tabloului si trebuie sa fie un tablou el insusi cu campurile “login” ce contine identificatorul si “passwd” ce contine parola criptata (hash).


- nu trebuie sa stocati parola in “clar” ci “hash” (vezi videoclipul zilei)
- alegeti-va bine algoritmul de criptare
- daca identificatorul (login-ul) transmis exista deja in in tablou va trebui sa returnati un mesaj de eroare “ERROR\n”

Exemplu:

```
$> rm ~/mamp/apps/j04/htdocs/private/passwd
$> curl -d login=toto1 -d passwd=titi1 -d submit=OK 'http://j04.local.42.fr:8080/ex02/create.php'
OK
$> more ~/mamp/apps/j04/htdocs/private/passwd
a:1:{i:0;a:2:{s:5:"login";s:5:"toto1";s:6:"passwd";s:128:"2bdd45b3c828273786937ac1b4ca7908a431019
e8b93c9fd337317f92fac80dace29802bedc33d9259c8b55d1572cb8a6c1df8579cdaa02256099ed52a905d38";}}
$> curl -d login=toto1 -d passwd=titi1 -d submit=OK 'http://j04.local.42.fr:8080/ex02/create.php'
ERROR
$> curl -d login=toto2 -d passwd= -d submit=OK 'http://j04.local.42.fr:8080/ex02/create.php'
ERROR
$>
```

Capitolul VI

Exercitiu ex03: `modif_account`

	Exercitiu: 03
<code>modif_account</code>	
Director de lucru: <code>ex03/</code>	
Fisier(e) de iesire: <code>index.html</code> , <code>modif.php</code>	
Functii autorizate: <code>echo</code> , <code>hash()</code> , <code>file_get_contents()</code> , <code>file_put_contents()</code> , <code>serialize()</code> , <code>unserialize()</code> , <code>\$_POST</code>	
Observatii: n/a	

Creati o pagina `index.html` ce contine un formular ce permite modificarea parolei asociata contului. Utilizatorul va trebui sa-si transmita identificatorul, parola actuala (ce va fi denumita “ancien (veche)” pentru a evita orice ambiguitate) si noua sa parola. Parola nu se va modifica decat daca login-ul corespunde cu un cont deja existent si parola veche cu cea stocata in fisier, la momentul modificarii. Daca identificatorul nu exista, daca parola veche este gresita, sau daca parola noua nu este valida, `modif.php` va returna “ERROR\n” fara alte detalii. In caz de reusita se va returna “OK\n”

- formularul se va numi “`modif.php`” si va folosi metoda “POST”
- campul “Identifiant (identificater)” se va numi “login”
- campul “Ancien mot de passe (parola veche)” va fi “oldpw”
- campul “Nouveau mot de passe (parola noua)” va fi “newpw”
- boutonul submit se va numi “submit” si va avea valoarea “OK” (daca nu receptionati aceasta valoare in campul submit, nu modificati contul si returnati eroarea “ERROR\n”)
- toate tag-urile formularului vor trebui sa fie fiecare singure pe un rand (vezi exemplul)


- trebuie sa folositi si sa modificati fisierul /private/passwd ce contine conturile create in ex02
- nu trebuie sa stocati parola in “clar” ci “hash” (vezi videoclipul zilei)
- alegeti un algoritm de criptare bun (vezi videoclipul zilei)

Exemple:

```
$> rm ~/mamp/apps/j04/htdocs/private/passwd
$> curl -d login=x -d passwd=21 -d submit=OK 'http://j04.local.42.fr:8080/ex02/create.php'
OK
$> curl -d login=x -d oldpw=21 -d newpw=42 -d submit=OK 'http://j04.local.42.fr:8080/ex03/modif.php'
OK
$> more ~/mamp/apps/j04/htdocs/private/passwd
a:1:{i:0;a:2:{s:5:"login";s:1:"x";s:6:"passwd";s:128:"fee32be7c00e73eab97a39549d79af73aec87b6fa22a0b
56867a4975fe82344cd9776c6d6dff419e0f2e415c492340bb8329bbfac0c872934df66466c2e0e5d3";}}
$> curl -d login=x -d oldpw=21 -d newpw=42 -d submit=OK 'http://j04.local.42.fr:8080/ex03/modif.php'
ERROR
$> curl -d login=x -d oldpw=42 -d newpw= -d submit=OK 'http://j04.local.42.fr:8080/ex03/modif.php'
ERROR
$>
```

Capitolul VII

Exercitiu ex04: auth

	Exercitiu: 04
auth	
Director de lucru: <i>ex04/</i>	
Fisier(e) de iesire: <code>auth.php</code> , <code>login.php</code> , <code>logout.php</code> , <code>whoami.php</code>	
Functii autorizate: <code>echo</code> , <code>hash()</code> , <code>file_get_contents()</code> , <code>file_put_contents()</code> , <code>serialize()</code> , <code>unserialize()</code> , <code>session_start()</code> , <code>\$_GET</code> , <code>\$_SESSION</code>	
Observatii: n/a	

Creati un fisier `auth.php` ce contine functia urmatoare:

```
function auth($login, $passwd);
```

Aceasta functie trebuie sa returneze TRUE daca contul `login/passwd` corepunde contului din fisierul `/private/passwd` si FALSE in caz contrar. Variabila `$passwd` contine parola in clar, deci INAINTE de criptare.

Creati de asemenea un fisier login.php ce ia ca parametru in url o variabila login si o variabila passwd. Aceasta pagina initiaza o sesiune, verifica valabilitatea perechii login/passwd si stocheaza pe sesiune o variabila “logged_on_user” ce contine:

- fie login-ul utilizatorului ce a transmis o pereche login/passwd corecta
- fie un sir vid in caz contrar

Daca perechea login/passwd este corecta pagina trebuie sa afiseze “OK\n”, iar in caz contrar “ERROR\n”. Acest fisier trebuie sa utilizeze fisierul auth.php folosind o directiva include.

```
$> rm ~/mamp/apps/j04/htdocs/private/passwd
$> curl -d login=toto -d passwd=titi -d submit=OK 'http://j04.local.42.fr:8080/ex02/create.php'
OK
$> curl 'http://j04.local.42.fr:8080/ex04/login.php?login=toto&passwd=titi'
OK
$>
```


Creati de asemenea un fisier logout.php ce nu ia niciun parametru dar care foloseste cookie-ul de sesiune pentru a-l sterge pe acesta. Aceasta pagina nu va afisa nimic.

Creati de asemenea un fisier whoami.php ce nu ia niciun parametru dar care foloseste cookie-ul de sesiune pentru afisarea login-ul continut in variabilele sesiunii “logged_on_user” urmat de “\n”. Daca aceasta variabila nu exista sau continutul este un sir vid, afisati doar “ERROR\n”

```
$> rm ~/mamp/apps/j04/htdocs/private/passwd
$> curl -d login=toto -d passwd=titi -d submit=OK 'http://j04.local.42.fr:8080/ex02/create.php'
OK
$> curl -c cook.txt 'http://j04.local.42.fr:8080/ex04/login.php?login=toto&passwd=titi'
OK
$> curl -b cook.txt 'http://j04.local.42.fr:8080/ex04/whoami.php'
toto
$> curl -b cook.txt 'http://j04.local.42.fr:8080/ex04/logout.php'
$> curl -b cook.txt 'http://j04.local.42.fr:8080/ex04/whoami.php'
ERROR
$>
```

Capitolul VIII

Exercitiu ex05: 42chat

	Exercitiu: 05
42chat	
Director de lucru: <i>ex05/</i>	
Fisier(e) de iesire: <code>index.html</code> , <code>create.html</code> , <code>modif.html</code> , <code>auth.php</code> , <code>create.php</code> , <code>modif.php</code> , <code>login.php</code> , <code>logout.php</code> , <code>speak.php</code> , <code>chat.php</code>	
Functii autorizate: <code>echo</code> , <code>session_start()</code> , <code>header()</code> , <code>hash()</code> , <code>file_get_contents()</code> , <code>file_put_contents()</code> , <code>serialize()</code> , <code>unserialize()</code> , <code>\$_GET</code> , <code>\$_POST</code> , <code>\$_REQUEST</code> , <code>fopen()</code> , <code>flock()</code> , <code>fclose()</code> , <code>time()</code> , <code>date()</code> , <code>date_default_timezone_set()</code> , <code>file_exists()</code> , <code>mkdir()</code>	
Observatii: n/a	

Scopul acestui exercitiu este de a crea un chat multi-utilizator. Trebuie sa incepeti cu reluarea fisierelor din exercitiilor anterioare pentru a realiza aplicatia urmatoare:

- pagina Home `index.html` propune un formular pentru conectarea la chat (formularul poate folosi fisierul `login.php`, cu metoda POST; modificati deci `login.php`), dar si un link pentru crearea contului (`create.html`) si pentru modificarea parolei (`modif.html`).
- `create.html` este formularul de criptare ex02. El apeleaza `create.php` conform acelorasi reguli.
- `modif.html` este formularul de criptare ex03. El apeleaza `modif.php` conform acelorasi reguli.
- `create.php` si `modif.php` trebuie sa afiseze “OK” in caz de succes SI sa redirecteze utilizatorul spre o pagina Home cu un header “Location:”. Utilizatorul va trebui apoi sa se autentifice.
- `login.php` in caz de succes trebuie sa afiseze un `iframe`, impartit orizontal in doua; partea de sus trebuie sa aiba 550px din pagina si sa contina “`chat.php`”, partea de

jos va trebui sa aiba 50px si sa contina "speak.php"

- speak.php permite adaugarea unui mesaj in chat. El trebuie sa verifice daca utilizatorul este corect log-at. Este un formular care se autoapeleaza cu un simplu camp msg transmis in metoda POST. Identitatea utilizatorului este triata de sesiune. speak.php trebuiesc completeze un tablou serializat dintr-un fisier /private/chat. Fiecare element al tabloului este de asemenea un tablou ce contine: "login", "time", "msg". Campul "login" este cel al utilizatorului ce a postat mesajul. Campul time este data/ora la care mesajul a fost trimis, in formatul. Campul msg este, in mod evident, mesajul.
- Atentie! Trebuie sa blocati fisierul cu flock pe timpul folosirii acestuia (citire / scriere) pentru a nu-l corupe in caz de scriere simultana a doi utilizatori, nici de citire partiala de catre un alt proces.
- chat.php permite afisarea continutului fisierului /private/chat formatandu-l. Vezi exemplul (atentie la timezone, <http://www.php.net/manual/en/timezones.php>)

Exemplu:

```
$> curl -d login=user1 -d passwd=pass1 -d submit=OK 'http://j04.local.42.fr:8080/ex05/create.php'
OK
$> curl -d login=user2 -d passwd=pass2 -d submit=OK 'http://j04.local.42.fr:8080/ex05/create.php'
OK
$> curl -c user1.txt -d login=user1 -d passwd=pass1 'http://j04.local.42.fr:8080/ex05/login.php'
...
<iframe name="chat" src="chat.php" width="100%" height="550px"></iframe>
<iframe name="speak" src="speak.php" width="100%" height="50px"></iframe>
...
$> curl -b user1.txt -d submit=OK -d msg=Bonjours 'http://j04.local.42.fr:8080/ex05/speak.php'
...
$> curl -b user1.txt -c user1.txt 'http://j04.local.42.fr:8080/ex05/logout.php'
$> curl -b user1.txt -d submit=OK -d msg=Bonjours 'http://j04.local.42.fr:8080/ex05/speak.php'
ERROR
$> curl -c user2.txt -d login=user2 -d passwd=pass2 'http://j04.local.42.fr:8080/ex05/login.php'
...
$> curl -b user2.txt -d submit=OK -d msg=Hello 'http://j04.local.42.fr:8080/ex05/speak.php'
...
$> more ~/mamp/apps/j04/htdocs/private/chat
a:2:{i:0;a:3:{s:5:"login";s:5:"user1";s:4:"time";i:1364287362;s:3:"msg";s:8:"Bonjours";}i:1;a:3:{s:5:
"login";s:5:"user2";s:4:"time";i:1364287423;s:3:"msg";s:5:"Hello";}}
$> curl -b user2.txt 'http://j04.local.42.fr:8080/ex05/chat.php'
[09:42] <b>user1</b>: Bonjours<br />
[09:43] <b>user2</b>: Hello<br />
$>
```

Bonus gratuit:

Odata ce ati realizat toate astea veti remarca ca iframe-ul chat.php nu se reincarca singur cand folositi speak.php. Iata un cod de javascript de adaugat in head-ul fisierului speak.php:


```
<script language="javascript">top.frames['chat'].location = 'chat.php';</script>
```

Atentie: actualizati fisierul /private/chat INAINTE de returnarea codului html a paginii, pentru a evita ca browser-ul sa reincarce chat.php prea devreme. Succes ;-)