42 PROJECT

Darkly

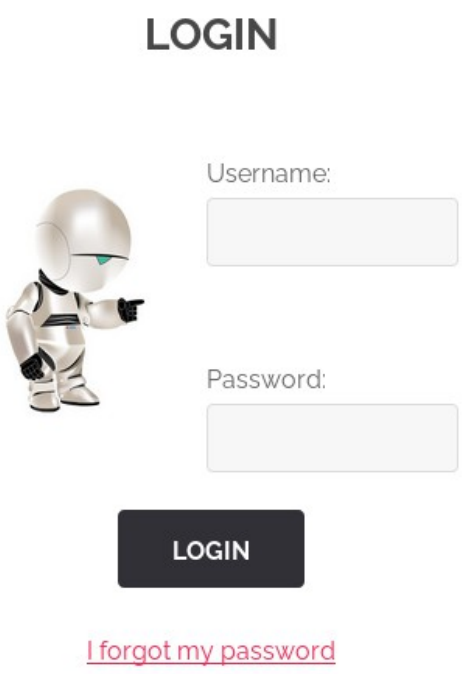
Darkly est un projet 42 visant à nous initier à la sécurité web. Une image ISO (tournant sur une machine virtuelle de type Linux – Oracle 32 bits) met à notre disposition un site web vulnérable, qu'il va falloir exploiter de 14 manières différentes.

ÉNUMÉRATION

Général :
> Site web en PHP version 5.3.0 (pas mal de CVE, mais ça ne semble pas être l'objet de l'exercice ici).
> "Spatial" by Templated (https://spatial.fleeksite.com/layout).

1. LOGIN PAGE

Le site web propose une page de login :



Lorsqu'on essaie des credentials aléatoires, un message "Sorry, Wrong Answer" nous est retourné. De même lorsqu'on laisse les champs entièrement vides.

> *Default credentials ?*
> *Bruteforce ?*
> *SQL Injection ?*
> *NoSQL Injection ? [rare]*
> *SQL Truncation ? [rare]*

2. PASSWORD RECOVERY PAGE

A partir de la page de login, on voit un lien "I forgot my password", qui mène à une page de password recovery, avec un simple bouton "Submit", et aucune autre option.

Lorsqu'on clique sur "Submit" sans rien modifier, le même message d'erreur "Sorry, Wrong Answer" nous est retourné.

> *Manipulation d'adresse ?*

3. SURVEY PAGE

Une page "survey" est disponible. Celle-ci indique différents noms, avec des notes, des nombres de voix :

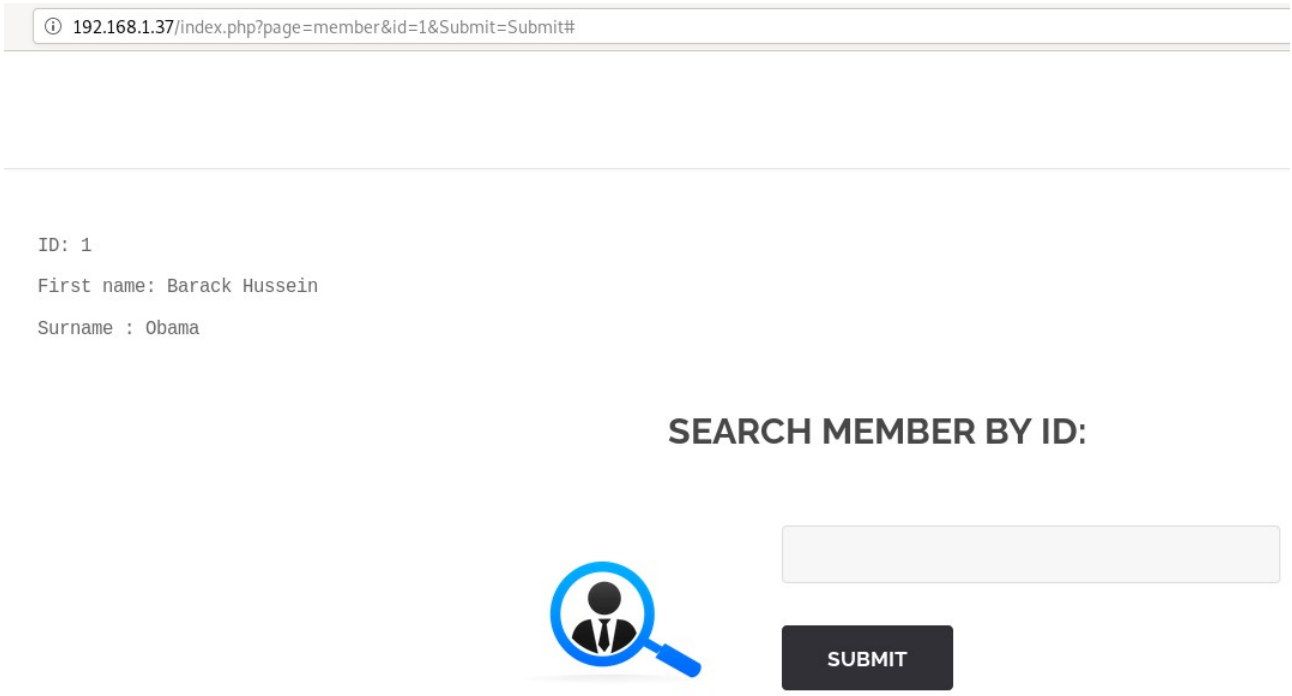
Make your choice

Grade	Average	Subject	Nb of vote(*indicative*)
1	4217.2	Laurie	4257
1	5.23529	Mathieu	17
1	8.64707	Thor	17

Il est possible de cliquer dans la première colonne pour indiquer un vote.

4. MEMBERS SEARCH PAGE

Une page nous permet de chercher des membres du site. Lorsqu'on entre un chiffre correspondant à l'ID d'un utilisateur existant, quelques informations nous sont affichées à propos de cet utilisateur.



Quelques ID's qui fonctionnent : 1, 2, 3, 5. On pourrait éventuellement FUZZ afin de trouver l'ensemble des IDs utilisateurs valides.

L'URL est intéressante lorsqu'on cherche un membre, puisqu'on a quelques paramètres qui nous permettent d'injecter de l'input :

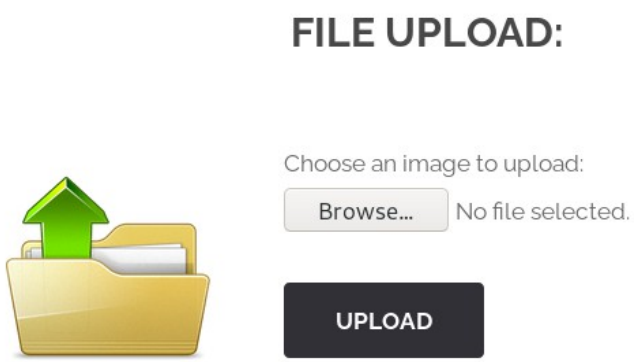
- **page** : à voir ci-dessous.
- **id** : l'id de l'utilisateur.
- **Submit** : un peu étrange comme paramètre, peut-être configure-t-il la méthode de soumission de l'input de recherche. Changer la valeur de ce paramètre ne semble pas changer le comportement de l'application.

> Fuzz members ?
> SQL Injection à partir du paramètre "id" ?

5. IMAGE UPLOAD PAGE

Une page nous permet d'upload des images. Cette page ne semble, à première vue, n'accepter que des images **jpg** ou **jpeg** (pas **png** ou autre).

```
/tmp/kermit.jpeg succesfully uploaded.
```

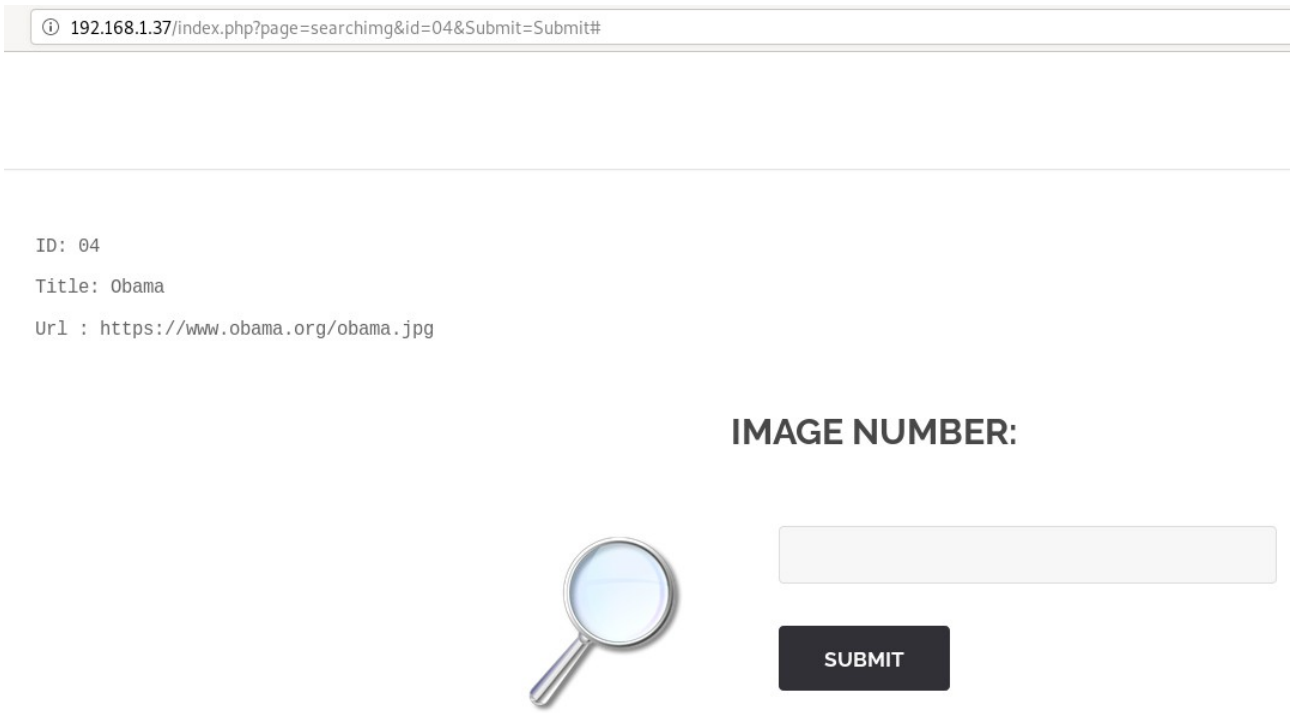


Suite à l'upload d'une image, le message ci-dessus est affiché. Il semble donc que nos images soient uploadés dans un premier temps dans le dossier `/tmp/` du serveur. On aurait donc besoin pour y accéder d'une faille LFI.

> *File upload vulnerabilities ?*

6. IMAGE SEARCH PAGE

Une page nous permet de rechercher des images à partir de leur ID :



Lorsque l'ID est valide, quelques informations sur une image nous sont révélées.

> *SQL Injection à partir du paramètre "id" ?*

7. IMAGES STORAGE

Sur la page d'accueil du site, deux images sont affichées sous le titre "Latest Uploaded Images". Leurs URLs sont les suivantes :

```
http://<IP>/images/04_Playstation.jpg
http://<IP>/images/05_Lounge.jpg
```

Le dossier `/images/` est probablement le dossier final dans lequel les images sont uploadées (avant, elles passent probablement par le dossier `/tmp` comme lorsqu'on a voulu uploader nos images nous-mêmes).

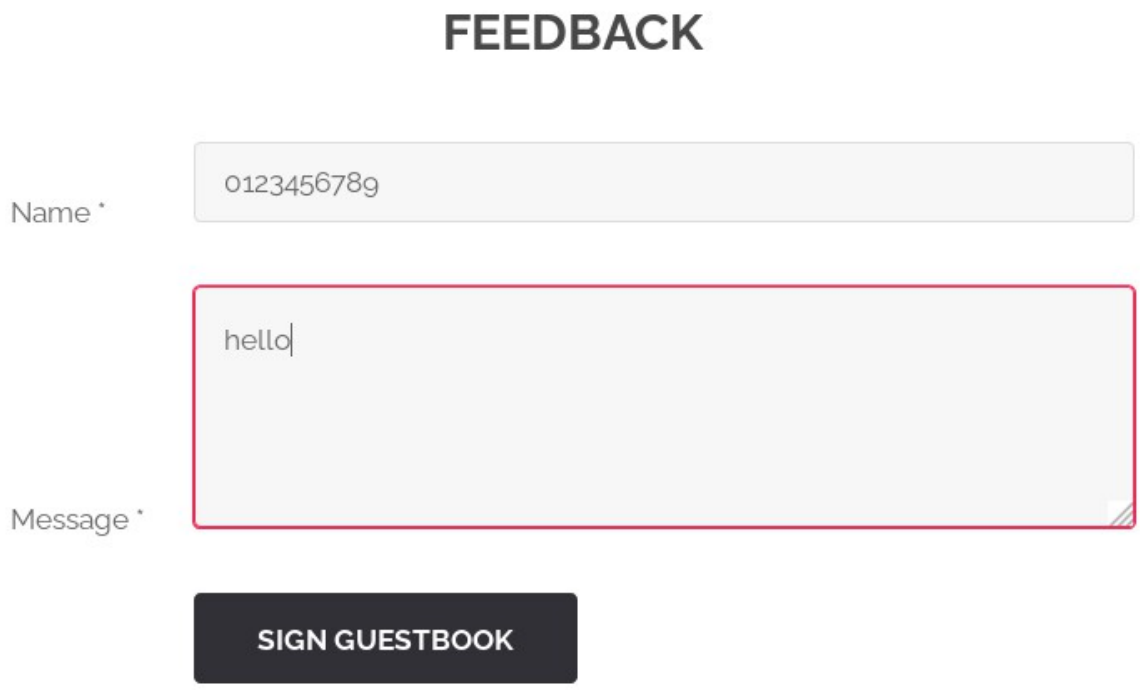
Ce dossier contient également les images de la page d'accueil (PRISM...), on le voit dans le code source de la page.

Lorsqu'on essaie d'accéder à ce dossier `/images/`, on a une erreur 403 Forbidden. La version du serveur utilisé par le backend est cependant leak, il s'agit de NGINX 1.8.0.

Les images dans le dossier `/images/` ne semblent pas correspondre avec les images résultant de la recherche d'images (voir 6).

8. FEEDBACK PAGE

Une page nous permet d'entrer un message, qui sera reflété sur la même page.



Le champ "name" est limité à 10 caractères. La limite du champ "Message" est bien plus haute. Des tests très basiques d'injection SQL ont montré que le champ "Message" n'était probablement pas vulnérable (les caractères spéciaux sont escaped). Le champ "Name" semble plus prometteur (une

balise <script> seule n'est pas affichée), mais limité aux 10 caractères, à moins qu'on puisse bypass cette limite en interceptant la requête.

> *XSS dans le champ "name" ?*

9. DIRECTORY FUZZING / INTERESTING LOCATIONS

Les sections ci-dessus regroupent des pages et fonctionnalités accessibles directement sur le site web. On effectue un directory fuzzing simple en complément.

>> Un scan des directories disponibles à la racine du site :

```
./ffuf -w /usr/share/dirb/wordlists/big.txt -u http://<IP>/FUZZ

admin           [Status: 301, Size: 184, Words: 6, Lines: 8]
audio           [Status: 301, Size: 184, Words: 6, Lines: 8]
css             [Status: 301, Size: 184, Words: 6, Lines: 8]
errors          [Status: 301, Size: 184, Words: 6, Lines: 8]
favicon.ico     [Status: 200, Size: 1406, Words: 2, Lines: 2]
fonts           [Status: 301, Size: 184, Words: 6, Lines: 8]
images          [Status: 301, Size: 184, Words: 6, Lines: 8]
includes        [Status: 301, Size: 184, Words: 6, Lines: 8]
js              [Status: 301, Size: 184, Words: 6, Lines: 8]
robots.txt      [Status: 200, Size: 53, Words: 4, Lines: 4]
whatever        [Status: 301, Size: 184, Words: 6, Lines: 8]
```

>> Un dirb simple, qui nous donne des résultats similaires, avec en plus un fuzzing des fichiers :

```
+ http://192.168.1.37/index.php (CODE:200|SIZE:6892)
+ http://192.168.1.37/favicon.ico (CODE:200|SIZE:1406)
+ http://192.168.1.37/robots.txt (CODE:200|SIZE:53)

==> DIRECTORY: http://192.168.1.37/admin/
==> DIRECTORY: http://192.168.1.37/audio/
==> DIRECTORY: http://192.168.1.37/css/
==> DIRECTORY: http://192.168.1.37/errors/
==> DIRECTORY: http://192.168.1.37/fonts/
==> DIRECTORY: http://192.168.1.37/images/
==> DIRECTORY: http://192.168.1.37/includes/
==> DIRECTORY: http://192.168.1.37/js/
==> DIRECTORY: http://192.168.1.37/whatever/

---- Entering directory: http://192.168.1.37/admin/ ----
+ http://192.168.1.37/admin/index.php (CODE:200|SIZE:1432)
==> DIRECTORY: http://192.168.1.37/admin/css/
==> DIRECTORY: http://192.168.1.37/admin/fonts/

---- Entering directory: http://192.168.1.37/css/ ----
==> DIRECTORY: http://192.168.1.37/css/images/

---- Entering directory: http://192.168.1.37/whatever/ ----
+ http://192.168.1.37/whatever/htpasswd (CODE:200|SIZE:38)
```

```
---- Entering directory: http://192.168.1.37/admin/css/ ----
==> DIRECTORY: http://192.168.1.37/admin/css/images/
```

> Le fichier robots.txt a le contenu suivant :

```
User-agent: *
Disallow: /whatever
Disallow: /.hidden
```

On connait déjà l'emplacement /whatever, ce fichier nous révèle cependant **/.hidden** (voir ci-dessous).

> Le path **/admin/** nous redirige sur une page de login différente que celle déjà évoquée (voir ci-dessous).

> Les paths `/css/` - `/errors/` - `/audio/` - `/images/` - `/fonts/` - `/includes/` - `/js/` nous renvoient des **403 Forbidden**.

> Le path `/whatever/` est disponible en directory listing, et ne contient que le fichier que le fuzzing nous a révélé, c'est-à-dire **htpasswd**. On récupère un nom d'utilisateur et un hash dans ce fichier :

```
root:8621ffdbc5698829397d97767ac13db3
```

> Cracker le hash de htpasswd ?
>Explorer le dossier .hidden ?

10. ADMIN PAGE

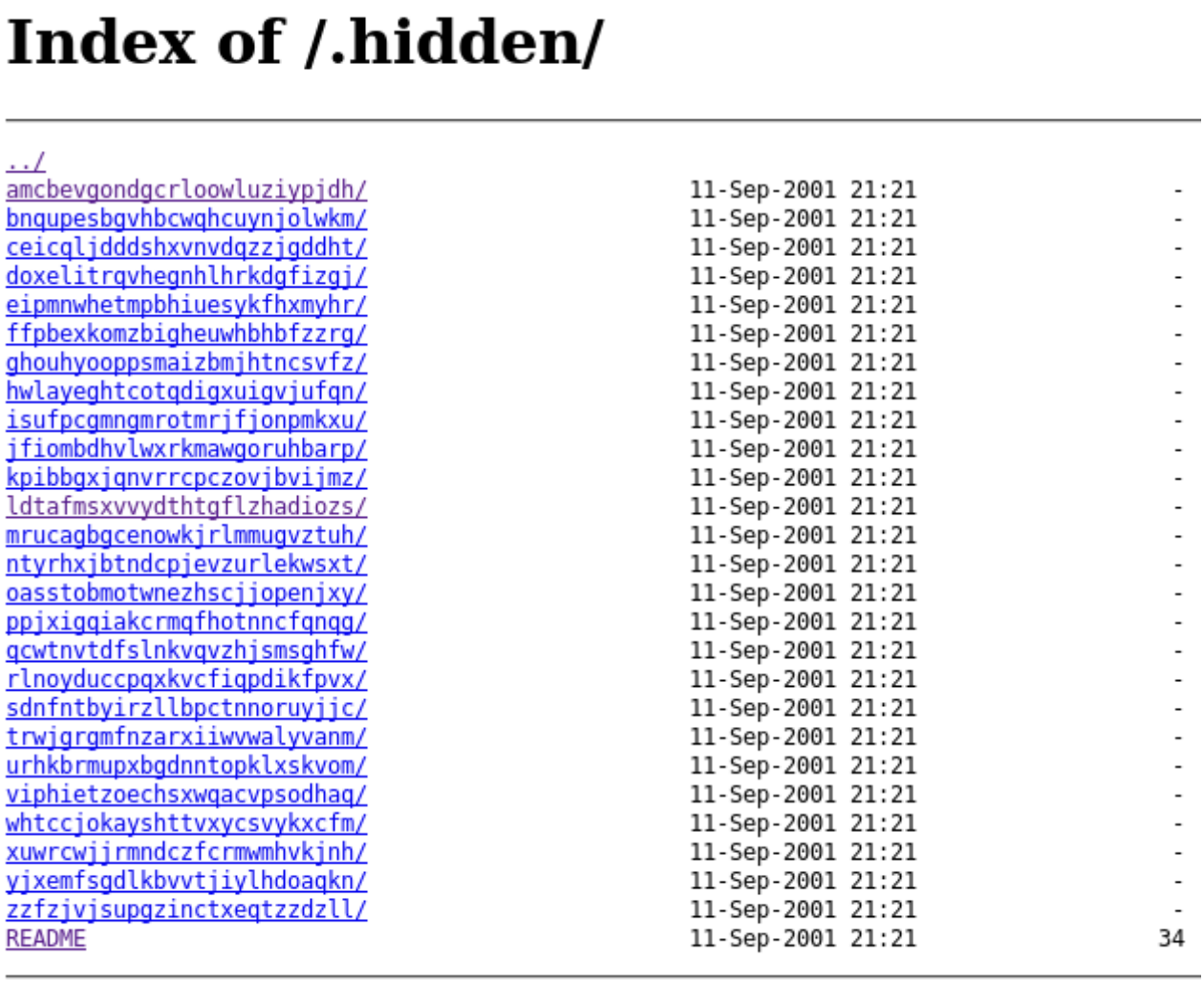
Le path `/admin/` nous redirige sur un espace de login pour une "zone sécurisée". Il s'agit d'un formulaire de login basique, avec username et password.

11. HIDDEN FOLDER

Le folder au path `http://<IP>/.hidden` contient un ensemble de dossiers aux noms générés apparemment aléatoirement, ainsi qu'un fichier README.

Lorsqu'on entre dans un de ces dossiers, on a de nouveau le même nombre de dossiers aléatoires. La profondeur de chaque dossier aléatoire composé lui-même de dossiers aléatoires est de 3. Lorsqu'on arrive au bout de la liste de dossiers, on a en général un fichier README avec un message humoristique.

Il s'agit je suppose ici de trouver le "bon" dossier qui contient un flag ou une information intéressante, en essayant de fuzz chaque dossier de sortie.



Index of /.hidden/

../		
amcbevgondgcrloowluziypjdh/	11-Sep-2001 21:21	-
bnqupesbgvhbcwqhcuynjolwkm/	11-Sep-2001 21:21	-
ceicqljdddshxvnvdqzzjgddht/	11-Sep-2001 21:21	-
doxelitrqvhegnhlhrkdgfizgj/	11-Sep-2001 21:21	-
eipmnwhetmpbhiuesykfhxmyhr/	11-Sep-2001 21:21	-
ffpbexkomzbigheuwhbhbfzzrg/	11-Sep-2001 21:21	-
ghouhyooppsmaizbmjhtncsvfz/	11-Sep-2001 21:21	-
hwlayeghtcotqdigxuigvjufqn/	11-Sep-2001 21:21	-
isufpcgmngmrotmrjfjonpmkxu/	11-Sep-2001 21:21	-
jfiombdhvlwxrkmawgoruhbarp/	11-Sep-2001 21:21	-
kpibbgxjqnvrrcpczovjbvijmz/	11-Sep-2001 21:21	-
ldtafmsxvvydthtgflzhadiozs/	11-Sep-2001 21:21	-
mrucagbgcenowkjrlmmugvztuh/	11-Sep-2001 21:21	-
ntyrhxjbtndcpjevzurlekwsxt/	11-Sep-2001 21:21	-
oasstobmotwnezhscjjopenjxy/	11-Sep-2001 21:21	-
ppjxigqiakcrmqfhotnncfqnqg/	11-Sep-2001 21:21	-
qcwtnvtdfslnkvqvzhjsmsghfw/	11-Sep-2001 21:21	-
rlnoyduccpqxkvcfiqpdikfpvx/	11-Sep-2001 21:21	-
sdnfntbyirzllbpctnnoruyjjc/	11-Sep-2001 21:21	-
trwjgrgmfnzarxiiwvwalyvanm/	11-Sep-2001 21:21	-
urhkbrmupxbgdnntopklxskvom/	11-Sep-2001 21:21	-
viphietzoechsxwqacvpsodhaq/	11-Sep-2001 21:21	-
whtccjokayshttvxycsvykxcfm/	11-Sep-2001 21:21	-
xuwrcwjjrmndczfcrmwmhvkjnh/	11-Sep-2001 21:21	-
yjxemfsgdlkbvvtjiylhdoaqkn/	11-Sep-2001 21:21	-
zzfzjvjsupgzinctxeqtzzdzll/	11-Sep-2001 21:21	-
README	11-Sep-2001 21:21	34

12. PAGE HANDLING

Lorsqu'on se balade sur le site, on s'aperçoit que les pages sont affichées avec des URLs formées de cette façon :

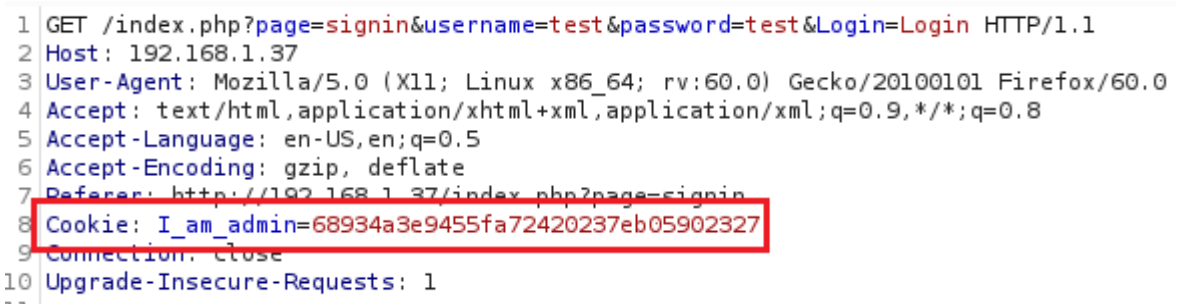
```
http://192.168.1.37/index.php?page=signin
http://192.168.1.37/index.php?page=survey
```
etc...

Ce qui pourrait laisser penser à une potentielle faille LFI, si la fonction utilisée en backend est un simple **include** PHP.

> *Local File Inclusion ?*

13. A STRANGE COOKIE

En interceptant les requêtes à destination de n'importe quelle page du site, on aperçoit un cookie qui semble "légèrement" suspect :



> *Cookie manipulation ?*

14. A REDIRECT FUNCTIONALITY

En se baladant sur le site et en examinant la manière dont les pages sont affichées, on aperçoit l'URL suivante pour les liens dirigeant vers les réseaux sociaux :

`http://192.168.1.37/index.php?page=redirect&site=facebook`

On a donc un **redirect** utilisé dans cette application, ce qui est toujours potentiellement intéressant.

> *Open Redirect vulnerability ?*

15. PAGE WITH A STRANGE URL

Dans le footer, le copyright est un lien menant à une page avec cet URL :

`http://192.168.1.37/index.php?`
`page=e43ad1fdc54babe674da7c7b8f0127bde61de3fbe01def7d00f151c2fcca6d1c`

Il s'avère que le hash du paramètre "page" est un hash SHA256, correspondant au string **TAMERE**. Pas trop sûr que faire de cette information cela dit. Je ne pense pas qu'il s'agisse d'un flag, étant donné qu'il n'illustrerait aucun exploit tel quel.